



KANNUR UNIVERSITY
കണ്ണൂർ സർവകലാശാല

(Abstract)

FYUG- BCA Programme -Scheme and Syllabus of the Fifth and Sixth Semesters- Approved & Implemented w.e.f 2024 Admission - Orders issued

ACADEMIC C SECTION

FYUGPSC/FYSC-III/9089/2024

Dated: 20.08.2026

- Read:-1.U .O No .ACADC/ACAD C3/22004/2024 dated 29.11.2024
2..U .O No .ACADC/ACAD C3/22004/2024 dated 24.09.2025
3.Letter No.FYUGPSC-FYSC-III/9089/2024 dated 15.04.2025
4.Letter of Even No.dated 03/12/2025
5.Minutes of the meeting of the Expert Committee held on 23.05.2026
6.Email dated 29.05.2026 from Smt. Rechitha C.R, memberof the Expert committee
7.Email dated 26.06.2026 from Dean, Faculty of Technology
8.Orders of the Vice Chancellor in file of even No. dated 13.08.2026

ORDER

1. The Scheme and Syllabus (First and Second Semesters only) of the FYUG BCA Programme were approved and implemented with effect from the 2024 admission, as per the paper read as (1) above.
2. The Scheme and Syllabus of the Third and Fourth Semesters of the FYUG BCA Programme were approved and implemented with effect from the 2024 admission, as per the paper read as (2) above.
3. As per the paper read as (3) above, an Expert Committee was constituted for preparing the complete Scheme and Syllabus of the FYUG BCA Programme.
4. As per the paper read as (4) above, the Expert Committee was reconstituted for preparing the complete Scheme and Syllabus of the FYUG BCA Programme.
5. As per the paper read as (6) above, Smt. Rechitha C.R., Member of the Expert Committee, submitted the Scheme and Syllabus of the Fifth and Sixth Semesters of the FYUG BCA Programme (2024 admission), along with the paper read as (5) above.
6. The Scheme and Syllabus, along with the minutes of the meeting of the Expert Committee, were forwarded to the Dean, Faculty of Technology, for verification. The Dean recommended approval of the same vide the paper read as (7) above.
7. The Vice Chancellor, after considering the recommendation of the Dean, and in exercise of the powers of the Academic Council conferred under Section 11(1), Chapter III of the Kannur University Act, 1996, and all other enabling provisions read together therewith, approved the Scheme and Syllabus of the Fifth and Sixth Semesters of the FYUG BCA Programme and accorded sanction to implement the same with effect from the 2024 admission in the affiliated colleges under the University, subject to reporting to the Academic Council.
8. The Scheme and Syllabus of the Fifth and Sixth Semesters of the FYUG BCA Programme



are appended to this U.O. and uploaded on the University website,
www.kannuruniversity.ac.in.

9. Orders are issued accordingly.

Sd/-

Bindu K P G

Joint Registrar (Acad)

For REGISTRAR

To: 1.The Principals of Arts and Science Colleges affiliated to Kannur University
2.The Controller of Examinations(through the PA)

Copy To: 1. PS to VC / PA to PVC / PA to R/PA to FO/PA to CE (to circulate among the section concerned)
2. DR / AR (Acad) / AR II Exam/JR II Exam
3. Computer Programmer/EXC I/AR VII (Exam)
4. Web Manager (for uploading in the website)
5. SF/DF/FC

Forwarded / By Order


SECTION OFFICER



KANNUR UNIVERSITY

Re-accredited by NAAC with 'B++' Grade

KUFYUGP

BACHELOR OF COMPUTER APPLICATIONS

SECOND & THIRD YEAR CURRICULUM & SYLLABUS

For 2024 Admission



www.kannuruniversity.ac.in

PREFACE

Welcome to the Four-year Integrated UG COMPUTER APPLICATIONS Programme of Kannur University. The Board of Studies of the University has designed this programme on the basis of the National Education Policy 2023 which critically envisions a brand-new holistic education system for the country, hinging on the effective adoption of modern teaching and training methods, application of technology, and imparting practical and contemporary skills, to shape the overall personality of students. Our programme is designed to equip students with a strong foundation in COMPUTER APPLICATIONS principles while also providing specialized training in Artificial Intelligence and Machine Learning. In today's digital age, these technologies are at the forefront of technological advancements, driving innovation across various industries including healthcare, finance, transportation, and entertainment.

The Programme in COMPUTER APPLICATIONS is designed with the objective of equipping the students to cope with the emerging trends and challenges in the field of computers and interrelated disciplines like computer engineering, COMPUTER APPLICATIONS, information systems, information technology, and software engineering. This programme involves various courses such as Value-added courses, Skill enhancement courses, multi-disciplinary courses and ability enhancement courses with an attribution of discipline specific core, discipline specific electives and various scholastic and co scholastic domains. This programme aims at helping the students define and recharge their creative, analytical, problem-solving, and critical thinking abilities, topped by actively pursuing digital literacy.

The BCA COMPUTER APPLICATIONS Honours program emphasizes a strong theoretical foundation complemented by extensive laboratory experience. Students will engage in hands-on experiments that reinforce classroom learning and develop critical technical skills. Through practical work, they will learn to design application programs, software, and analyse data, thus bridging the gap between theory and practice.

For those opting for the BCA COMPUTER APPLICATIONS Honours with Research track, the program offers an enriched experience with a significant focus on independent research. This track is designed for students who wish to delve deeper into specific areas of interest, culminating in a research thesis. Under the mentorship of faculty members, students will undertake original research projects, honing their ability to conduct scientific inquiries,

think critically, and contribute to the body of knowledge in COMPUTER APPLICATIONS. This rigorous training prepares graduates for careers in academia, research institutions, and industry.

The successful revision of this curriculum would not have been possible without the collective efforts and inputs from the Expert Committee members, Ad hoc committee members, COMPUTER APPLICATIONS academic council member, resource persons and the unwavering support of COMPUTER APPLICATIONS faculty members from the affiliated colleges. Their dedication and expertise have played an instrumental role in shaping a curriculum that is relevant, up-to-date, and consistent with international scholarly criteria.

We wish you to have a motivating atmosphere to make use of your extreme potential and caliber to complete this programme and to serve the nation by enriching yourself.

BEST WISHES

(BCA Syllabus Expert Committee)

BCA Syllabus Expert Committee

1	Ms. Kavitha M R, HoD & Assistant Professor, KBS Government College, Thalassery
2	Ms. Rechitha C R, Assistant Professor, KBS Government College, Thalassery
3	Ms. Anitha Haridas, Assistant Professor, Chinmaya Arts & Science College, Chala
4	Mr. Kevinson Kurian, HoD, Dept of Computer Science, Don Bosco Arts & Science College

Adhoc Member

1	SreevidyaPN, Chairperson, BoE-BCA-Odd Semesters, Assistant Professor, KBS Government College, Thalassery
---	--

Special Thanks To

Dr Daphna Chacko, Assistant Professor, Department Of Computer Science , Krishna Menon Memorial Govt. Women's College

INTRODUCTION

Kannur University - Four-Year Undergraduate Programme: Backdrop and Context

The implementation of the Four-Year Undergraduate Programme (FYUGP) has been driven by the pressing need to address contemporary challenges ensuring responsive changes to the evolving needs of students, industry, and society at large. Recognizing the curriculum as the cornerstone of any education system, it requires regular refinement to align with evolving socioeconomic factors. Higher education must provide students with practical and technical skills relevant to their fields of interest, necessitating the development of a job-oriented curriculum. Despite significant increases in access and expansion of higher education over the years, concerns persist regarding the quality and relevance of educational outcomes, particularly in terms of employability skills. As the world becomes increasingly interconnected, our education system must evolve to instill 21st-century skills, enabling students not only to survive but to thrive in this dynamic environment. Moreover, there is a growing need for higher education institutions to embrace social responsibility and contribute to the development of a knowledge society capable of driving sustainable development through innovation. With the central objective of fostering a robust knowledge society to support a knowledge economy, the Government of Kerala has initiated steps to reform higher education. Accordingly, three commissions were established to suggest reforms in higher education policy, legal and regulatory mechanisms, and evaluation and examination systems. It is within this context that a comprehensive reform of the undergraduate curriculum has been proposed, leading to the restructuring of the Four-Year Undergraduate Programme (FYUGP).

VISION AND MISSION OF KANNUR UNIVERSITY

Vision:

To establish a teaching, residential and affiliating University and to provide equitable and just access to quality higher education involving the generation, dissemination and a critical

application of knowledge with special focus on the development of higher education in Kasargod and Kannur Revenue Districts and the Manandavady Taluk of Wayanad Revenue District.

Mission:

- To produce and disseminate new knowledge and to find novel avenues for application of such knowledge.
- To adopt critical pedagogic practices which uphold scientific temper, the uncompromised spirit of enquiry and the right to dissent.
- To uphold democratic, multicultural, secular, environmental and gender sensitive values as the foundational principles of higher education and to cater to the modern notions of equity, social justice, and merit in all educational endeavours.
- To affiliate colleges and other institutions of higher learning and to monitor academic, ethical, administrative, and infrastructural standards in such institutions.
- To build stronger community networks based on the values and principles of higher education and to ensure the region's intellectual integration with national vision and international standards.
- To associate with the local self-governing bodies and other statutory as well as nongovernmental organizations for continuing education and also for building public awareness on important social, cultural and other policy issues.

PROGRAMME OUTCOMES

PO1	Critical Thinking and Problem-Solving-Apply critical thinking skills to analyze information and develop effective problem-solving strategies for tackling complex challenges.
PO2	Effective Communication and Social Interaction-Proficiently express ideas and engage in collaborative practices, fostering effective interpersonal connections.

PO3	Holistic Understanding-Demonstrate a multidisciplinary approach by integrating knowledge across various domains for a comprehensive understanding of complex issues.
PO4	Citizenship and Leadership-Exhibit a sense of responsibility, actively contribute to the community, and showcase leadership qualities to shape a just and inclusive society.
PO5	Global Perspective-Develop a broad awareness of global issues and an understanding of diverse perspectives, preparing for active participation in a globalized world.
PO6	Ethics, Integrity and Environmental Sustainability-Uphold high ethical standards in academic and professional endeavors, demonstrating integrity and ethical decision-making. Also acquire an understanding of environmental issues and sustainable practices, promoting responsibility towards ecological well-being.
PO7	Lifelong Learning and Adaptability-Cultivate a commitment to continuous self- directed learning, adapting to evolving challenges, and acquiring knowledge throughout life.

PROGRAMME SPECIFIC OUTCOMES

PSO1:	Apply the knowledge of Computer Applications to understand, analyze, and solve real-world computing problems effectively.
PSO2:	Design, develop, test, and maintain software applications using modern programming languages, tools, and technologies.
PSO3:	Apply modern computing techniques for efficient organization, processing, storage, retrieval, and secure handling of information.
PSO4:	Apply current computing tools, technologies, and methodologies ethically to create innovative and practical IT solutions.
PSO5:	Work collaboratively and apply analytical and creative thinking to solve complex problems in Computer Applications and related fields.

PSO6:	Demonstrate professional ethics, social responsibility, environmental awareness, teamwork, and effective communication in computing practices.
PSO7:	Adapt to emerging technologies through continuous learning, self-improvement, and professional development.

STRUCTURE OF THE PROGRAMME

The Programme of instruction will consist of Lecture courses, Practical courses, comprehensive Viva-voce, Seminar, internship/ industrial visit, and Project work.

- 1. Lecture courses:** Courses involving lectures relating to a field or discipline by a faculty member
- 2. Tutorial courses:** Courses involving problem-solving and discussions relating to a field or discipline under the guidance of qualified personnel in a field of learning,
- 3. Laboratory work:** A course requiring students to participate in a project or practical or lab activity that applies previously learned/studied principles/theory related to the chosen field of learning, work/vocation, or professional practice under the supervision of an instructor.
- 4. Comprehensive Viva-voce :** This is an essential assessment included in the Programme to evaluate the student's grasp of the subject matter and their ability to apply their knowledge as defined in the course outcomes. It also provides an opportunity for the student to engage in academic discussions and receive valuable feedback from experts in the field.
- 5. Seminar:** A course requiring students to participate in structured discussion/conversation or debate focused on assigned tasks/readings, current or historical events, or shared experiences guided or led by an expert or qualified personnel in a field of learning
- 6. Internship/ Institutional visit:** All students shall undergo a Field Trip/Summer Internship/Apprenticeship in a Firm, Industry or Organization; or Training in labs with faculty and researchers or other Higher Education Institutions (HEIs) or research institutions. Departments can actively promote internships that can eventually lead to research project

work. Institutional visit Incorporating institutional or industrial visits in the Programme brings immense value to the students, making their learning journey more enriching and preparing them for successful careers in COMPUTER APPLICATIONS-related fields.

7. **Research Project:** These students who have opted for the honours with research should complete a research project under the guidance of the mentor and should submit a research report for evaluation. They need to successfully defend the research project to obtain 12 credits under a faculty member of the University/College. The research shall be in the Major discipline

Course and Credit Structure for Different Pathways

Course Distribution for Students in Semesters I – VI

(1) **Single Major:** The 6 courses together in B and C can be in different disciplines.

(2) **Major with Multiple Disciplines:** B and C represent two different disciplines.

(3) **Major with Minor:** B and C represent the same Minor discipline.

I SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	AEC1 (English)	4	3	75
2	AEC2 (Additional Language)	3	3	75
3	MDC	3	3	75
4	DSC A1	5	4	100
5	DSC B1	4/5	4	100
6	DSC C1	4/5	4	100
Total		23/25	21	525

II SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	AEC2 (English)	4	3	75

2	AEC3 (Additional Language)	3	3	75
3	MDC	3	3	75
4	DSC A2	5	4	100
5	DSC B2	4/5	4	100
6	DSC C2	4/5	4	100
Total		23/25	21	525

III SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	MDC (Kerala Studies)	3	3	75
2	VAC 1	3/4	3	75
3	DSC A3	4/5	4	100
4	DSC A4	4/5	4	100
5	DSC B3	4/5	4	100
6	DSC C3	4/5	4	100
Total		22/27	22	550

IV SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 1	3/4	3	75
2	VAC 2	3/4	3	75
3	VAC 3	3/4	3	75
4	DSC A5	5	4	100
5	DSC A6	5	4	100

6	DSC A7	5	4	100
Total		24/27	21	550

V SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 2	3/4	3	75
2	DSC A8	5	4	100
3	DSC A9	4/5	4	100
4	DSC A10	4	4	100
5	DSE A11	4/5	4	100
6	DSE A12	4/5	4	100
7	Internship	0	2	50
Total		24/28	25	575

VI SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 3	3/4	3	75
2	DSC A13	4	4	100
3	DSC A14	4	4	100
4	DSC A15*	8	4	100
5	DSE A16	4/ 5	4	100
6	DSE A17	4/ 5	4	100
Total		27/30	23	625

* This course is mandatory

- (4) *Double major pathway*: A and B represent the courses offered by the two departments. Students should choose one of the disciplines as their major 1 and the other as major 2

I SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	AEC1 (English)	4	3	75
2	AEC2 (Additional Language)	3	3	75
3	MDC	3	3	75
4	DSC A1	5	4	100
5	DSC B1	4/5	4	100
6	DSC C1	4/5	4	100
Total		23/25	21	525

II SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	AEC2 (English)	4	3	75
2	AEC3 (Additional Language)	3	3	75
3	MDC	3	3	75
4	DSC A2	5	4	100
5	DSC B2	4/5	4	100
6	DSC C2	4/5	4	100
Total		23/25	21	525

III SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks

1	MDC (Kerala Studies)	3	3	75
2	VAC 1	3/4	3	75
3	DSC A3	4/5	4	100
4	DSC A4	4/5	4	100
5	DSC B3	4/5	4	100
6	DSC C3	4/5	4	100
Total		22/27	22	550

IV SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 1	3/4	3	75
2	VAC 2	3/4	3	75
3	VAC 3	3/4	3	75
4	DSC A5	5	4	100
5	DSC A6	5	4	100
6	DSC A7	5	4	100
Total		24/27	21	550

V SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 2	3/4	3	75
2	DSC A8	5	4	100
3	DSC A9	4/5	4	100

4	DSC A10	4	4	100
5	DSE A11	4/5	4	100
6	DSE A12	4/5	4	100
7	Internship	0	2	50
Total		24/28	25	575

VI SEMESTER				
Sl No.	Course	Hours/Week	Credits	Total Marks
1	SEC 3	3/4	3	75
2	DSC A13	4/5	4	100
3	DSE A14	4/5	4	100
4	DSC A15	8	4	100
5	DSC A16	4/ 5	4	100
6	DSE A17	4/ 5	4	100
Total		27/32	23	625

On completion of 3-yearUG Student will get 68 credits in majorA (48 + 18+2 = 68 (50%of 133)) and 53 credits in MajorB (44 + 9 = 53 (40% of 133))

GENERAL FOUNDATION COURSES							
MULTI DISCIPLINARY COURSES (MDC)							
SEME STER	COURSE CODE	COURSE NAME	CREDITS			HO URS PER WE EK	MARKS
			LEC TUR E/ TUT ORI AL	PR AC TI CA L	TO TA L		
I	KU1MDCCAP101	Basics of IT for all	3	0	3	3	75
	KU1MDCCAP102	Digital Marketing	3	0	3	3	75
II	KU2MDCCAP103	Python Programming for all	3	0	3	3	75
	KU2MDCCAP104	Introduction to Data Science	3	0	3	3	75
VALUE ADDED COURSES (VAC)							
SEME STER	COURSE CODE	COURSE NAME	CREDITS			HO URS PER WE EK	MARKS
			LEC TUR E/ TUT ORI AL	PR AC TI CA L	TO TA L		
III	KU3VACCAP101	Hardware and Networking Essentials	2	1	3	4	75
	KU3VACCAP102	Cyber Law and Ethics	3	0	3	3	75
	KU3VACCAP103	Introduction to Data Analytics	3	0	3	3	75
IV	KU4VACCAP104	Basics of Computer Networks	3	0	3	3	75
	KU4VACCAP105	Basics of Internet of Things	3	0	3	3	75
	KU4VACCAP106	R Programming for Data Analytics	2	1	3	4	75

SKILL ENHANCEMENT COURSES (SEC)							
SEM EST ER	COURSE CODE	COURSE NAME	CREDITS			HO UR S PE R WE EK	MARKS
			LEC TU RE/ TUT ORI AL	PR AC TI CA L	TO TA L		
IV	KU4SECCAP101	Software Project Management	3	0	3	3	75
	KU4SECCAP102	Artificial Intelligence (AI) Assisted Tools	2	1	3	4	75
	KU4SECCAP103	Operating System Administration	2	1	3	4	75
V	KU5SECCAP104	Digital Marketing	2	1	3	4	75
	KU5SECCAP105	Game Application Development	2	1	3	4	75
	KU5SECCAP106	Feature Engineering using python	2	1	3	4	75
	KU5SECCAP107	PHP and MySQL	2	1	3	4	75
	KU5SECCAP108	Agile Software Development for SDE	3	0	3	3	75
VI	KU6SECCAP109	Fundamentals of Cloud Computing	3	0	3	3	75
	KU6SECCAP110	Android Application Development	2	1	3	4	75
	KU6SECCAP111	Entrepreneurship in Software Engineering	3	0	3	3	75
	KU6SECCAP112	Professional Skills and Placement Readiness	3	0	3	3	75
	KU6SECCAP113	UI/UX Design	2	1	3	4	75

DISCIPLINE SPECIFIC COURSES							
SEM ESTER	COURSE CODE	COURSE NAME	CREDITS			HO URS PER WE EK	MARKS
			LECT URE/ TUT ORIA L	PR AC TI CA L	TO TA L		
I	KU1DSCCAP101	Foundations of Computers and Programming	3	1	4	5	100
	KU1DSCCAP102	Design Thinking	3	1	4	5	100
	KU1DSCCAP103	Essential IT Tools	3	1	4	5	100
	KU1DSCCAP104	Fundamentals of App Development	3	1	4	5	100
	KU1DSCCAP105	Fundamentals of Web Development	3	1	4	5	100
II	KU2DSCCAP106	Programming with C and C++	3	1	4	5	100
	KU2DSCCAP107	Multimedia and Graphic Designing	3	1	4	5	100
	KU2DSCCAP108	Cyber Security and Ethics	4	0	4	4	100
	KU2DSCCAP109	Introduction to Database Management System	3	1	4	5	100
	KU2DSCCAP110	Ethical Hacking	3	1	4	5	100
III	KU3DSCCAP201	Discrete Mathematics	4	0	4	4	100
	KU3DSCCAP202	Object Oriented Programming through Java	3	1	4	5	100
	KU3DSCCAP203	Digital Systems & Introduction to Microprocessors	3	1	4	5	100
	KU3DSCCAP204	Mathematical Foundations of Computing	3	1	4	5	100
	KU3DSCCAP205	Linux System Administration	3	1	4	5	100

	KU3DSCCAP206	Mobile Application Development	3	1	4	5	100
IV	KU4DSCCAP207	Software Engineering	4	0	4	4	100
	KU4DSCCAP208	Database Management System	3	1	4	5	100
	KU4DSCCAP209	Data Structures and Algorithms	3	1	4	5	100
V	KU5DSCCAP301	Python Programming	3	1	4	5	100
	KU5DSCCAP302	Computer Network and Network security	4	0	4	4	100
	KU5DSCCAP303	Web Technology	3	1	4	5	100
	KU5DSCCAP304	Artificial Neural Network	3	1	4	5	100
	KU5INTERN301	Internship	0	0	2	0	50
VI	KU6DSCCAP306	Theory of Computation and applications in Compiler Design	4	0	4	4	100
	KU6DSCCAP307	Principles of Operating Systems	4	0	4	4	100
	KU6DSCCAP308	Design and Analysis of Algorithms	4	0	4	4	100
	KU6DSCCAP309*	Applied Computing	0	4	4	8	100
	KU6DSCCAP310	Introduction to Deep learning	3	1	4	5	100
	KU6DSCCAP311	Introduction to Compiler Design	4	0	4	4	100

* This course is mandatory

ELECTIVE COURSES IN BCA WITH SPECIALISATION							
SEM EST ER	COURSE CODE	COURSE NAME	CREDITS			HO URS PER WE EK	MARKS
			LE CT UR E/ TU TO RI AL	PR AC TI CA L	TO TA L		
Data Science #							
V	KU5DSECAP301	Statistical Methods for Data Science using Python	3	1	4	5	100
V	KU5DSECAP302	Introduction to Data Science	3	1	4	5	100
VI	KU6DSECAP303	Data Mining Concepts and Algorithms	4	0	4	4	100
VI	KU6DSECAP304	Machine Learning using Python	3	1	4	5	100
Artificial Intelligence & Machine Learning *							
V	KU5DSECAP305	Mathematical Foundations of Machine Learning	3	1	4	5	100
V	KU5DSECAP306	Artificial Intelligence	4	0	4	4	100
VI	KU6DSECAP307	Machine Learning using Python	3	1	4	5	100
VI	KU6DSECAP308	Natural Language Processing	3	1	4	5	100
Application Development							

V	KU5DSECAP309	Frontend Development using React	3	1	4	5	100
V	KU5DSECAP310	Database Management for Applications	3	1	4	5	100
VI	KU6DSECAP311	Backend Development & Full Stack Integration	3	1	4	5	100
VI	KU6DSECAP312	Mobile Application Development using Flutter	3	1	4	5	100
Cyber Security & Cloud Computing							
V	KU5DSECAP313	Introduction to Cyber Security	3	1	4	5	100
V	KU5DSECAP314	Cryptography	4	0	4	4	100
VI	KU6DSECAP315	Ethical Hacking	3	1	4	5	100
VI	KU6DSECAP316	Cloud Security and Introduction to Digital Forensic	4	0	4	4	100
* BCA Artificial Intelligence students shall select the Artificial Intelligence & Machine Learning basket as Discipline Specific Elective in Fifth and Sixth Semesters # BCA Data Science students shall select the Data Science as Discipline Specific Elective in Fifth and Sixth Semesters							

ELECTIVE COURSES IN BCA WITH NO SPECIALISATION							
SEM EST ER	COURSE CODE	COURSE NAME	CREDITS			HO URS PER WE EK	MARKS
			LE CT UR E/ TU TO RI AL	PR AC TI CA L	TO TA L		
V	KU5DSECAP317	Cryptography	4	0	4	4	100
	KU5DSECAP318	Cloud computing and Big Data	3	1	4	5	100
	KU5DSECAP319	Artificial Intelligence and Neural Network	4	0	4	4	100
	KU5DSECAP320	Computer Organization	4	0	4	4	100
	KU5DSECAP321	Computer Graphics	3	1	4	5	100
VI	KU6DSECAP322	Soft Computing	4	0	4	4	100
	KU6DSECAP323	Mobile Application Development Using Flutter	3	1	4	5	100
	KU6DSECAP324	Introduction to Blockchain	4	0	4	4	100
	KU6DSECAP325	Parallel Computing Concepts	4	0	4	4	100
	KU6DSECAP326	Digital Image Processing	3	1	4	5	100
	KU6DSECAP327	Software Testing	4	0	4	4	100

Selection of Discipline Specific Elective Course

- The minimum number of elective courses that can be offered in the three-year programme shall be four, with two each in semester V and VI.
- The maximum number of elective courses that can be offered in the three-year programme shall be determined based on the workload of the department.
- If the minimum strength for a discipline specific elective course is not met, students may be required to choose an alternate discipline specific elective.
- The Department/College shall notify the list of available elective courses at the beginning of each academic semester.
- Students shall be required to successfully complete four of elective courses from the approved pool of specialisation basket or from no specialization basket.
- If four elective courses in semesters V and VI, with total 16 credits, belong to one specialization, they constitute a specialisation in the Major discipline.
- Students opting for a specialization must select all mandatory elective papers from the prescribed specialization basket.
- Students opting for no specialization can select courses from either specialization basket or from no specialization basket.

ASSESSMENT AND EVALUATION

- The assessment shall be a combination of Continuous Comprehensive Assessment (CCA) and an End Semester Evaluation (ESE)
- As per the regulation of Kannur University, one credit corresponds to 25 marks. Hence a 3-credit course must be evaluated for 75 marks and 4 credit courses for 100 marks. The ratio of continuous comprehensive assessment (CCA) to End semester examination (ESE) for theory/lecture courses is 30:70 and for the practical courses, it is 40:60.
- The 4-credit courses (Discipline specific courses, Discipline specific elective)
 - i. courses with only theory
 - ii. courses with 3-credit theory and 1-credit practical.
 - iii. courses with only practical

1. In 4-credit courses with only theory components, out of the total 5 modules of the syllabus, one teacher specific module with 20% content is designed by the faculty member teaching that course, and it is internally evaluated.
2. In 4-credit courses with 3-credit theory and 1-credit practical components, out of the total 5 modules of the syllabus, 4 modules are for theory and the fifth module is for practical.
3. In 4-credit courses with 4 credit practical, skills/knowledge implementation in research//prototype/software development, the process of translating conceptual designs, architectural guidelines, and business logic into functional, deployable code.

Course Credit	Credit		Mark		L		P		
	L	P	L	P	CCA	ESE	CCA	ESE	Total marks
4	4	0	100	0	30	70	0	0	100
	3	1	75	25	25	50	10	15	100
	0	4	0	100	0	0	40	60	100
3	L	P	L	P	CCA	ESE	CCA	ESE	Total marks
	3	0	75	0	25	50	0	0	75
	2	1	50	25	15	35	10	15	75

- The 3 credit courses (Foundational Courses) are of two types:
 - i courses with only theory
 - ii courses with 2-credit theory and 1-credit practical.
- In 3-credit courses with only theory components, out of the total 5 modules of the syllabus, one teacher specific module with 20% content is designed by the faculty member teaching that course, and it is internally evaluated.
- In 3-credit courses with 2-credit theory and 1-credit practical components, out of the total 5 modules of the syllabus, 4 modules are for theory and the fifth module is for practical.

- Continuous Evaluation includes assignments, seminars, periodic written examinations, or other measures as proposed in the syllabus and approved by the university.

Practical exams

- There shall be a Continuous Evaluation of practical courses conducted by the Course- In-Charge.
- An observation book should be maintained for the experiments done in the lab and the same should be evaluated during the continuous evaluation.
- The process of continuous evaluation of practical courses shall be completed before 10 days from the commencement of the end-semester examination.
- The end-semester practical examination and viva-voce, and the evaluation of practical records shall be conducted by the course in-charge and an internal examiner appointed by the Department Council. Duration of ESE may be 2 to 2.5 Hrs.
- Those who passed in continuous evaluation alone will be permitted to appear for the end semester examination and viva-voce
- The end semester practical examination will in general have the following components:

SI No	Component of Evaluation	Marks
1	MODIFICATION	2
2	RECORD	2
3	VIVA	3
4	CODE WRITING	3
5	OUTPUT	5
TOTAL		15

KU1DSCCAP101- Foundations of Computers and Programming, KU1DSCCAP103- Essential IT Tools/ KU2DSCCAP107 - Multimedia and graphic designing, KU4SECCAP102 Artificial Intelligence (AI) Assisted Tools will have the following components as 4 and 5

#For the course **KU2DSCCAP106- Programming With C and C++** Code writing is divided into Part A and Part B each of 1.5 marks and output is divided into Part A and Part B each of 2.5 marks.

KU1DSCCAP101- Foundations of Computers and Programming		
1	CODE WRITING	3
2	ALGORITHM/ FLOWCHART	2
3	OUTPUT	3
KU1DSCCAP103- Essential IT Tools/ KU2DSCCAP107 - Multimedia and graphic designing		
1	PERFECTION OF THE WORK	4
2	COMPLETENESS	4
KU4SECCAP102 Artificial Intelligence (AI) Assisted Tools		
1	PROMPTS / STEPS WRITING	4
2	PERFECTION OF THE WORK	4

Mark Distribution for Courses

The detailed mark distribution for 3 credit and 4 credit courses are given below:

L – Lecture/Theory, P – Practical/Practicum components, CCA – Continuous Comprehensive Assessment, ESE – End Semester Evaluation

- 4 Credit Course (Theory only)**

Evaluation Type		Marks
ESE		70
CCA		30
a)	*Test Paper	15
b)	**Assignment/ Book- Article Review	10
c)	Seminar/ Viva -Voce /Case Study	5
Total		100

- 4 Credit Course (3 credit theory + 1 credit practical)**

Evaluation Type			Marks	Evaluation Type			Marks	Total
Lecture			75	Practical			25	100
a)	ESE		50	a)	ESE		15	
b)	CCA		25	b)	CCA		10	
	i	*Test Paper	12		i	Punctuality and Lab Skill	3	
	ii	**Assignment/ Book- Article Review	5		ii	Test Papers	5	
	iii	Seminar/ Viva-Voce/Case Study	8		iii	Observation Book	2	

● **3 Credit Course (Theory only)**

Evaluation Type		Marks
ESE		50
CCA		25
a)	*Test Paper	12
b)	**Assignment/ Book- Article Review	5
c)	Seminar/ Viva -Voce /Case study	8
Total		75

● **3 Credit Course (2 credit theory + 1 credit practical)**

Evaluation Type			Marks	Evaluation Type			Marks	Total
Lecture			50	Practical			25	
a)	ESE		35	a)	ESE		15	
b)	CCA		15	b)	CCA		10	

	i	*Test Paper	8		i	Test Papers	5	75
	ii	**Book-Article review/ Assignment	2		ii	Lab Skill and Punctuality	3	
	iii	Seminar/ Viva-Voce/Case study	5		iii	Observation Book	2	

* **Minimum two test papers**

** **Or any other evaluation technique like quiz, open book exam, group activity etc**

- **4 Credit Course (0 credit theory + 4 credit practical)**

Evaluation Type		Marks
ESE		60
a)	Presentation (Demonstration of the project or present the research findings)	15
b)	Viva voce based on presentation	15
c)	Novelty, Relevance & Contribution	10
d)	Version control using Git and GitHub	10
e)	Report	10
CCA		40
a)	Proposal and Planning	5
b)	Punctuality & Work book	5
c)	Presentation & Viva-Voce (Minimum 3 presentations)	30
Total		100

INTERNSHIP

- All students should undergo Internship of 2-credits during the fourth semester summer break in a firm, industry or organization, or training in labs with faculty and researchers of their own institution or other Higher Educational Institutions (HEIs) or research institutions.
- Internship can be for enhancing the employability of the student or for developing the research aptitude.
- Internship can involve hands-on training on a particular skill/ equipment/ software. It can be a short project on a specific problem or area. Attending seminars or workshops related to an area of learning or skill can be a component of Internship.
- A faculty member/ scientist/ instructor of the respective institution, where the student does the Internship, should be the supervisor of the Internship.
- Refer KU5INTERN301 Internship, course page and appendix for guidelines and proforma

PROJECT IN HONOURS PROGRAMME

- In the Honours programme, the student has the option to do a Project of 12-credits in Major instead of three major Courses or Project of 8-credits in Major and one major course in semester 8.
- The Project can be done in the same institution/ any other higher educational institution (HEI)/ research centre/ training centre.
- The Project in Honours programme can be a short research work or an extended internship or a skill-based training programme.
- A faculty member of the respective institution, where the student does the Project, should be the supervisor of the Project.

PROJECT IN HONOURS WITH RESEARCH PROGRAMME

- Students who secure 75% marks and above (equivalently, CGPA 7.5 and above) cumulatively in the first six semesters are eligible to get selected to Honours with Research stream in the fourth year.
- In Honours with Research programme, the student has to do a mandatory Research Project of 12-credits instead of three Core Courses in Major in semester 8.
- The number of seats for the Honors with research shall be determined as per the availability of eligible faculty.
- The selection criteria for Honors with research stream shall be in accordance with the guidelines of UGC or as approved by Kannur University.
- Students who have chosen the honours with research stream shall be mentored by a faculty with a PhD.
- The mentor shall prescribe suitable advanced-level courses for a minimum of 20 credits to be taken within the institutions along with the papers on research methodology, research ethics, and research topic-specific courses for a minimum of 12 credits which may be obtained either within the institution or from other recognized institutions, including online and blended modes.
- These students who have opted for the honours with research should complete a research project under the guidance of the mentor and should submit a research report for evaluation. They need to successfully defend the research project to obtain 12 credits under a faculty member of the University/College within the University.
- The research outcomes of their project work may be published in peer-reviewed journals or presented at conferences or seminars or patented.

EXTERNAL EVALUATION

- Examinations will be conducted at the end of each semester.
- Individual questions are evaluated in marks and the total marks are converted into grades by the University based on a 10-point grading system.

Letter Grade	Grade Point (P)
--------------	-----------------

O (Outstanding)	10
A+ (Excellent)	9
A (Very Good)	8
B+ (Good)	7
B (Above Average)	6
C (Average)	5
P (Pass)	4
F (Fail)	0
Ab (Absent)	0

- A minimum of grade point 4 (Grade P) is needed for the successful completion of a Course.
- A student who has failed in a Course can reappear for the End Semester Examination of the same Course along with the next batch without taking readmission or choose another Course in the subsequent Semesters of the same programme to acquire the minimum credits needed for the completion of the Programme.
- There shall not be provision for improvement of CE and ESE.
- A student who has successfully completed the CE requirements in a subsequent semester can also appear for the ESE subject to the maximum duration permitted.

Computation of SGPA and CGPA

The following method is recommended to compute the Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA):

- The SGPA is the ratio of the sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e. **SGPA** (S_i) = $\Sigma(C_i \times G_i) / \Sigma C_i$ Where C_i is the number of credits of the course and G_i is the grade point scored by the student in the course.

Example:

Semester	Course	Credit	Letter Grade	Grade point	Credit Point (Credit x Grade)
I	Course 1	3	A	8	3 X 8 = 24
I	Course 2	4	B+	7	4 X 7 = 28
I	Course 3	3	B	6	3 X 6 = 18
I	Course 4	3	O	10	3 X 10 = 30
I	Course 5	3	C	5	3 X 5 = 15
I	Course 6	4	B	6	4 X 6 = 24
		20			139
SGPA					139/20= 6.95

- The Cumulative Grade Point Average (CGPA) is also calculated in the same manner taking into account all the courses undergone by a student over all the semesters of a programme, i.e. $CGPA = \sum(C_i \times S_i) / \sum C_i$ Where S_i is the SGPA of the semester and C_i is the total number of credits in that semester.
- The SGPA and CGPA shall be rounded off to 2 decimal points and reported in the transcripts. Transcript (Format): Based on the above recommendations on Letter grades, grade points and SGPA and CGPA, the HEIs may issue the transcript for each semester and a consolidated transcript indicating the performance in all semesters.

Example:

Semester I	Semester II	Semester III	Semester IV	Semester V	Semester VI
Credit: 21	Credit: 21	Credit: 22	Credit: 24	Credit: 23	Credit: 22
SGPA: 6.9	SGPA: 7.8	SGPA: 5.6	SGPA: 6.0	SGPA: 6.3	SGPA: 8.0
$CGPA = (21 \times 6.9 + 21 \times 7.8 + 22 \times 5.6 + 24 \times 6.0 + 23 \times 6.3 + 22 \times 8.0) / 133 = 6.74$					

- The SGPA is the ratio of the sum of the product of the number of credits with the grade points scored by a student in all the courses taken by a student and the sum of the number of credits of all the courses undergone by a student, i.e. **SGPA** (S_i) = $\Sigma(C_i \times G_i) / \Sigma C_i$ Where C_i is the number of credits of the course and G_i is the grade point scored by the student in the course.

CGPA	Overall letter Grade
9.5 and above	O
8.5 and above but less than 9.5	A+
7.5 and above but less than 8.5	A
6.5 and above but less than 7.5	B+
5.5 and above but less than 6.5	B
4.5 and above but less than 5.5	C
4.0 and above but less than 4.5	D
Less than 4.0	F

Appearances for Continuous Evaluation (CE) and End Semester Examination (ESE) are compulsory, and no Grade shall be awarded to a candidate if the candidate is absent for CE or ESE or both.

VALUE ADDED COURSES (VAC)

KU3VACCAP101: HARDWARE AND NETWORKING ESSENTIALS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	VAC	100 - 199	KU3VACCAP101	3 (2T+ 1P)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	-	25	50	75	1.5 hrs.

Course Description:

This value-added course is designed to provide learners with foundational knowledge of computer systems and networking, essential for any modern technical or non-technical profession. The course introduces students to different types of computers and microcomputers, their hardware and software components, and the basic structure of computer programming languages. It also offers an in-depth understanding of the internal architecture of systems, including CPUs, memory units, motherboards, and I/O devices. Moving beyond standalone systems, the course covers core concepts in computer networking such as network types (LAN, WAN), topologies, protocols (TCP/IP, HTTP, FTP), and the OSI model.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Identify various types of computers, microcomputers, and their software components.	R

2	Explain the function of hardware components such as CPU, memory, motherboard, and I/O devices.	U
3	Compare different computer languages and operating systems (e.g., Windows vs. Linux).	An
4	Describe basic networking concepts, types of networks, OSI model, and network devices.	U
5	Configure simple network setups using IPv4/IPv6 in both peer-to-peer and client-server models.	A
6	Differentiate between network protocols and topologies and interpret how they affect network design.	An

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2						2
CO 2				2			
CO 3	2	2					
CO 4	2	2		2			
CO 5		2		2		2	
CO6	2			2			2

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
1		Introduction to Computers and Software	7

	1 2 3 4	Types of Computers: Desktop, Laptop, Tablet, and Microcomputers Software Components: Operating System (OS), Driver Software, Application Software, Utility Software Features of Operating Systems: Key Differences between Windows and Linux Computer Languages: High-Level, Low-Level, and Machine Language, Assembler, Compiler, Linker, Loader	
2	Hardware Components and Microprocessor Basics		8
	1	Hardware Components: CPU, Input Devices, Output Devices	
	2	Motherboard: Internal and External Connectors, Chipset, Platform Controller Hub (PCH), Clock Generator, BIOS, CMOS	
	3	Microprocessor: Execution Unit, Control Unit, Cache Memory	
	4	Memory: Introduction to Primary and Secondary Memory, DRAM vs. SRAM, ROM and its types, Role of ROM in a computer	
	5	I/O Devices: Keyboard, Monitor, Printer, Mouse, Touchscreen	
3	Computer Networking Fundamentals		8
	1	Introduction to Computer Networking: Benefits and Importance of Networking, Types of Networks: PAN, LAN, MAN, WAN	
	2	Network Packet Structure: Packet, Segment, and Frame, Contents of a Packet Header	
	3	OSI Model (7 Layers): Application, Presentation, Session, Transport, Network, Data Link, Physical	
4	Network Protocols, Addressing, and Configuration		7

	1	Network Protocols: TCP/IP, IP, HTTP, HTTPS, FTP, SMTP	
	2	Network Topologies: Bus, Star, Ring, Mesh.	
	3	Devices: Hub, Switch, Router, Bridge, Repeater, Gateway, Modem	
	4	Network Addressing: Logical Address, Physical Address, Port Address, Specific Address	
	5	Cables and Connectors: Twisted Pair, Coaxial, Fiber Optic	
	6	Network Configuration: Peer-to-Peer (IPv4/IPv6) – Windows, Client–Server Configuration (IPv4), Internet Café Setup (IPv4), Domain-Based Network System	
5	Teacher Specific Module		30
	<i>Directions</i>		
	1. Identifying Hardware Components of a Computer : a. Power off the computer and open the CPU cabinet b. Identify the components on the motherboard c. Observe I/O devices d. Document the purpose of each device 2. PC Assembling 3. Linux installation 4. Illustration of Networking Hardware components 5. Identifying and Comparing Network Cables a. Document where each cable is used and its transmission speed range. b. Crimping: Physical Demonstration (Preferred). If hardware components are unavailable, demonstrate the procedure using instructional videos or network simulation software. 6. Identification of logical address, physical address, and port address 7. Setup a Wired/Wireless LAN with more than two systems		

	8. Configure IP address of a system and Share files over LAN 9. Testing network connectivity using Ping and IPConfig commands 10. Case study : Understand Troubleshooting basic hardware and network issues	
--	---	--

Essential Readings:

- 1.Computer Fundamentals *Author:* P.K. Sinha & Priti Sinha *Publisher:* BPB Publications
- 2.Introduction to Computers *Author:* Peter Norton *Publisher:* McGraw-Hill Education
- 3.Data Communications and Networking *Author:* Behrouz A. Forouzan *Publisher:* McGraw-Hill
- 4.Fundamentals of Computer Networks *Author:* D. Black *Publisher:* Pearson
- 5.Operating Systems: Internals and Design Principles *Author:* William Stallings.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test Papers	5
Total		75

KU3VACCAP102: CYBER LAW AND ETHICS

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
3	VAC	100 - 199	KU3VACCAP102		3 (3T+ 0P)	45
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	-	-	25	50	75	1.5 Hrs

Course Description: This value-added course provides a foundational understanding of the digital world's legal, ethical, and security aspects. It is designed to enhance the digital literacy and legal awareness of students from all disciplines by introducing key concepts in cybersecurity, cybercrimes, IT laws, digital rights, and ethical responsibilities in cyberspace. Through real-world case studies, discussions, and interactive sessions, learners will gain insights into the evolving challenges in the cyber domain and be empowered to act as responsible digital citizens. The course aims to bridge the gap between technology use and legal-ethical awareness in today's increasingly connected world.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Explain the fundamental concepts of cybersecurity and various types of cybercrimes.	U
2	Identify and describe the key provisions of the IT Act and its relevance in cyberspace.	R,U
3	Analyze legal frameworks and compare national and international cyber laws.	U,An
4	Evaluate the ethical dimensions of online behavior and digital responsibility.	R, E
5	Apply ethical and legal principles in hypothetical cybercrime scenarios or case studies.	R,U,A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2			2			
CO 2	2			3			2
CO 3				3			2
CO 4	2			2			
CO 5	2			2			2

COURSE CONTENTS

Contents for Classroom Transaction:

MODULE	UNIT	DESCRIPTION	HOURS
1	Cyber Security and Crimes		8
	1	Introduction to Cyber Security	
	2	Cyber Threat Landscape: Viruses, Worms, Ransomware, Phishing, Social Engineering	
	3	Classification of Cyber Crimes:	
	4	Cyber Terrorism and National Security Threats	
	5	Investigation and Reporting of Cyber Crimes	

	6	Tools and Techniques for Cyber Security	
2	Cyber Law and IT ACT		8
	1	Cyberspace, Cyber Jurisprudence, Jurisprudence and Law	
	2	Genesis and Need for Cyber Law	
	3	Overview of the Information Technology Act, 2000	
	4	Amendments to the IT Act (2008 and beyond)	
	5	Legal Recognition of Electronic Documents and Digital Signatures	
	6	Certifying Authorities and Digital Certificates	
	7	Penalties and Adjudication under the IT Act	
	8	Role of CERT-In and other regulatory bodies	
3	Cyber Law and Legislation		10
	1	International Laws and Treaties Related to Cyber Space (Budapest Convention, GDPR)	
	2	Comparative Study of Cyber Laws: India vs. US/EU	
	3	Jurisdiction and Sovereignty in Cyber Space	
	4	Intellectual Property Rights in the Digital Environment	
	5	Data Protection and Privacy Laws (India's DPDP Act)	
	6	Role of the Judiciary in Cyber Law Interpretation	
		Case Studies: Landmark Judgments	
4	Cyber Ethics		10
	1	Definition and Importance of Cyber Ethics	
	2	Ethics vs. Law in Cyberspace	
	3	Common Unethical Practices: Plagiarism, Piracy, Hacking, Cyber-Bullying	
	4	Digital Citizenship and Responsible Internet Use	
	5	Social Media Ethics and Digital Footprint	
	6	Ethics in Artificial Intelligence and Data Analytics	
	7	Ethical Frameworks and Decision Making in IT	
5	Teacher Specific Module		9
	<i>Directions</i>		

Essential Readings:

1. Cyber Law: Simplified by Vivek Sood

- **Publisher:** Tata McGraw-Hill
- Covers cyber crimes, the IT Act, legal issues in e-commerce, and case law

2. Cyber Law & Cyber Crimes by Barkha & U. Rama Mohan

- **Publisher:** Asia Law House
- Detailed analysis of IT Act, cyber crimes, cyber forensics, and case studies.

3. Information Technology Law and Practice by Vakul Sharma

- **Publisher:** Universal Law Publishing
- Authoritative text on Indian cyber law with detailed references to IT Act, rules, and court decisions.

4. Cyber Laws by Krishna Kumar Yadav

- **Publisher:** Allahabad Law Agency
- Focuses on legal frameworks, cyber crimes, e-governance, and digital signatures.

5. Cyber Ethics: Morality and Law in Cyberspace by Richard Spinello

- **Publisher:** Jones & Bartlett Learning (International Edition)
- Covers ethical issues in internet use, privacy, piracy, AI, and social media ethics.

Supplementary Readings / References:

- **The Information Technology Act, 2000 (with amendments)** – Government of India
- **Budapest Convention on Cybercrime** – Council of Europe (for international context)
- **Digital Personal Data Protection Act, 2023** – (India's data protection framework)

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation		25
a)	Test Paper	12
b)	Assignment	5
c)	Seminar/Viva- Voce	8
Total		75

KU3VACCAP103: INTRODUCTION TO DATA ANALYTICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	VAC	100-199	KU3VACCAP103	3(3T+0P)	45

Learning Approach (Hours/Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/Internship	Tutorial	CE	ESE	Total	1.5 Hrs
3	-	-	25	50	75	

Course Description: Data analytics is a multidisciplinary field that employs various analysis techniques, including math, statistics, and computer science, to draw insights from data sets. Since Data is endless and huge, data has become the most important entity for all sectors such as Banking, Education, Health sector and Government etc., The course includes everything from simply analyzing data to theorizing ways of collecting data and creating the frameworks needed to store it.

Course Prerequisite: Nil

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
1	Familiarize fundamental concepts of data science and analytics	R
2	Identify various sources of data and explain methods for data collection, validation, and cleaning	U
3	Apply basic statistical methods and analyze simple business scenarios using data analytics techniques	A, An
4	Demonstrate an understanding of how data analytics supports decision-making in various industries	A

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	2						
CO2	2		2	2			
CO3							
CO4	2			3		2	

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE : Introduction to Data Science and Analytics		
	1	Data Science - Overview of Data Analytics: Types and Lifecycle	8
	2	Importance and Applications of Data Analytics in Industries (Healthcare, Retail, Finance, Manufacturing, etc.)	
	3	Role of a Data Analyst vs. Data Scientist. Ethical Considerations and Data Privacy	
	4	Types of Data: Structured vs. Unstructured. Sources of Data: Surveys, Sensors, Social Media, Databases, Web, Introduction to Big Data	
II	MODULE TITLE: Data Collection and Cleaning		
	1	Data Collection Methods (Manual, Online Forms, APIs, Public Datasets)	8
	2	Common Data Quality Issues	
	3	Techniques for Data Cleaning: Removing Duplicates, Handling Missing Values, Data Transformation	
	4	Introduction to Data Validation. Formatting and Organizing Data (Using Excel/Google Sheets)	
III	MODULE TITLE : Introduction to Statistics for Data Analysis		
	1	Basics of Statistics: Mean, Median, Mode, Variance, Standard Deviation	10
	2	Introduction to Probability. Correlation and Causation	

	3	Sampling Techniques and Data Distribution	
	4	Descriptive Statistics. Hypothesis Testing AB Testing	
IV	MODULE TITLE: Real-world Applications and Case Studies		
	1	Descriptive Analytics: Summarizing Data and Predictive Analytics: Forecasting Trends (Conceptual Overview)	10
	2	Business Use Cases in Different Industries	
	3	How Companies Use Data to Drive Decisions	
	4	Case Studies in Marketing, Healthcare, Finance, etc.	
V	MODULE TITLE: TEACHER SPECIFIC MODULE		
			9

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

1. Data Analytics by V. Rajaraman & C.S. Ananda, PHI Learning
2. Fundamentals of Data Science by S. K. Gupta, Khanna Publishing
3. Data Science and Analytics by V.K. Jain. Khanna Publishing
4. Business Analytics by U. Dinesh Kumar, Wiley India

Assessment Rubrics:

Evaluation Type	Marks
End Semester Evaluation (ESE) Total	50
a) Theory	50
CCA Total	25
a) Test Papers	12
b) Assignment	5
c) Viva/Seminar	8
TOTAL	75

KU4VACCAP104: BASICS OF COMPUTER NETWORKS

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
4	VAC	100-199	KU4VACCAP104		3(3T+0P)	45
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	--	--	25	50	75	1.5hrs

Course Description: This value-added course introduces the foundational concepts of computer networks in a simplified and practical manner, suitable for students from any academic background. The course covers the basics of how computers communicate, the role of networking devices, transmission media, common internet protocols, and practical tools used in daily networking tasks. The hands-on component is designed to build familiarity with real-world networking tools and applications, empowering students with essential digital skills applicable in various career paths.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Recall basic concepts and terminology related to computer networks	R
2	Explain different types of networks, transmission media, and networking devices	R, U
3	Identify and describe the functions of common networking protocols and applications	U
4	Use basic networking commands and tools to check connectivity and configurations	R, A
5	Set up a simple peer-to-peer network and perform basic troubleshooting	A, E, C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2			2			
CO 2	2			3			2
CO 3	2			2			
CO 4	2					2	
CO 5	2	2				2	

COURSE CONTENTS

Contents for Classroom Transaction:

MODULE	UNIT	DESCRIPTION	HOURS
1	Introduction to Networking		8
	1	Definition and importance of computer networks	
	2	Types of networks: LAN, MAN, WAN, PAN	
	3	Network topologies: Star, Bus, Ring, Mesh, Hybrid	
	4	Client-server vs. peer-to-peer models	
	5	Basic concepts of IP addressing and DNS	
2	Transmission Media and Network Devices		8
	1	Transmission Media Guided media: Twisted pair, Coaxial cable, Optical fiber	
		2	
	3		
	4	Firewall and Gateway (basic concepts) Wired vs Wireless networks (basic comparison)	
		Common Protocols and Applications	
1	Common networking protocols: TCP/IP, UDP, HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS, DHCP		
	2		

	3	Concept of Ports and Sockets	
	4	Introduction to Internet and Intranet	
	5	Web browsing, Email communication	
		File transfer and remote login basics (FTP, SSH)	
4	Networking Tools and Hands-on Lab		10
	1	Basic Networking Commands:	
	2	ping, ipconfig / ifconfig, tracert / traceroute, netstat, nslookup	
	3	Wireshark (basic packet capture and analysis)	
	4	Introduction to Packet Tracer or similar simulators	
	5	LAN setup basics using switches and routers	
	6	Activities: Set up a simple peer-to-peer network using two computers and file sharing Use ping, tracert, ipconfig, and nslookup commands to troubleshoot network issues. Simulate a network topology in Cisco Packet Tracer (e.g., star network with 3 clients)	
5	Teacher Specific Module		9

Essential Readings:

1. Computer Networking: A Top-Down Approach

- **Authors:** James F. Kurose, Keith W. Ross
- **Edition:** 8th Edition (latest available)
- **Publisher:** Pearson

2. Data Communications and Networking

- **Author:** Behrouz A. Forouzan
- **Edition:** 5th Edition
- **Publisher:** McGraw Hill

3. Computer Networks

- **Author:** Andrew S. Tanenbaum, David J. Wetherall
- **Edition:** 5th Edition
- **Publisher:** Pearson

4. Networking All-in-One For Dummies

- **Author:** Doug Lowe
- **Edition:** Updated regularly (check for latest)
- **Publisher:** Wiley

Supplementary Reading (for lab and practical focus):

1. Introduction to Networking Basics

- **Author:** Patrick Ciccarelli, Christina Faulkner
- **Publisher:** Wiley
- **Focus:** Basic concepts + hands-on exercises

2. Cisco Networking Academy Lab Manuals

- Can be used if you're using **Packet Tracer** in your lab sessions
- Available for free with Cisco Networking Academy account

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation		25
a)	Test Papers	12
b)	Assignment	5
c)	Seminar/Viva- Voce	8
Total		75

KU4VACCAP105: BASICS OF INTERNET OF THINGS (IoT)

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
4	VAC	100 - 199	KU4VACCAP105		3 (3T+0P)	45
Learning Approach (Hours/ Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	-	-	25	50	75	1.5 hrs

Course Description: This value-added course provides a foundational understanding of the Internet of Things (IoT), a transformative technology driving innovation across industries. The course is designed to equip students with essential knowledge and practical skills in IoT systems, covering key components such as sensors, actuators, microcontrollers, communication protocols, cloud platforms, and security practices. Emphasis is also placed on real-world applications in smart homes, healthcare, and industrial automation, along with discussions on ethical considerations and security challenges in connected environments.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Define the fundamental concepts, evolution, and applications of IoT.	R
2	Explain the roles of key components in IoT systems such as sensors, actuators, and connectivity tools.	U
3	Demonstrate using microcontrollers like Arduino and Raspberry Pi in building simple IoT projects.	A
4	Analyze different IoT communication protocols and architectures (Edge vs. Cloud computing).	An
5	Evaluate security risks in IoT systems and recommend mitigation strategies.	E

6	Design a basic IoT prototype integrating sensors, connectivity, and data visualization tool.	C
---	--	---

**Remember , Understand (U), Apply (A), Analyse (An), Evaluate , Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2						
CO 2	2						
CO 3			2				
CO 4	2			3			
CO 5	2						2
CO 6	2	3				2	

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
1	Introduction to IoT		8
	1 2 3 4	Introduction: Definition and basic concepts of IoT, Historical development of IoT, Benefits and challenges of IoT Key Components of IoT: Sensors and actuators, Gateways and cloud platforms, Connectivity technologies: Wi-Fi, Bluetooth, Zigbee, etc., IoT platforms. IoT Architecture: Edge computing vs. cloud computing, Communication protocols: MQTT, CoAP, HTTP. Applications of IoT : Smart homes, Industrial IoT (IIoT), Healthcare applications	
2	IoT Devices and Sensors		

	1	Types of IoT Devices: Wearables, Smart appliances, Embedded systems	
	2	Sensors and Actuators: Types of sensors: temperature, humidity, motion, etc., Interfacing sensors with microcontrollers	
	3	IoT Prototyping: Basics of Arduino and Raspberry Pi, Building simple IoT projects	
3	Communication and Networking		
	1	Wireless Communication Protocols for IoT: Wi-Fi, Bluetooth, Zigbee, LoRaWAN, Suitability of protocols for various scenarios	8
	2	Cloud Platforms for IoT: Introduction to AWS IoT Core, Microsoft Azure IoT Hub, Google Cloud IoT Core, Uploading and visualizing sensor data	
4	Security in IoT		10
	1	Vulnerabilities and Threats in IoT Systems: Data breaches, Hacking, Privacy concerns	
	2	Security Strategies: Encryption, Authentication, Access Control	
	3	Ethical Considerations: Data privacy, Bias in IoT systems, Responsible technology use	
5	Teacher Specific Module		9
	<i>Directions</i>		

Essential Readings:

1. Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti.
2. Building the Internet of Things by Maciej Kranz.
3. Getting Started with Raspberry Pi by Matt Richardson and Shawn Wallace.
4. IoT Projects with Arduino by Emily Friedel and Terry Martin.
5. Fundamentals of IoT Communication Technologies by Rolando Herrero.
6. Designing Connected Products by Claire Rowland.
7. Practical Internet of Things Security By Brian Russell and Drew Van Duren.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation		25
a)	Test Paper	12
b)	Assignment	5
c)	Seminar/Viva- Voce	8
Total		75

KU4VACCAP106: R PROGRAMMING FOR DATA ANALYTICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	VAC	100-199	KU4VACCAP106	3 (2T+ 1 P)	60

Learning Approach (Hours/ Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	-	25	50	75	1.5 Hrs

Course Description: This value-added course is designed to introduce students to the fundamentals of R programming and its powerful capabilities in data analytics. R is a widely-used, open-source programming language specially tailored for statistical computing and data visualization. Through a hands-on approach, students will gain practical experience in data import, cleaning, manipulation, visualization, and performing basic statistical analysis. This course is ideal for beginners, it lays the foundation for advanced learning in areas like machine learning, data science, and business analytics

Course Prerequisite: Introduction to Data Analytics and Basic Programming Skill

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
1	Apply basic R programming skills to load, explore, and manipulate datasets for data analysis tasks.	A
2	Use R packages to clean, transform, and manage data efficiently.	A,An
3	Create a variety of data visualizations (e.g., bar charts, histograms, scatter plots)	A, An, C
4	Perform basic statistical analysis and interpret results	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSO

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	2		2				
CO2	2		2	2			
CO3		2					2
CO4	2	2				2	

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE : Introduction to R and RStudio		
	1	Overview of R and its applications in data analytics	7
	2	Installing R and Rstudio and Understanding the RStudio interface	
	3	Writing and running basic R scripts	
	4	Data types, variables, and basic operations	
II	MODULE TITLE : Data Handling and Manipulation		
	1	Importing data (CSV, Excel, etc.)	8
	2	Data frames, vectors, lists, and matrices	
	3	Data cleaning: handling missing values, renaming columns, Subsetting and filtering data	
	4	Introduction to the dplyr package for data manipulation	
III	MODULE TITLE: Data Visualization in R		
	1	Basic plotting with plot (), barplot(), hist()	8
	2	Introduction to the ggplot2 package	
	3	Creating bar charts, histograms, line charts, and scatter plots. Customizing plots (labels, colors, themes)	
	4	Exporting and saving graphs	
IV	MODULE TITLE: Basic Statistical Analysis		
	1	Descriptive statistics: mean, median, mode, standard deviation	7
	2	Data summarization with summary () and aggregate()	
	3	Correlation and simple linear regression, Basic hypothesis testing (t-test, chi-square test)	
	4	Introduction to the caret package for basic modeling	
V	LAB EXPERIMENTS		

	1. Write a script to assign values to two variables and print the results of arithmetic operations of these two variables. 2. Create variables of different data types and check the data types using the class() function 3. Write a script to import a csv and excel file(using readxl library) and print the data 4. Create and explore vectors, lists, matrices, and data frames. 5. A R script to handle missing values, rename columns, and filter data. 6. Use dplyr for summarizing data. 7. Use plot(), barplot(), and hist() to create simple visualizations. 8. Create a basic ggplot2 plot using a sample dataset and customize plot elements like colors, labels, themes, and points. Save plots as image files (e.g., PNG, PDF). 9. Compute basic statistics: mean, median, mode, standard deviation. Summarize datasets using built-in functions. 10. Explore relationships between variables using correlation and regression. Perform t-test and chi-square test.	30
--	---	----

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

1. Available Online: <https://r4ds.had.co.nz> (Free and official online version)
2. *Data Analytics Using R* by Seema Acharya, McGraw Hill Education

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test Papers	5
Total		75

SKILL ENHANCEMENT COURSES (SEC)

KU4SECCAP101: SOFTWARE PROJECT MANAGEMENT

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
4	SEC	100-199	KU4SECCAP101		3(3T+0P)	45
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	0	0	25	50	75	1.5 Hrs

Course Description: This course introduces students to the principles, methods, and best practices of software project management. It emphasizes the application of project management processes in the context of software development. Students will learn to plan, schedule, budget, and monitor software projects, along with managing risks, quality, and team dynamics. The course integrates both traditional and modern methodologies, including Agile, to prepare students for real-world software project challenges. Tools like Gantt charts, PERT, and project tracking software are introduced.

Course Prerequisite: C Programming, Java or C++ Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of project management in the context of software development.	U
2	Apply project planning techniques, including WBS, network diagrams, and scheduling.	A
3	Estimate project resources, costs, and timelines using standard models.	A
4	Identify and manage project risks, communication plans, and procurement processes.	U
5	Use tools and techniques to manage project scope, quality, and performance.	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2			2			2
CO 2	2			2			
CO 3		2		2			
CO 4	2						
CO 5		2		2			

COURSE CONTENTS

Contents for Classroom Transaction:

MODULE	UNIT	DESCRIPTION	HOURS
1	MODULE TITLE -Introduction to Project Management		8
	1	Understanding Projects and Project Management a) What is a Project?, Attributes of a project, Project Constraints b) What is project Management? c) Role of the Project manager,Suggested Skills for Project Managers	
	2	Project Life Cycle and SDLC a) Project and Product Lifecycles b) The Context of Information Technology Projects c) Recent Trends Affecting Information Technology Project Management	
	3	Project Management Process Groups a) Five Project Management Process Groups b) Mapping the Process Groups to the Knowledge Areas c) Developing an IT Project Management Methodology	

2	MODULE TITLE- Project Scope and Schedule Management		8
	1	Project scope Management a) What is Project Scope Management? b) Planning Scope Management c) Collecting Requirements d) Defining Scope	
	2	Project scope Management Contd... a) Creating the Work Breakdown Structure b) The WBS Dictionary c) Validating Scope d) Controlling Scope	
	3	Project Schedule Management a) The Importance of Project Schedules a) Planning Schedule Management b) Defining Activities c) Sequencing Activities	
	4	Project Schedule Management contd.... a) Estimating Activity Durations b) Developing the Schedule- Gantt Charts, Critical Path Method c) Program Evaluation and Review Technique (PERT) d) Controlling the Schedule	
3	MODULE TITLE- The Project Cost and Quality Management		10
	1	Project Cost Management a) The Importance of Project Cost Management b) Basic Principles of Cost Management c) Planning Cost Management d) Estimating Costs e) Determining the Budget f) Controlling Costs	
	2	Project Quality Management a) What is Project Quality Management? b) Planning Quality Management c) Managing Quality d) Controlling Quality e) Tools and Techniques for Quality Control f) ISO standards and Six Sigma Basics g) Improving It Project Quality-Maturity Models,CMMI	

4	MODULE TITLE- Managing Project Changes, Risk, People and Communication		10
	1	*Managing Changes a) Managing changes in Traditional and Agile methods b) Configuration Management	
	2	*Project Risk Management a) Risk Management Process b) Define Standards, Identify Risks, Some Common Risks in Software Development, Classify Risks c) Risk Management Strategies, Budgeting for Risks d) Risk Monitoring and Control	
	3	*Managing people and organising communication a) Managing people b) Project Organisation Structures c) Managing Communication	
	4	Open Source Tools for Managing Projects	
5	Teacher Specific Module		9
	<i>Familiarise Tools for Software Project Management</i>		
	<i>Emerging Trends in Software Project management</i>		

Essential Readings:

1. Kathy Schwalbe, Information Technology Project Management, Cengage Learning, Inc., 9th Edition, Student Edition: ISBN-13: 978-1-337-10135-6.
2. Adolfo Villafiorita, Introduction to Software Project Management, CRC Press Taylor & Francis Group, ISBN-13: 978-1-4665-5954-7 (eBook - PDF) (For topics prefixed with * only)

References:

Books:

1. Bob Hughes, Mike Cotterell and Rajib Mall, Software Project Management, McGraw Hill Education

2. Andrew Stellman , Jennifer Greene, Applied software Project management, O'Reilly Media**Web:**
1. PMI (Project Management Institute)-<https://www.pmi.org>
2. OpenProject(Free Project Management Tool)-<https://www.openproject.org>
3. ProjectLibre(Free alternative to MS Project) -<https://www.projectlibre.com>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study presentation	8
Total		75

KU4SECCAP102: ARTIFICIAL INTELLIGENCE (AI) ASSISTED TOOLS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours	
4	SEC	100-199	KU4SECCAP102	3(2T+1P)	60	
Learning Approach (Hours/ Week)			Marks Distribution		Duration of ESE (Hours)	
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	-	25	50	75	1.5 Hrs

Course Description:

This Skill Enhancement Course introduces learners to the foundational concepts and practical applications of Artificial Intelligence (AI) tools across diverse domains. The course offers a hands-on learning experience with popular AI-powered platforms used for text generation, image design, audio/video creation, and productivity enhancement. Special focus is given to prompt engineering, enabling learners to interact effectively with AI systems to generate accurate and creative outputs.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of AI and categories of AI tools.	U
2	Describe the applications of text-based, image-based, audio, video, and productivity AI tools in real-world contexts.	R,U
3	Demonstrate using popular AI tools for content creation, communication, and task automation.	U, A
4	Analyze the differences between traditional and AI-assisted approaches in content generation and productivity enhancement.	R,An
5	Evaluate the ethical implications and potential limitations of using AI tools in professional and academic settings.	R,An,E

6	Construct effective prompts to optimize outputs from text, image, and multimedia-based AI tools.	R, C
---	--	------

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2						2
CO 2	2			2			
CO 3	2		2				
CO 4	2				2		2
CO 5	3			2	2		
CO 6	3	2		2			

COURSE CONTENTS

Contents for Classroom Transaction:

MODUL E	UNI T	DESCRIPTION	HOURS
1	Introduction to AI and AI Tools		7
	1	Introduction to Artificial Intelligence.	
	2	History and evolution of AI	
	3	Categories of AI: Narrow AI, General AI, and Super AI	
	4	Overview of AI applications in various sectors	
	5	Types of AI tools: Text, Image, Audio, Video, Productivity, etc.	
	6	Ethical implications and responsible AI usage	
	7	Activity: Explore and report on 3 different free AI tools from various categories.	
2	Exploring Text-Based and Image Design-Based AI Tools		8

	1 2 3 4 5	Text generation tools: ChatGPT, Gemini, Jasper, Writesonic Image generation/design tools: Canva AI, DALL·E, Adobe Firefly, Midjourney Prompting basics for text/image AI tools Comparing traditional vs AI-assisted content creation Activities: Text Tool Task: Use ChatGPT to write a short article or summary on a given topic. Image Tool Task: Design a poster or social media graphic using Canva AI or DALL·E	
3	Audio, Video, and Productivity Tools		8
	1 2 3 4 5	Audio tools: Descript, Murf.ai, ElevenLabs. Video tools: Pictory, Synthesia, Runway Productivity tools: Notion AI, Grammarly, Otter.ai, Microsoft Copilot Use cases in education, business, and content creation Activities: Create Audio/Video: Generate a voiceover or short AI video for a presentation. Productivity Challenge: Use AI tools like Notion AI or Grammarly for document enhancement and planning.	
4	Introduction to Prompt Engineering		7
	1 2 3 4 5	Introduction to the prompt, Role of prompts in AI tools Anatomy of a good prompt Types of prompts: Instructional, conversational, role-based Prompt tuning: Improving results through iteration Use in text, image, and code generation Activities: Prompt Lab: Experiment with different prompt styles in ChatGPT to produce creative writing or code. Role-based Prompting: Simulate expert roles (e.g., historian, teacher) using prompt scenarios.	
5	Teacher Specific Module		30
	Sample Lab Exercise: 1. Generate short-form content using an AI text tool on any topic.		

	2. Create visual content using Canva AI or DALL·E for the above topic 3. Generate a voiceover using an AI audio tool, Murf.ai, Descript, or ElevenLabs. 4. Use AI tools to create a simple explainer video using Pictory, Heygen or Synthesia. 5. Use AI productivity tools to write and organize ideas, using Notion AI, Grammarly, or Microsoft Copilot. 6. Write effective prompts (minimum three different prompts) for AI tools and observe the outputs. Demonstrate for poor and better prompting. Test the three types of prompts: Instructional, Conversational and Role-Based	
--	--	--

Essential Readings:

1. Artificial Intelligence Basics

- Title:** *Artificial Intelligence: A Guide for Thinking Humans*
Author: Melanie Mitchell
Publisher: Penguin
Description: Offers a clear introduction to core AI concepts with real-world implications.
Level: Beginner to Intermediate
- Title:** *Artificial Intelligence: A New Synthesis*
Author: Nils J. Nilsson
Publisher: Morgan Kaufmann
Description: A foundational textbook covering traditional and modern AI.
Level: Intermediate

2. AI Tools and Applications

- Title:** *The Art of Prompt Engineering with ChatGPT: Crafting Effective Prompts for Work, Creativity, and Learning*
Author: Nathan Hunter
Publisher: Independently published
Description: Focuses on prompt crafting strategies for getting optimal results from text-based AI tools.
Level: Beginner to Intermediate

3. Multimedia and Productivity Tools

- Title:** *AI for Creators: A Guide to AI Tools for Content Creation*
Author: Chris Lu
Publisher: Self-published
Description: Hands-on guide to tools like Midjourney, Descript, Synthesia, and

others for creative work.

Level: Beginner

- **Title:** *AI and You: How to Think, Create and Collaborate with AI*

Author: Dan Fitzpatrick

Publisher: Routledge

Description: Includes practical examples and tool-based workflows for content creators and professionals.

Level: Beginner

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test Papers	5
Total		75

KU4SECCAP103: OPERATING SYSTEM ADMINISTRATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	SEC	100-199	KU4SECCAP103	3 (2T+ 1P)	60

Learning Approach (Hours/Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	1.5 Hrs
2	2	-	25	50	75	

Course Description: This course is designed to provide students with comprehensive, hands-on knowledge of operating system management using two of the most widely used platforms: **Linux** and **Windows**. Students can gain critical skills needed to install, configure, maintain, and troubleshoot both Linux and Windows operating systems in standalone and networked environments. It equips participants with practical expertise and will develop a dual-platform skill for a wide range of roles in IT support, system administration, and technical operations especially for those who pursue a career in system administration.

Course Prerequisite: KU1DSCCAP101

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
1	Demonstrate the ability to install and configure Linux and Windows operating systems	A
2	Effectively manage users, groups, and file permissions	U
3	Monitor and manage system processes, services, and scheduled tasks	A, An
4	Configure network settings and apply basic system security measures,	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	2						2
CO2	2						
CO3	2						
CO4	2	2		2			2

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE: Operating System Fundamentals and Installation		
	1	Overview of OS types: Windows vs. Linux	7
	2	Linux : Installing Linux distributions (Ubuntu, CentOS, etc.), Dual boot configuration, Understanding file system hierarchy, Basic shell usage and terminal commands, Configuring system settings (display, power timezone, etc.)	
	3	Windows: Installing Windows OS (Home, Pro, Server editions), Disk partitioning and formatting, Device Manager and driver installation, Configuring Windows settings (Control Panel, System Properties), Introduction to PowerShell	
II	MODULE TITLE : User and File System Management		
	1	Linux : Creating and managing user accounts and groups, File and directory permissions (chmod, chown)	8
	2	Linux: Sudoers configuration and access control, Shell environment familiarising	
	3	Windows: Managing user accounts and groups (Local Users and Groups MMC)	
	4	Windows: NTFS permissions and inheritance, User Account Control (UAC), Group Policy basics	
	MODULE TITLE: Process, Task, and Service Management		

	1	Linux: Managing processes (ps,top, kill, nice), Running background and foreground tasks	8
	2	Linux: Managing services with systemd and service, Cron jobs and scheduling tasks	
	3	Windows: Task Manager and Process Explorer, Managing services (services.msc)	
	4	Windows: Task Scheduler, Windows Event Viewer for diagnostics	
IV	MODULE TITLE: Network Configuration and Security		
	1	Linux: IP configuration and netplan/nmcli, Firewall configuration using ufw/iptables	7
	2	Linux: SSH setup and remote access, Basic network troubleshooting (ping, netstat, traceroute)	
	3	Windows: IP settings and adapter configuration, Windows Defender Firewall rules	
	4	Windows: Remote Desktop configuration, Network troubleshooting tools (ipconfig, netstat, ping, tracert)	
V	LAB EXPERIMENTS		
		1. Install Linux and Windows and implement dual booting. Choose manual partitioning in Linux, configure Disk partitioning (NTFS format) in Windows. 2. Run and observe the output of these commands: whoami, uname -a, df -h, mkdir sample, touch testfile.txt, mv testfile.txt sample/, rm -r testfolder 3. Run the following commands and observe the output: Get-ComputerInfo, Get-Process, Get-Service,Get-Disk 4. Create a test user in Linux and Windows operating systems and add the user to the admin group and check the privileges. 5. Demonstrate ps and cron commands with various options in Linux and task manager and task scheduler in Windows 6. Configure network in Linux and Windows environments and demonstrate remote access.	30

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

- Windows Administration Fundamentals by Ravikanth Chaganti
- Open Source For You -journal

- Yashavant Kanetkar, UNIX Shell Programming, BPB
- <https://learn.microsoft.com/en-us/windows/win32>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test Papers	5
Total		75

KU5SECCAP104 : DIGITAL MARKETING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	SEC	100-199	KU5SECCAP104	3(2+1)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2		25	50	75	1 Hr 30 Minutes

Course Description: This course introduces the fundamental concepts and techniques of digital marketing in today's fast-growing digital environment. Students will explore key channels such as SEO, social media, email, and content marketing, while gaining basic practical skills to create and manage effective online marketing campaigns.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of digital marketing, social media marketing, and content marketing strategies.	U
2	Create and manage engaging digital content for social media platforms to improve online audience interaction and brand visibility.	C
3	Apply Search Engine Optimization (SEO) techniques and paid advertising methods to increase website traffic and online reach.	A
4	Design and implement email marketing campaigns using appropriate tools and performance measurement techniques.	C
5	Analyze digital marketing campaign performance using web analytics and reporting tools for effective decision-making.	An

Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)

Mapping of COs with PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	1	1	2	1	2	2
CO2	3	2	1	2	1	2	2
CO3	3	2	2	3	1	2	3
CO4	3	2	2	3	1	2	2
CO5	3	2	3	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Basics of Digital Marketing		7
	1	Overview of digital marketing, Importance of digital marketing for businesses	
	2	Terminologies, 10Ps of digital marketing, Landscape/digital marketing channels: Overview	
	3	Basics of creating a digital marketing strategy, Setting SMART Goals in Digital Marketing	
	4	Understanding Customer Behavior in the Digital Age	
2	Social Media Marketing & Content Marketing		8
	1	Introduction to Social Media Marketing, Importance of Social Media in Business, Popular Social Media Platforms: Facebook, Instagram, LinkedIn, YouTube, Types of Social Media Content, Benefits and Challenges of Social Media Marketing, Social Media Marketing Strategies, Building Brand Communities on Key Platforms	
	2	Creating Business Profiles on Social Media, Basics of Social Media Campaigns, Social Media Branding Basics, Introduction to Influencer Marketing, Paid Promotions on Social Media	
	3	Types of Digital Content: Text content, Images, Videos, Infographics, Blogs	

		Basics of Content Creation, Writing Simple and Engaging Content, Storytelling in Digital Marketing Content Planning for Improved Ranking, Content Marketing Platforms and Tools	
	4	Engaging Content Creation for Social Media Channels, Introduction to Social Media Analytics. Understanding Likes, Shares, Comments, and Reach, Measuring Engagement Rate	
3	SEO and Paid Advertising		8
	1	Introduction to Search Engine Optimization (SEO), Importance of SEO in Online Marketing, Introduction to Paid Advertising, Difference between Organic Search and Paid Search, Overview of Search Engines and Search Results	
	2	Search Engine Optimization (SEO) Principles - Optimizing Websites for Search Visibility, Content Optimization for SEO, Image Optimization Basics, Mobile-Friendly Website Design, Website Speed and Performance Basics, Internal Linking and Navigation	
	3	How Search Engines Work, Keywords and Keyword Research Basics, Types of Keywords: Short-tail keywords, Long-tail keywords, On-Page SEO Basics: Title tags, Meta descriptions, Headings, URL structure Introduction to Off-Page SEO, Importance of Backlinks	
	4	Paid Advertising Fundamentals, Types of Paid Ads: Search ads, Display ads, Social media ads and Video ads, Introduction to Pay-Per-Click (PPC) Advertising	
	5	Introduction to Google Ads, Introduction to Meta Ads, Creating Simple Search Advertisements, Creating Social Media Advertisements	
	6	Introduction to Website Traffic Analysis, Measuring Advertisement, Performance, Understanding Clicks, Impressions, and CTR, Conversion Tracking Basics	
4	Email Marketing		7
	1	Types of Marketing Emails: Promotional emails, Newsletter emails, Welcome emails, Reminder emails	
	2	Effective Email Marketing Strategies - Building Email Lists and Segmentation Creating Marketing Emails: Structure of a Marketing Email, Writing Effective Subject Lines, Creating Simple Email Content, Adding Images, Buttons, and Links, Designing Responsive Email Templates Call-to-Action (CTA) in Emails, Personalization in Email Marketing	
	3	Introduction to Email Analytics: Understanding Open Rate, Understanding Click-Through Rate (CTR), Bounce Rate and Unsubscribe Rate, Improving Email Performance	

	4	Email Marketing Automation Tools: Mailchimp, Brevo, MailerLite Professional Email Writing Etiquette, Best Practices for Effective Campaigns	
5	Teacher Specific Module :		30
	Directions		
	1. Exercise 1: Keyword Research (SEO Basics) a. Aim: To identify relevant keywords for a product/service. b. Tools Required: Web browser (Google search) c. Procedure: i. Choose a product (e.g., handmade soaps, backpacks). ii. Think of words a customer may type in Google. iii. List at least 10 keywords.iv. Check related searches on Google for more ideas. d. Output: List of 10–15 relevant keywords. 2. Exercise 2: Social Media Post Creation a. Aim: To design a basic promotional social media post. b. Tools Required: Canva/any design tool c. Procedure: i. Select a business (e.g., bakery, clothing brand). ii. Create a caption for promotion. iii. Add 5–8 relevant hashtags.iv. Design or describe an image for the post. d. Output: One complete social media post. 3. Exercise 3: Content Writing (Blog) a. Aim: To create simple marketing content. b. Tools Required: Text editor/Blog site c. Procedure: i. Choose a topic (e.g., “Newly launched Sunscreen”, ”Newly launched online Tutorial App”). ii. Write 80-100 words. iii. Use simple language and a clear structure (intro, body, conclusion). d. Output: : A short blog/article. 4. Exercise 4: Email Marketing Draft a. Aim: To design a promotional email. b. Tools Required: Text editor/Email c. Procedure: i. Choose a product or offer. ii. Write a subject line. iii. Add greeting and message content.iv. Include a call-to-action (e.g., “Buy Now”, ”Shop Now”, ”Join Now” etc.,). d. Output: : One sample marketing email. 5. Exercise 5: Competitor Analysis a. Aim: To study digital marketing strategies of competitors. b. Tools Required: Web browser c. Procedure: i. Visit 2 company websites or social media pages. ii. Observe their content, offers, and posting style. iii. Note key differences.		

	d. Output: Short comparison report (5–6 points). 6. Exercise 6: Digital Campaign Planning a. Aim: To design a simple digital marketing campaign. b. Tools Required: Notebook/Text editor c. Procedure: - Select a product/service. - Identify target audience - Choose a platform (Instagram, Facebook, Whatsapp, Email, etc.). - Suggest one campaign idea. d. Output: Basic campaign plan. 7. Exercise 7: Hashtag Research a. Aim: To identify effective hashtags. b. Tools Required: Instagram/Google c. Procedure: - Choose a topic (fitness, food, fashion). - Search for trending hashtags. - List at least 10 hashtags. d. Output: List of hashtags.	
--	---	--

Essential Readings:

- Digital Marketing 2026 by Seema Gupta | 3rd Edition, Mc Graw hill
- Fundamentals of Digital Marketing, Puneet Singh Bhatia, Pearson Education
- Digital Marketing For Dummies
- Digital Marketing: Tools, Techniques, and Trends

Suggested Readings:

- Digital Marketing Strategy, Implementation and Practice Dave Chaffey, Fiona Ellis-Chadwick, Publisher Pearson, 8th Edition
- https://learndigital.withgoogle.com/digitalgarage?utm_source=chatgpt.com
- https://academy.hubspot.com/?utm_source=chatgpt.com

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	12

b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

- **Employability for the Course:** Completing a digital marketing course prepares a student for a wide array of specialized career paths: SEO Specialist, Social Media Manager, Performance Marketer / PPC Analyst, Content Strategist and Digital Analyst etc.,

KU5SECCAP105: GAME APPLICATION DEVELOPMENT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	SEC	100-199	KU5SECCAP105	3(2+1)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	25	50	75	1 Hr 30 Minutes

Course Description: *This course introduces the fundamentals of game development using the Unity platform and C# programming. Students learn game design concepts, gameplay programming, user interaction, physics, animation, audio, and UI development for creating interactive games. The course also covers testing, debugging, optimization, and deployment of game applications through practical hands-on projects.*

Course Prerequisite: Programming skill

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of game development using the Unity engine, including game architecture, interface, scenes, assets, and development workflow.	U
2	Develop gameplay mechanics and interactive features using C# programming concepts such as scripting, object-oriented programming, events, and user input handling.	A C
3	Apply physics, animation, audio, and UI systems in game projects to create engaging and realistic gaming experiences within the Unity environment.	A C
4	Design, test, debug, and deploy complete game applications by following basic game design principles, performance optimization techniques, and multi-platform deployment methods.	An C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	3	1	2	1	1	3
CO2	3	3	2	2	1	1	3
CO3	3	3	2	2	2	2	3
CO4	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Introduction to Game Development and Unity Theory		7
	1	Introduction to Game Development Industry and Game Types (Overview of game industry, Types of games: 2D, 3D, Mobile, PC, Web)	
	2	Game Development Pipeline and Game Engines (Stages of game development, Introduction to game engines, Role of Unity)	
	3	Unity Environment and Interface (Unity Hub and Editor, Workspace and project setup, Folder structure)	
	4	Core Unity Components (Scenes, GameObjects, Components, Prefabs)	
	5	Asset Management in Unity (Asset store and Asset importing, Package Manager basics)	
2	Basics of Gameplay Programming with C#		8
	1	Introduction to C# Programming (Variables, Data types, Operators, Conditional statements)	
	2	Functions, Classes, and Scripting Concepts (Functions, Classes, Script structure)	
	3	MonoBehaviour Lifecycle and Event Handling (Awake(), Start(),	

		Update(), FixedUpdate())	
	4	Input Handling and Object Manipulation (Keyboard input, Transform manipulation, Basic movement)	
	5	Collision Detection and Gameplay Interaction (Collisions, Triggers, Simple interactions)	
3	Physics, UI, Animation, and Audio Systems		8
	1	Unity Physics System (Rigidbody, Collider, Gravity, Forces)	
	2	Introduction to Animation (Animation basics, Animator concepts)	
		User Interface Design (Canvas, Buttons, Text, Images, Panels)	
	3	Scene Management (Multiple scenes, Scene transitions)	
	4	Audio Systems (Audio Sources, Sound effects, Background audio)	
4	Game Design, Testing, and Deployment		7
	1	Fundamentals of Game Design (Game mechanics, Level design, Player experience)	
	2	Player Feedback and Game Feel (Visual feedback, Sound feedback, Interaction design)	
	3	Testing and Debugging (Bug identification, Debugging tools)	
	4	Build and Deployment (Build settings, Exporting for PC/Web/Mobile)	
5	Teacher Specific Module : LAB Exercise		30
	<i>Directions</i>		
	1) Unity Environment and Scene Creation 2) Basic Gameplay Programming 3) Physics, UI, and Audio Integration 4) Mini Game Prototype Development		

Essential Readings:

1. Introduction to Game Design, Prototyping, and Development — Jeremy Gibson Bond
2. Unity Game Development in 24 Hours — Mike Geig
3. Unity in Action — Joseph Hocking
4. Learning C# by Developing Games with Unity — Harrison Ferrone
5. [Unity Documentation](#) - (Online Resource)
6. [Unity Learn](#) - (Online Resource)

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Case study	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

- **Employability for the Course:** Game Developer

KU5SECCAP106 : FEATURE ENGINEERING USING PYTHON

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	SEC	100-199	KU5SECCAP106	3 (2T+ 1P)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2		25	50	75	1 Hr 30 Minutes

Course Description: This course focuses on transforming raw data into meaningful features through preprocessing and transformation techniques. It covers handling missing data, scaling, encoding, feature creation, and feature selection, along with dimensionality reduction using PCA.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand and classify data and feature types relevant to preprocessing and analysis.	U
2	Apply data preprocessing techniques including handling missing values, data cleaning, scaling, and normalization.	A
3	Construct and transform features using methods such as binning, encoding, and polynomial feature creation.	A
4	Evaluate and select relevant features using appropriate feature selection techniques.	An
5	Perform dimensionality reduction using Principal Component Analysis (PCA) and implement feature engineering workflows using Scikit-learn.	A

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	3	2	2	1	2
CO2	3	3	3	2	2	1	2
CO3	3	3	3	2	2	1	2
CO4	3	2	3	3	2	1	2
CO5	3	3	3	3	3	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Foundations of Data and Features		7
	1	Introduction to data and features in machine learning, Importance of feature engineering	
	2	Data types and features: Numerical, Categorical, Ordinal, Discrete, Continuous, Interval and Ratio, Text, Image	
	3	Basic Feature Preprocessing: Handling Missing Data, Data Cleaning, Feature Scaling, Normalization, and Transformation.	
	4	Text preprocessing : Tokenization, stop-word removal, basic representation : TF-IDF (Basic idea)	
	5	Image preprocessing: resizing, normalization, and basic data augmentation (flipping, rotation, cropping, and translation) (Basic idea)	
2	Module TITLE: Introduction to Python		9
	1	Dataset : Kaggle, UCI Machine Learning Repository, csv	
	2	NumPy, Array creation and indexing, Basic operations (mean, sum, reshape)	

	3	Pandas: Data structures: Series and DataFrame, Loading datasets, Data inspection (head(), info(), describe()), Handling missing values, Data filtering, selection, and basic transformations	
	4	Scikit-learn: Introduction to Scikit-learn for feature engineering, covering preprocessing (imputation, scaling, encoding), feature creation, feature selection, dimensionality reduction (PCA), and basic pipeline construction. (Overview only)	
	5	NLTK, OpenCV for data preprocessing (Basic introduction)	
	6	Matplotlib: Line plot, Bar chart, Histogram	
	7	Seaborn: Statistical plots: Boxplot, Scatter plot, Heatmap	
3	MODULE TITLE: Exploratory Data Analysis (EDA)		7
	1	Histograms: Understanding data distribution, Identifying skewness	
	2	Boxplots: Detecting outliers, Visualizing spread and median	
	3	Correlation Analysis : Correlation matrix computation, Measuring relationships between variables, Identifying highly correlated features, Visualization using heatmaps	
4	MODULE TITLE : Feature Engineering Techniques		7
	1	Techniques for Numerical Data: Binning and Discretization, Polynomial and Interaction Features.	
	2	Categorical Data Techniques: One Hot Encoding, Label Encoding.	
	3	Feature extraction vs. feature selection, Steps in feature selection. Feature Selection Methods: Filter, Wrapper, and Hybrid.	
	4	Feature Reduction: Introduction and application of Principal Components Analysis. (Concept only)	
5	Teacher Specific Module :		30
	Use datasets in CSV format sourced from platforms such as Kaggle or equivalent. Perform all experiments using Python libraries: NumPy, Pandas, Matplotlib, Seaborn, and Scikit-learn.		
	Develop the ability to analyze datasets and apply appropriate preprocessing and feature engineering techniques, with proper interpretation of results.		
	Lab 1: Data Loading and Inspection Load dataset using Pandas		

	● View structure (head(), info(), describe()) ● Identify data types (numerical, categorical) Lab 2: Identifying Feature Types ● Classify features as: ○ Numerical / categorical ○ Discrete / continuous ○ Ordinal features ● Convert data types where required Lab 3: Handling Missing Data ● Detect missing values ● Apply imputation (mean, median, mode) ● Compare before and after results Lab 4: Data Cleaning ● Remove duplicates ● Handle inconsistent data (e.g., string formatting, categories) ● Basic outlier identification Lab 5: Text Preprocessing Preprocess text data and convert it into numerical features using TF-IDF Lab 6 : Image Preprocessing ● Load and Display Image ● Read an image file ● Display it ● Resize to 224×224 ● Normalize Image : Convert pixel values from 0–255 to 0–1 ● Apply Augmentation Techniques Lab 7: Histograms ● Plot histograms for numerical features ● Observe data distribution (normal, skewed) ● Identify skewness Lab 8: Boxplots ● Create boxplots for selected features ● Identify outliers ● Interpret median and spread Lab 9: Correlation Analysis ● Compute correlation matrix	
--	--	--

	- Analyze relationships between variables - Identify highly correlated features Lab10: Correlation Visualization - Plot heatmap for correlation matrix - Interpret strength and direction of relationships Lab 11: Binning and Discretization - Apply equal-width and equal-frequency binning - Convert continuous data into categorical bins Lab 12: Polynomial and Interaction Features - Generate polynomial features (e.g., square, cube) - Create interaction features between variables - Compare dataset before and after feature creation Lab 13: Encoding Categorical Variables - Apply label encoding - Apply one-hot encoding Lab 14: Feature Selection - Apply filter methods (e.g., correlation-based selection) - Select important features using statistical techniques Lab 15: Dimensionality Reduction - Apply Principal Component Analysis (PCA) - Interpret transformed features CASE STUDY TOPICS (any one of the following, or can also select any other suitable application, download suitable dataset for the application and do the preprocessing.) - Student Academic Performance Prediction - Dataset: Student Performance Dataset – UCI Repository - Case Study Objective: Predict student performance based on study habits, attendance, family background, and lifestyle factors. - Feature Engineering Concepts - Handling missing values - Label encoding - Feature scaling - Correlation-based feature selection	
--	---	--

	2. Titanic Passenger Survival Prediction a. Dataset: Titanic Dataset – Kaggle b. Case Study Objective: Predict whether a passenger survived using engineered features from passenger information. Create new feature: Family Size, Convert gender to numeric values, Normalize fare values c. Feature Engineering Concepts i. Handling categorical data ii. Creating family-size features iii. Age-group binning iv. Missing value imputation 3. House Price Prediction Using Housing Features a. Dataset: House Prices Advanced Regression Techniques – Kaggle b. Case Study Objective: Improve house price prediction through feature transformation and selection. Create “house age” feature, Combine room-related features, Handle null values in garage and basement data c. Feature Engineering Concepts: i. Polynomial features ii. Outlier removal iii. Log transformation iv. Feature importance analysis 4. Customer Segmentation Using Mall Customer Data a. Dataset: Mall Customers Dataset – Kaggle b. Case Study Objective: Group customers based on income and spending behavior. Normalize annual income, Create spending categories, Apply PCA before clustering c. Feature Engineering Concepts i. Feature scaling ii. Clustering preparation iii. Dimensionality reduction iv. Data visualization features	
--	---	--

Essential Readings:

1. M.C. Trivedi, Data Science and Data Analytics Using Python Programming, Khanna Publishing House, 2024.
2. Max Kuhn, Kjell Johnson, **Feature Engineering and Selection: A Practical Approach for Predictive Models**, CRC Press, Taylor & Francis Group

3. Feature Engineering for Machine Learning (Techniques, Tools and Applications), Khanna Publishing House
4. Guozhu Dong, Huan Liu, Feature Engineering for Machine Learning and Data Analytics, CRC Press
5. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). Fundamentals of Data Theory and Practice. Elsevier. ISBN-13: 9780323917780

Suggested Readings:

1. Ng, Andrew. (2018). Machine learning yearning (Draft, MIT Licensed). GitHub. ISBN10: 199957950X, ISBN-13: 978-1999579500.
2. Provost, Foster, & Fawcett, Tom. (2013). Data science for business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Case study	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

Employability for the Course:Data Analyst, Machine Learning Engineer

KU5SECCAP107 : PHP and MySQL

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	SEC	100-199	KU5SECCAP107	3(2T+1P)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2		25	50	75	1 Hr 30 Minutes

Course Description: This course introduces students to **server-side web development using PHP and MySQL**, focusing on building **dynamic, database-driven web applications**. Learners will understand client-server architecture, develop interactive websites, and integrate backend logic with relational databases. The course emphasises **hands-on learning**, enabling students to design, develop, and deploy simple web applications such as login systems and CRUD-based projects. PHP and MySQL are widely used technologies for web development and provide a strong foundation for backend programming careers

Course Prerequisite: **Programming & Basic understanding of DBMS**

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand basic concepts of web development and client-server architecture.	U
2	Apply PHP programming constructs to build dynamic web pages.	RA
3	Design and manage relational databases using MySQL.	RC
4	Develop database-driven web applications using PHP-MySQL integration.	C
5	Create simple web applications such as login systems and CRUD applications.	AC

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	3	1		1	1	1
CO 2	3	3	2	2		2	
CO 3	2	3	3	2	2		1
CO 4	3	3		3		3	3
CO 5	2	2	3	3	3	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Introduction to Web Technologies		8
	1	Basics of Web Development (Frontend vs Backend)	
	2	Client-Server Architecture	
	3	Introduction to PHP & MySQL	
	4	Installation of LAMPP/XAMPP/WAMP Server	
	5	Basic HTML for PHP	
2	PHP Programming Fundamentals		8
	1	PHP syntax, variables, constants, data types, input and output functions	
	2	Operators and expressions, PHP Echo And Quotes, Formatting Using sprintf, Outputting Tabular Data, Outputting Data And Time	
	3	Control structures (if, switch, loops)	
	4	Functions and scope, Arrays and strings	
	5	Form handling (GET, POST)	

3	Advanced PHP Concepts		7
	1	Sessions and cookies	
	2	File handling, Error and exception handling	
	3	Introduction to OOP in PHP	
	4	PHP security basics (validation, sanitization)	
4	MySQL & PHP Integration		7
	1	Introduction to databases and DBMS , Connecting PHP with MySQL (mysqli/PDO)	
	2	SQL basics (DDL, DML, DCL) , Creating databases and tables, CRUD operations	
	3	MySQL: Connect, Create DB, Create Table, Insert Data, Get Last ID, Insert Multiple, Prepared, Select Data, Where, joins, Order By, Delete Data, Update Data. Creating Authentication system (login & registration)	
5	Teacher Specific Module: For practical courses, add lab experiments		30
	1. Installation and configuration of local server (LAMPP/XAMPP/WAMP) 2. Implement PHP programs using variables, loops, and functions 3. Implement Form handling using GET and POST 4. Create and manage MySQL databases and tables 5. Perform CRUD operations using PHP and MySQL 6. Session-based login/logout system 7. File handling, upload and download 8. Mini Project (Anyone): a. Students' Event Registration System i. Features 1. Event registration form 2. Store participant details 3. Display participant list while selecting an event/all b. Online Quiz Application i. Features 1. Display multiple-choice questions 2. Calculate score 3. Store result in database c. Student Feedback System i. Features 1. Student login (simple) 2. Submit feedback form		

	3. Store feedback in MySQL 4. Admin can view feedback d. Simple Login and Registration System i. Features 1. User registration 2. Login authentication 3. Welcome dashboard 4. Logout	
--	--	--

Essential Readings:

1. “PHP and MySQL Web Development” – Luke Welling & Laura Thomson
2. “Head First PHP & MySQL” – Lynn Beighley & Michael Morrison
3. “Learning PHP, MySQL & JavaScript” – Robin Nixon

Suggested Readings:

1. “PHP: The Complete Reference” – Steven Holzner
2. “PHP for Beginners” – Ivan Bayross (commonly used in Indian curricula)
3. “Programming the World Wide Web” – Robert W. Sebesta
4. Online Documentation: PHP Manual (php.net) , MySQL Documentation

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar/Casestudy	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

KU5SECCAP108: AGILE SOFTWARE DEVELOPMENT FOR SDE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	SEC	100-199	KU5SECCAP108	3(3+0)	45

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	0		25	50	75	1 Hr 30 Minutes

Course Description: This course introduces the principles, methodologies, and practices of Agile software development and product management. It covers Agile values, Scrum framework, user stories, Lean Canvas, Agile architecture, design practices, and Agile testing approaches. The course also provides exposure to modern software development practices such as continuous integration, refactoring, test-driven development, and technical debt management.

Course Prerequisite: Software Engineering fundamentals

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	To understand agile development processes and the principles behind the Agile Manifesto	U
2	Create user stories and apply User-Centric Design and Lean Canvas concepts for product development.	A, C
3	Apply Scrum practices including sprint planning, backlog management, and Agile team roles.	A
4	Understand Agile architecture, prototyping, refactoring, and technical debt management practices.	U
5	Apply Agile testing approaches and test automation concepts.	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	2	1	3	2	2	3
CO2	3	3	1	2	2	2	3
CO3	3	3	2	3	2	2	3
CO4	3	3	2	3	2	2	3
CO5	3	3	2	3	2	2	1212

COURSE CONTENTS

Contents for Classroom Transaction:

MO DU LE	UN IT	DESCRIPTION	HOURS
1	Introduction to Agile Methods and Product Development:		8
	1	Introduction to Agile – Agile values and principles	
	2	Agile vs. traditional development models	
	3	Overview of Agile Manifesto, user stories & User-Centric Design (Using Lean Canvas) – Writing effective user stories	
	4	Introduction to User-Centric Design, Using Lean Canvas for product strategy,	
	5	Product Visioning – Techniques to define product vision.	
2	Agile Frameworks and Software Development Practices		8
	1	Scrum Framework – Scrum processes: sprints, sprint planning, sprint review, and retrospectives	
	2	Scrum roles – Product Owner, Scrum Master, Development Team, Key Scrum artifacts – Product Backlog, Sprint Backlog, and Increment	
	3	Project velocity and Burn down chart	
3	Agile Architecture and Design		10
	1	Agile approach to software architecture,,,	

	2	Prototypes and wireframes in agile	
		Agile architecture practices – evolving design, emergent architecture	
	3	Pair Programming & Refactoring – Principles and benefits, Continuous code refactoring and technical debt management.	
4	Testing		
	1	Testing: The Agile lifecycle and its impact on testing	10
	2	Test driven development– Acceptance tests and verifying stories, writing a user acceptance test	
	3	Developing effective test suites, Continuous integration, Code refactoring	
	4	Risk based testing, Regression tests, Test automation.	
5	Teacher Specific Module :		9
	Guide and evaluate students in Agile project activities, Scrum practices, product design, software architecture, and testing tasks through continuous assessment		
	Case study : Create: ● 3 user stories, ● a simple Lean Canvas ● product vision statement ● a product backlog, ● sprint backlog, ● simple burn down chart ● basic wireframes/mockups, ● perform simple code refactoring ● user acceptance test cases, ● regression test cases		

Essential Readings:

1. Ken Schwaber & Mike Beedle, Agile Development with Scrum, K Prentice Hall, 2001
2. Peter Schuh, Charles River, Integrating Agile Development in the Real World, Media, 2005
3. Robert C. Martin, “Agile Software Development, Principles, Patterns and Practices”, First International Edition, Prentice Hall.
4. Pedro M. Santos, Marco Consolaro, and Alessandro Di Gioia, “Agile Technical Practices Distilled: A learning journey in technical practices and principles of software design”, First edition, Packt Publisher.

5. David J. Anderson and Eli Schragenheim, —Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Prentice Hall, 2003.

Online Courses and Video lectures

1. Agile Software Development,
<https://www.edx.org/course/agile-software-development>
2. Agile Software Development,
<https://www.coursera.org/learn/agile-software-development>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
CCA		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Total		75

KU6SECCAP109: FUNDAMENTALS OF CLOUD COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	SEC	100-199	KU6SECCAP109	3(3+0)	45

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	-	-	25	50	75	1 Hr 30 Minutes

Course Description: This course introduces the fundamental concepts and techniques of cloud computing, including cloud architecture, virtualization, service and deployment models, and cloud platforms. It focuses on understanding how computing resources are delivered over the internet and how cloud technologies are used to store, manage, and process data efficiently. The course also covers cloud security, real-world applications, and emerging trends, enabling students to understand and apply cloud-based solutions to modern computing problems.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of cloud computing, evolution of computing, and basic cloud architecture.	U
2	Apply the concepts of virtualization and cloud technologies	A
3	Understand and use basic cloud platform services, pricing models, and management features on platforms like Amazon Web Services, Microsoft Azure, and Google Cloud Platform.	A
4	Understand the concepts of cloud security and applications, including risks and challenges, data security and privacy, backup and disaster recovery, real-world applications, big data integration, and emerging trends in cloud computing.	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	1	1	1			2
CO 2	3	2	3	1			2
CO 3	3	2	3	2	1		3
CO 4	2	1	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE- INTRODUCTION TO CLOUD COMPUTING		8
	1	Introduction , Features of cloud computing, Benefits and limitations of cloud computing,	
	2	Evolution of computing platforms Mainframe → Client-Server → Distributed Systems → Cloud Computing	
	3	Cloud Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud	
	4	Cloud Service Models: PaaS, SaaS, IaaS, XaaS	
	5	Basic Cloud Architecture: Front end, Back end	
2	MODULE TITLE- VIRTUALIZATION AND CLOUD TECHNOLOGIES		8
	1	Virtualization ,Types of virtualization Application virtualization-Network virtualization - OS virtualization- Storage virtualization- Server virtualization- Data virtualization	
	2	Tools and Products available for Virtualization	

	3	Data Centers in Cloud computing	
3	MODULE TITLE- CLOUD PLATFORMS AND SERVICE MANAGEMENT		10
	1	Cloud Service Providers: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform	
	2	Cloud Pricing Models: Pay as you go, Subscription model, Reserved Instances / Savings Plans, Spot Instances (AWS concept)	
	3	Cloud Management Cloud Management Task: Auditing System Backups, Monitoring & Logging, Solution Testing and Validation Benefits of Effective Cloud Management	
4	MODULE TITLE- CLOUD SECURITY & APPLICATIONS		10
	1	Introduction: Cloud security, Risks & Challenges in Cloud	
	2	Data Security & Privacy Backup & Disaster Recovery	
	3	Real-world Applications of cloud	
	4	Introduction to Big Data and Cloud Integration	
	5	Emerging Trends in cloud: Edge computing, AI & ML, IoT and Cloud Integration, Green cloud computing	
5	Teacher Specific Module		9
	<i>Directions</i>		
	1.	Identify everyday apps and classify them under cloud services using platforms like Amazon Web Services, Microsoft Azure, and Google Cloud Platform.	

Essential Readings:

1. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGraw Hill Edition, Fourth Reprint, 2010
2. Mastering Cloud Computing – Rajkumar Buyya
3. Ronald L.Krutz, Russell vines, Cloud Security: A Comprehensive Guide to Secure Cloud
4. Kumar Saurabh, Cloud Computing, Wiley India
5. Gautam, Enterprise Cloud Computing Technology Architecture Applications, Shroff

Suggested Readings:

1. K. Chandrasekaran – Essentials of Cloud Computing
2. <https://www.slideshare.net/slideshow/fundamentals-of-cloud-computing-191620361/191620361>
3. https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem_2/DECAP470_CLOUD_COMPUTING.pdf

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Total		75

KU6SECCAP110: ANDROID APPLICATION DEVELOPMENT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	SEC	100-199	KU6SECCAP110	3(2+1)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	1	25	50	75	1 Hr 30 Minutes

Course Description: It covers the aspects of Android application development, designing user interfaces , how to persistently store data by using content providers, creating & managing pictures and menus , sending messages and e-mails, working with location services and maps, etc.

Course Prerequisite: Object Oriented Programming through Java

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the Android platform architecture and develop basic applications using Activities, Fragments, and Intents to manage application flow and user interaction.	U
2	Design interactive and responsive user interfaces using Views and View Groups, and implement features such as image handling, menus, and event-driven UI components	U A C
3	Implement persistent data management in Android applications using SQLite, SharedPreferences, and file storage, and integrate communication features such as Email and SMS services.	A C
4	Develop advanced Android applications by integrating Location Services and Maps, and implementing multimedia features including graphics, animation, audio, video, and camera functionalities.	A C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	3	1	2	1	1	2
CO2	3	3	1	2	1	2	2
CO3	3	3	3	2	1	2	2
CO4	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Overview of Android & Using Activities, Fragments and Intents		8
	1	Overview: Introduction, Android APIs, Android Architecture, Application Framework, Features of Android, Application Components.	
	2	The Manifest File. Developing and Executing the First Android Application, Using IDE to Create, Run & Explore an Application, Command-Line Tools	
	3	Working with Activities: Creating an Activity. Starting an Activity. Managing the Lifecycle of an Activity. Applying Themes and Styles to an Activity. Displaying a Dialog in the Activity. Hiding the Title of the Activity.	
	4	Using Intents: Exploring Intent Objects, Exploring Intent Resolution. Exploring Intent Filters, Resolving Intent Filter Collision, Linking the Activities Using Intent. Obtaining Results from Intent, Passing Data Using an Intent Object	
	5	Fragments: Fragment Implementation. Finding Fragments. Adding, Removing, and Replacing Fragments. Finding Activity Using Fragment	

2	Using Views and View Groups		8
	1	Working with View Groups : The LinearLayout Layout, The RelativeLayout Layout, The ScrollView Layout, The TableLayout Layout, The FrameLayout Layout, The TabLayout Using the Action Bar	
	2	Working with Views: Using the TextView, Using the EditText View, Using the Button View, Using the RadioButton View,	
	3	Working with Views: Using the CheckBox View, Using the ImageButton View, Using the ToggleButton View, Using the RatingBar View	
3	User Interface, Handling Pictures and Menus with Views		7
	1	Handling UI Events : Handling User Interaction with Activities, Handling User Interaction with the Views	
	2	Working with Image Views: Displaying Images in the Gallery View, Displaying Images in the Grid View, Using the ImageSwitcher View	
	3	Notifying the User: Creating the Toast Notification, Creating the Status Bar Notification, Creating the Dialog Notification	
	4	Creating Menus: The Options Menu, The Context Menu, The SubMenus	
4	Storing the Data Persistently		7
	1	Introducing the Data Storage Options: Using Preferences Using the Internal Storage: Exploring the Methods Used for Internal Storage, Developing an Application to Save User Data Persistently in File	
	2	Using the External Storage: Exploring the Methods Used for External Storage, Developing Application to Save File in SD Card..	
	3	Using the SQLite Database: Creating the Database Helper Class, Creating the Layout and Main Activity Class, Creating the Layout and Activity for the Insert Operation, Creating the Layout and Activity to Search a Record, Creating the Activity Class to Fetch All Records	
	4	Building an Application to Send Email. Handling SMS: Sending SMS Using SmsManager. Sending SMS Using Intent: Receiving SMS Using the BroadcastReceiver Object	
5	Teacher Specific Module : Lab experiments		30
	<i>Directions</i>		
	Experiment 1: Student Profile Navigation App (Activities & Intents)		

	Aim: To develop an app to navigate between activities using explicit intents. Procedure: 1. Create a new Android project 2. Add two Activities 3. Design Activity 1 with EditText (Name, Roll No) and Button 4. On button click, pass data using Intent 5. Retrieve data in Activity 2 using <code>getIntent()</code> 6. Display data using TextView Sample Code: <pre>Intent i = new Intent(MainActivity.this, SecondActivity.class); i.putExtra("name", name); startActivity(i);</pre> Output: Displays entered student details in second screen Experiment 2: News App using Fragments Aim: To create a dynamic UI using Fragments. Procedure: 1. Create two fragments (List & Detail) 2. Display list of items in ListFragment 3. Replace DetailFragment on item click 4. Use FragmentManager Sample Code: <pre>FragmentTransaction ft = getSupportFragmentManager().beginTransaction(); ft.replace(R.id.container, new DetailFragment()); ft.commit();</pre> Output: Content updates without opening new Activity Experiment 3: Registration Form (Views & ViewGroups) Aim: To design UI using layouts. Procedure: 1. Use LinearLayout as root 2. Add EditText, RadioButton, Spinner 3. Add Submit Button	
--	--	--

	4. Display input using Toast Output: User enters data and submits form Experiment 4: Calculator using GridLayout Aim: To design a calculator UI. Procedure: 1. Add GridLayout 2. Add buttons (0–9, +, -) 3. Add TextView for display 4. Handle button clicks Output: Performs basic arithmetic Experiment 5: Image Gallery App Aim: To display images dynamically. Procedure: 1. Add images to drawable 2. Use ImageView 3. Add Next/Previous buttons 4. Change images on click Output: Images change on button click Experiment 6: Menu Handling App Aim: To implement options menu. Procedure: 1. Create menu XML 2. Override onCreateOptionsMenu() 3. Handle item clicks Output: Menu displayed and options selectable Experiment 7: Profile App with Image Selection	
--	---	--

	Aim: To select an image from the gallery. Procedure: 1. Add button to open gallery 2. Use Intent ACTION_PICK 3. Handle result 4. Display image Output: Selected image displayed Experiment 8: Login using SharedPreferences Aim: To store login data. Procedure: 1. Use SharedPreferences 2. Save data on login 3. Retrieve data on app start Sample Code: <pre>SharedPreferences sp = getSharedPreferences("MyData", MODE_PRIVATE); SharedPreferences.Editor ed = sp.edit(); ed.putString("username", name); ed.apply();</pre> Output: User stays logged in Experiment 9: Notes App using SQLite Aim: To store structured data. Procedure: 1. Create SQLite database 2. Create table 3. Insert and retrieve data 4. Display in ListView Output: User can manage notes Experiment 10: File Storage App	
--	--	--

	Aim: To store data in files. Procedure: 1. Save text using FileOutputStream 2. Read using FileInputStream 3. Display data Output: Saved data retrieved successfully	
--	---	--

Essential Readings:

1. Android Application Development (with kitkat support) - Black Book by Pradeep Kothari, Dreamtech Press
2. Android Application Development with Kotlin by Hardik Trivedi:BPB Publications
3. Fundamentals of Android App Development by Sujit Kumar Mishra, BPB Publications
4. Beginning Android 4 Application Development by Wei Meng Lee, Wrox

Suggested Readings:

1. <https://developer.android.com/get-started/codelabs>
2. <https://www.geeksforgeeks.org/android/introduction-to-android-development/>
3. <https://developer.android.com/topic/architecture/intro>

Assessment Rubrics:

Evaluation Type	Marks
End Semester Evaluation	50
Theory	35
Practical	15
CCA	25
Continuous Evaluation (Theory)	15

a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

KU6SECCAP111: ENTREPRENEURSHIP IN SOFTWARE ENGINEERING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	SEC	100-199	KU6SECCAP111	3(3+0)	45

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	0		25	50	75	1 Hr 30 Minutes

Course Description: This course is to introduce students to entrepreneurship in the software domain, develop skills in idea generation and validation. This will help in understanding software product development and business models. The course will provide knowledge on startup creation, funding, and growth

Course Prerequisite: Knowledge in Software Engineering

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of entrepreneurship and the software industry	U
2	Identify and validate software business ideas	A
3	Apply knowledge of software development and business models	A
4	Analyze startup processes, funding, and growth strategies	An

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	2	1	2	2	2	3

CO2	3	2	1	2	3	2	3
CO3	3	3	2	2	2	2	2
CO4	3	2	1	3	3	3	2

COURSE CONTENTS

Contents for Classroom Transaction:

MO DU LE	U NI T	DESCRIPTION	HOUR S
1	Introduction to Entrepreneurship & Software Industry		8
	1	Definition, Meaning and types of entrepreneurship	
	2	Role of entrepreneurs in the digital economy	
	3	Characteristics of successful software entrepreneurs	
	4	Overview of software industry (product vs service companies)	
2	Idea Generation & Validation		8
	1	Identifying problems and opportunities	
	2	Idea generation techniques (brainstorming, design thinking basics)	
	3	Market research and competitor analysis	
	4	Minimum Viable Product (MVP) concept	
	5	User validation techniques (feedback, surveys, prototyping)	
3	Software Product Development & Business Models		10
	1	Team building and project management basics	
	2	UI/UX fundamentals	
	3	Business models (SaaS, freemium, subscription, licensing)	
	4	Costing, pricing, and revenue strategies	
4	Startup Launch & Growth		10
	1	Steps to start a software venture	

	2	Funding sources (bootstrapping, angel investors, venture capital)	
	3	Legal basics (intellectual property, data privacy)	
	4	Digital marketing (SEO, social media marketing)	
	5	Scaling and sustainability challenges	
5	Teacher Specific Module : Activities Guidelines		9
	<i>Directions</i>		
	• Case studies of successful software startups(any 2) • You are planning to create a software-based solution (app or website) to solve a real life problem faced by students or society, Identify one Problem or Opportunity, List 3 possible ideas/solutions, Do market research by Checking if similar solutions already exist and List 3 basic features your product must have • You want to create a mobile app for students that helps them manage notes and reminders, Identify Basic Costs, Fix a Price and Estimate Simple Revenue. Suggest a pricing plan (e.g., ₹199/month per subscriber) and Justify why it suits the product and target users • You have started a small startup called “StudyMate”, a mobile app that helps students manage notes, reminders, and exam schedules. Identify Target Audience, Choose Digital Platforms, Create a Simple Promotion Idea, Create a short ad message (2–3 lines) for your app, Measure your marketing		5

Essential Readings:

1. Entrepreneurship Development And Small Business Enterprises
Poomima M Charantimath, Pearson
2. Zero to One, Peter Thiel, Penguin Random House Company, New York.
3. Entrepreneurial Development, by C B Gupta
4. The StartupOwner’s Manual, Step by Step Guide for building a Great Company, Steve Blank & Bob Dorf
5. Indian Software Industry: Business Strategy and Dynamic Co-Ordination, by Parthasarathi Banerjee

Suggested Readings:

1. Engineering Software Products, An introduction to Modern Software Engineering, Ian Sommerville, Pearson Edition
2. <https://theleanstartup.com/>
3. <https://www.simplilearn.com/entrepreneurship-course-free-skillup>
4. India's Software Industry: Politics, Institutions and Policy Shifts by Siddhartha Mukerji
5. <https://cursa.app/en/free-course/technology-entrepreneurship-eha>
6. <https://www.hubspot.com/startups/resources>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
CCA		25
a)	Test Papers	12
b)	Assignment	4
c)	Viva/Seminar	5
d)	Activities	4
Total		75

- o **Employability for the Course:** This course enhances entrepreneurial, technical, managerial, and innovation skills required in the modern software industry and startup ecosystem. It provides employability opportunities like Software Entrepreneur, Startup Founder / Co-Founder, Product Developer, Software Project Coordinator and Business Analyst etc.,

KU6SECCAP112: PROFESSIONAL SKILLS AND PLACEMENT READINESS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	SEC	100-199	KU6SECCAP112	3(3+0)	45

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	0	0	25	50	75	1 Hr 30 Minutes

Course Description: This course is designed to enhance students' aptitude, analytical thinking, programming logic, technical knowledge, and placement readiness skills required for campus recruitment processes. The course covers quantitative aptitude, logical and analytical reasoning, programming aptitude, technical interview preparation, and professional development activities. Students will develop problem-solving abilities, coding logic, communication skills, and interview confidence through practice-oriented learning, aptitude exercises, debugging tasks, group discussions, mock interviews.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Develop quantitative aptitude and logical reasoning skills required for competitive examinations and campus placements.	A
2	Strengthen programming logic, analytical thinking, and problem-solving abilities using coding-based aptitude exercises.	A
3	Improve understanding of technical concepts in Operating Systems, DBMS, Computer Networks, Data Structures, and Algorithms.	U An
4	Build professional readiness through CV preparation, LinkedIn profile development, group discussions, and interview activities.	A C
5	Prepare students for placement processes by improving employability skills, technical competency, and self-confidence.	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	1	2	1	2	3
CO2	3	3	2	2	1	1	3
CO3	3	3	3	2	2	1	2
CO4	2	1	1	3	2	3	3
CO5	3	2	2	3	2	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Quantitative Aptitude		8
	1	Number systems	
	2	Percentages	
	3	Ratio and proportion	
	4	Profit and loss	
	5	Time and work	
	6	Time, speed and distance	
	7	Simple and compound interest	
	8	Data interpretation	
2	MODULE TITLE : Logical & Analytical Aptitude		8
	1	Series and patterns	
	2	Coding-decoding	

	3	Blood relations	
	4	Direction Sense Test	
	5	Syllogisms, Logical puzzles	
3	MODULE TITLE: Programming Aptitude		10
	1	Algorithms and flowcharts, Pseudocode	
	2	Pattern-based problem solving Basic programming logic, Logical Thinking for Coding Problems Pattern Recognition and Analytical Reasoning Number and Pattern Problems • Number pattern problems • Symbol and character pattern problems ○ Star pyramid ○ Alphabet patterns ○ Number pyramids ○ Inverted and hollow patterns Series and Sequence Problems • Fibonacci series • Arithmetic and geometric sequences • Square and cube number series • Pattern-based summation problems ○ Example: $1 + 1/2 + 1/4 + 1/8 + \dots$ • Next term prediction Analytical Reasoning Programming Problems • Check prime number • Reverse a number • Palindrome number • Armstrong numbers between limits • Factorial and factorial series • Greatest common divisor (GCD) and LCM • Sum of digits and number manipulation problems String and Array Logic Problems • Find frequency of characters • Substring search • Reverse a string • Check palindrome string • Second largest element in array • Remove duplicate elements from an array	

		● Rotate an array by K positions ● Find maximum sum subarray ● Search and sorting basics ● 2D array logic problems	
	3	Output prediction (Arrays, Control Structures, Pointers, Functions, Recursion in C Programming)	
	4	Debugging (Arrays, Control Structures, Pointers, Functions, Recursion in C Programming) (Familiarising, Detecting and Correcting Errors)	
	MODULE TITLE: Technical & Placement Readiness		10
4	1	LinkedIn profile creation and optimization	
	2	CV preparation, covering letter	
	3	Group discussion: Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate	
	4	HR Interview Preparation: HR round: Self Introduction, Strength and Weakness Analysis	
	5	Technical interview preparation Discuss MCQ and concept-based questions Operating System concepts ● Process and thread ● CPU scheduling ● Memory management ● Deadlock basics Database Management System (DBMS) ● ER model and normalization ● SQL queries ● Primary and foreign keys ● ACID Property, Computer Networking ● OSI and TCP/IP models ● IP addressing ● Routing and switching basics ● HTTP, DNS, and network security basics Data Structures and Algorithms ● Arrays, stacks, queues, linked lists	

		• Trees and graphs (basic idea) • Searching and sorting algorithms • Time and space complexity basics Software Engineering • SDLC, Agile Model • Blackbox and Whitebox Testing	
5	Teacher Specific Module :		9
	<i>Directions</i>		
	Faculty may conduct practical sessions, lab exercises, coding practice, mock aptitude tests, technical MCQs, group discussions, resume workshops, interview simulations.		

Essential Readings:

1. Aggarwal, R. S. (2017). Quantitative Aptitude for Competitive Examinations. S. Chand Publishing.
2. Verma, R. (2018). Fast Track Objective Arithmetic. Arihant Publications.
3. Rizvi, M. A. (2005). Effective Technical Communication. Tata McGraw-Hill Publishing.
4. Sonmez, J. (2015). Soft Skills: The Software Developer's Life Manual. Manning Publications.
5. Mitra, B. K. (2011). Personality Development and Soft Skills. Oxford University Press.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Continuous Evaluation (Theory)		25
a)	Aptitude Test	5

b)	Programming Test	5
c)	Group Discussion	5
d)	CV Preparation	5
e)	Technical Interview	5
Total		75

- **Employability for the Course:** Upon successful completion of the course, students will be able to demonstrate improved aptitude, programming logic, analytical reasoning, technical interview skills, communication abilities, and professional preparedness, thereby enhancing their employability, placement readiness, and confidence for campus recruitment processes and IT career opportunities.

KU6SECCAP113: UI/UX DESIGN

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	SEC	100-199	KU6SECCAP113	3(2+1)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	-	25	50	75	1 Hr 30 Minutes

Course Description: This course introduces students to the fundamentals of User Interface (UI) and User Experience (UX) Design, focusing on designing intuitive, user-friendly digital products such as websites and mobile applications. Students will learn design thinking, user research, wireframing, prototyping, and usability principles, along with hands-on experience using modern design tools. The course emphasises practical skills and creativity, enabling learners to design visually appealing and user-centred interfaces.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand fundamentals of UI and UX design principles.	U
2	Apply design thinking to solve user problems.	RA
3	Create wireframes and prototypes for applications.	RC
4	Use UI/UX design tools for interface development.	UA
5	Design and evaluate user-friendly digital interfaces	CE

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	2	1	1	1	1	1
CO 2	3	3	2	1	2	1	1
CO 3	2	3	3	2	2	3	3
CO 4	2	2	3	3	2	3	3
CO 5	3	3	3	3	3	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Introduction to UI/UX Design		7
	1	Basics of UI vs UX.	
	2	Importance of user-centred design.	
	3	Design thinking process (Empathise, Define, Ideate, Prototype, Test).	
	4	Principles of good design (consistency, simplicity, feedback).	
	5	Overview of UI elements (buttons, icons, forms).	
2	UX Design Process		8
	1	User research methods (surveys, interviews).	
	2	Creating user personas.	
	3	User journey mapping.	
	4	Information architecture.	
	5	Wireframing concepts (low-fidelity design).	

3	UI Design Principles		8
	1	Visual design fundamentals (color, typography, layout)	
	2	Grid systems and alignment	
	3	Design systems and style guides	
	4	Accessibility basics (contrast, readability)	
	5	Responsive design basics	
4	Prototyping & Tools		7
	1	Introduction to design tools (e.g., Figma, Adobe XD)	
	2	Creating high-fidelity designs	
	3	Interactive prototyping	
	4	Usability testing basics	
	5	Design handoff concepts	
5	Teacher Specific Module : For practical courses add lab experiments		30
	1. Create user personas for a given problem 2. Develop user journey maps 3. Design low-fidelity wireframes (paper/digital) 4. Create high-fidelity UI screens using Figma 5. Build clickable prototypes 6. Conduct basic usability testing and feedback analysis 7. Mini Project (Any one): • Mobile app design (e.g., food delivery, education app) • Website redesign (college website, portfolio) • E-commerce UI prototype		

Essential Readings:

1. **“Don’t Make Me Think” – Steve Krug**
2. **“The Design of Everyday Things” – Don Norman**
3. **“UI/UX Design Basics for Beginners” – Joel Marsh**

Suggested Readings:

1. **“About Face: The Essentials of Interaction Design” – Alan Cooper**
2. **“100 Things Every Designer Needs to Know About People” – Susan Weinschenk**
3. **“Lean UX” – Jeff Gothelf**

4. Online Resources:
 - i. Figma tutorials
 - ii. UX blogs and design communities

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		50
Theory		35
Practical		15
CCA		25
Continuous Evaluation (Theory)		15
a)	Test Papers	8
b)	Assignment	2
c)	Viva/Seminar	5
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		75

DISCIPLINE SPECIFIC COURSES (DSC)

KU3DSCCAP201: DISCRETE MATHEMATICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200-299	KU3DSCCAP201	4 (4T+0P)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	Nil	-	30	70	100	2 Hrs.

Course Description: : This course introduces mathematical techniques that are foundations for analysing and understanding problems in computer science.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Provide a basic understanding of fundamental mathematical concepts such as sets, functions.	U, A
2	Acquire knowledge in Mathematical Logic	U
3	Acquire knowledge in Predicate Calculus	U, A
4	Awareness about the importance of Graph Theory in Computer Application	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1		2			2		3
CO 2		2			2		3
CO 3		2			2		3

CO 4		2			2		3
------	--	---	--	--	---	--	---

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE 1:		12
	1	Set Theory: Basic concepts- Set Operations, Properties of Set operations, Subset Venn diagram- Cartesian product.	
	2	Mathematical Logic - Propositional Calculus -Statement, Connectives, negation, conjunction, disjunction, conditional, biconditional, statement-- Conjunctive Normal Forms (CNF) and Disjunctive Normal Forms (DNF).	
	3	Propositions, logical operations (basic connectives), compound statements, construction of truth table, conditional statements, tautology, contradiction, contingency. Equivalence of formula- Well Formed Formula (WFF)- Tautologies, Normal Forms, Rules of inference.	
	4	Methods of proofs: Rules of inference for propositional logic, modus ponens, modus tollens, syllogism, proof by contradiction, Predicate logic, quantifiers (basic introduction)	
	5	Mathematical Induction	

2	MODULE 2: Functions and Relations		12
	1	Functions, properties of functions (domain, range), composition of functions, surjective (onto), injective (one-to-one) and bijective functions, inverse of functions. Some useful functions for Computer Science: Exponential and Logarithmic functions, Polynomial functions, Ceiling and Floor functions.	
	2	Relations - Relations and Their Properties, Functions as relations, Closure of Relations, Composition of relations, Equivalence Relations and Partitions.	
	3	Partial Ordering, Hasse Diagram. Basics of counting, Pigeonhole principle, permutation, combination, Binomial coefficients, Binomial theorem	

3	MODULE 3: Elementary Graph Theory		
	1	Basic terminologies of graphs -- connected and disconnected graphs, subgraph, paths and cycles, complete graphs, digraphs, weighted graphs, bipartite graph- complete bipartite graph- Isomorphic graph	12
	2	Trees: Definition- spanning tree- minimal spanning tree (MST)- DFS- BFS- incidence matrix - Traveling salesman's problem.	

4	MODULE 4:		
	1	Planar graph- Shortest Paths in Weighted Graphs- Euler's Paths and Circuits, Hamiltonian Paths and Circuits.	12
	2	Storage representation and manipulation of graphs. Coloring, chromatic number.	

5	MODULE 5: Teacher Specific Module		
	Teacher can implement proper methodologies and evaluation metrics related with the topics		12

Essential Readings:

1. Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Kamala Krithivasan, McGraw Hill Education
2. Kolman B., Busby R. and Ross S., Discrete Mathematical Structures, 6th Edition, Pearson Education, 2015.
3. Deo Narsingh, Graph Theory with Application to Engineering and Computer Science, Prentice Hall, India, 1979

Suggested Readings:

1. J. K. Sharma Discrete Mathematics, Macmillan Publishers India Limited
2. Grimaldi Ralph P. and Ramana B. V., Discrete and Combinatorial Mathematics: An Applied Introduction, Fifth Edition, Pearson Education, 2007.
3. West Douglas B., Introduction to Graph Theory, Second Edition, Pearson Education, 2015

Assessment Rubrics:

End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation (Theory)		30
a)	Test Paper-1	15
b)	Assignment	10
c)	Viva/seminar/Case study	5
Total		100

KU3DSCCAP202: OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200-299	KU3DSCCAP202	4 (3T+1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1.5 Hrs.

Course Description: This course introduces the concepts of object-oriented programming like abstraction, encapsulation, inheritance, polymorphism and applies them in solving problems. The object-oriented concepts are introduced through Java language. The course equips students to setup JDK environment to create, debug and run Java programs

Course Prerequisite: KU1DSCCAP101: Foundations of Computers and Programming
KU1DSCCAP106 : Programming with C and C++

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic principles of the object-oriented programming	U, A
2	Develop small to medium sized applications using Java	A, C
3	Demonstrate an introductory understanding of graphical user interfaces, multithreaded programming,	U, A, C
4	Develop applications with database connectivity	U, A, C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	2					3
CO 2	3	3	2	2	2		
CO 3	2	2		2			3

CO 4	3	3	2		2		
------	---	---	---	--	---	--	--

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE 1:		
	1	Fundamentals of Object-Oriented Programming: Basic Concepts of Object-Oriented Programming (OOP)-- Objects and Classes. Encapsulation, Inheritance, Polymorphism Benefits and Applications of OOP.	11
	2	Java Evolution: Java Features, Difference between Java, C and C++, Java and Internet, Java Environment. Java Tokens and statements, Implementing Java program and JVM, Command Line Arguments	
	3	Overview of Java Language: Introduction to Simple Java Program, Use of Comments and Math function, Java Program Structure	
	4	Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, StringBuffer, File, this reference	
2	MODULE 2:		
	1	Inheritance – Inheritance types, super keyword, preventing inheritance: final classes and methods.	12
	2	Polymorphism – method overloading and method overriding, abstract classes and methods	
	3	Interfaces- Interfaces Vs Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface, inner class.	
	4	Packages- Defining, creating and accessing a package, importing packages.	

3	MODULE 3:		
	1	Exception handling-Benefits of exception handling, the classification of exceptions - exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, creating own exception subclasses.	12
	2	Multithreading – Differences between multiple processes and multiple threads, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication	
	3	GUI Programming with Swing - The AWT class hierarchy, Introduction to Swing, Swing Vs AWT, Hierarchy for Swing components, Overview of some Swing components – JButton, JLabel, JTextField, JTextArea, simple Swing applications	
	4	Layout management – Layout manager types – border, grid and flow Event Handling- Events, Event sources, Event classes, Event Listeners, Delegation event model, Examples: Handling Mouse and Key events, Adapter classes	

4	MODULE 4:		
	1	Collection Framework in Java – Introduction to java collections, Overview of java collection framework, Commonly used collection classes- Array List, Vector, Hash table, Stack,	10
	2	Connecting to Database – JDBC Type 1 to 4 drivers, connecting to a database, querying a database and processing the results, updating data with JDBC	

5	LAB EXPERIMENTS		
	- Write a program to read two numbers from user and print their product. - Write a program to print the square of a number passed through command line arguments. - Write a program to send the name and surname of a student through command line arguments and print a welcome message for the student. - Write a java program to find the largest number out of n natural numbers. - Write a java program to find the Fibonacci series & Factorial of a number using recursive and non-recursive functions. - Write a java program to multiply two given matrices.		30

	7. Write a Java program for sorting a given list of names in ascending order. 8. Write a Java program that checks whether a given string is a palindrome or not. Ex:MADAM is a palindrome. 9. Write a java program to read n number of values in an array and display it in reverse order. 10. Write a Java program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations. 11. Create a JAVA class called Student with the following details as variables within it. - USN, NAME, BRANCH, PHONE, PERCENTAGE - Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings. 12. Write a Java program that displays the number of characters, lines and words in a text. 13. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea() methods to calculate the area and perimeter of a circle. 14. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, override the calculateSalary() method to calculate and return the salary based on their specific roles. 15. Write a Java program using an interface called 'Bank' having function 'rate_of_interest()'. Implement this interface to create two separate bank classes 'SBI' and 'PNB' to print different rates of interest. Include additional member variables, constructors also in classes 'SBI' and 'PNB'. 16. Write a Java package program for the class book and then import the data from the package and display the result. 17. Write a Java program for finding the cube of a number using a package for various data types and then import it in another class and display the results. 18. Write a Java program for demonstrating the divide by zero exception handling. 19. Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.	
--	---	--

	20. Create an exception subclass UnderAge, which prints Under Age along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the Method test () throws UnderAge exception if the variable age passed to it 21. Write a Java program to create three threads and to display Good Morning for every one second, hello for every two seconds and welcome for every three seconds by using Thread class 22. Write a Java program that creates three threads. First thread displays OOP, the second thread displays Through and the third thread displays Java by using Runnable interface. 23. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area 24. Implement a Java program for handling key events when the keyboard is pressed, released, typed 25. Write a Java swing program that reads two numbers from two separate text fields and display the sum of two numbers in the third text field when button add is pressed. 26. Write a Java program to design student registration form using Swing controls. The form should have the following fields and a button Save • Name, Register No, Email Id, Gender, Branch, Address 27. Write a Java program to shuffle elements in arraylist 28. Write a Java program to iterate through all elements in a HashMap 29. JDBC program to insert, Delete and Update records into Employee table 30. JDBC program to display database metadata. 31. JDBC program to display Resultset metadata. 32. JDBC program to connect to Student table. Implement the record scrolling functions – first(), last(), next(), previous(), beforeFirst(), afterLast(), absolute() and relative().	
--	--	--

Essential Readings:

1. Understanding object-oriented programming with Java, T.Budd, Pearson Education
2. Java: How to Program , P.J.Deitel and H.M.Deitel, PHI
3. Object Oriented Programming through Java, P.RadhaKrishna, Universities Press.

Suggested Readings:

1. Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson
2. Introduction to Programming and Object–Oriented Design Using Java -- Jaime Niño, Frederick A. Hosch

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU3DSCCAP203: DIGITAL SYSTEMS AND INTRODUCTION TO MICROPROCESSORS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200-299	KU3DSCCAP203	4 (3T+ 1 P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1.5 Hrs.

Course Description: This course introduces the technical knowledge of digital circuits. Digital systems are at the heart of almost all modern mechatronics and electronics technologies. Describe the architecture & organization of 8086 Microprocessors. Understand and classify the instruction set of the 8086 microprocessor and distinguish the use of different instructions and apply it in assembly language programming. Relate the addressing modes used in the instructions. Realize the Interfacing of memory & various I/O devices with 8086 microprocessors. Interface various peripheral IC's with Intel 8086 microprocessor for its various applications.

Course Prerequisite: KU1DSCCAP101: Foundations of Computers and Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
1	Explain the fundamental principles of Combinational digital circuits	U
2	Compare and describe the fundamental concepts of microprocessor systems.	U, A
3	Describe the architecture & organization of 8086 Microprocessor. Understand and classify the instruction set of 8086 microprocessor and distinguish the use of different instructions and apply it in assembly language programming	A, C
4	Realize the Interfacing of microprocessors with peripheral devices.	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	3		3				3
CO2	3		3				2
CO3	3		3				2
CO4	3		3	3			2

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE: Digital Systems Basics		
	1	Number Systems and Codes: Binary, Octal and hexadecimal conversions	11
	2	BINARY ARITHMETIC: Addition, Subtraction. Excess -3 code, Gray code, BCD,	
	3	Data Representation: Data types - Complements (1's and 2's)–FixedPoint representation and Floating-Point representation.	
	4	Logic gates-AND, OR, NOT, NAND, NOR, EXOR, EXNOR.	
II	MODULE TITLE: Building Blocks of Digital Systems		
	1	Boolean Laws and theorems, Sum of Products method, Product of Sum method	12
	2	Boolean Algebra – DeMorgan's Theorem	
	3	K map representation and simplification (up to four variables)	
	4	Pairs, Quads, Octets, Don’t care conditions. Combinational circuits: Adders -Full adder	
III	MODULE TITLE: Introduction to 8086		
	1	Introduction to 8086 – Microprocessor architecture	12
	2	Functional Block Diagram	

	3	Register Organization of 8086	
	4	8086 PIN DIAGRAM. Addressing modes	
IV	MODULE TITLE: Assembly Language Programming		
	1	Introduction to Assembly language programming Instruction set - Data transfer instructions, Arithmetic and Logic instructions, Branch instructions, Loop instructions	10
	2	String Instructions, Assembler Directives and operators.	
	3	Flag Manipulation instructions, Shift and Rotate instructions, String instructions, Assembler Directives and operators.	
	4	STACK STRUCTURE OF 8086, Modes of data transfer- Programmed I/O, interrupt I/O, DMA.	
V	LIST OF LAB EXPERIMENTS		
		Implement using MASM. 1. Study of Assembler and Debugging commands. 2. A program to add a data byte located at offset 0500H in 2000H segment to another data byte available at 0600H in the same segment and store the result at 0700H in the same segment. 3. 8086 Assembly program that takes two numbers, adds them and displays the result on the screen (in decimal) 4. A program to swap two 8-bit numbers using a temporary variable. 5. A program for BCD addition of two 16 bit numbers. 6. An ALP to find the maximum number from the given array of N numbers. 7. A program to insert a byte into a string. 8. A program to check whether a string is palindrome or not. 9. A program to arrange a given array of N bytes in ascending order. 10. A program for Sum of Array Elements 11. A program to Count Even and Odd Numbers in an Array	30

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

1. K. M. Bhurchandi and A. K. Ray, Advanced Microprocessor and Peripherals, 3rd Ed, TMH
2. Ramesh Gaonkar, Microprocessor Architecture, Programming, and Applications with the 8085, 6th Ed, Penram International Publishing
3. Thomas L. Floyd, Digital Fundamentals, 11th Ed, Pearson
4. M. Morris Mano, Computer System Architecture, 3rd Ed, Pearson
5. Douglas V. Hall, Microprocessors and Interfacing: Programming and Hardware, 2nd Ed, McGraw Hill

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU3DSCCAP204: MATHEMATICAL FOUNDATIONS OF COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200 - 299	KU3DSCCAP204	4 (3T+ 1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1.5 Hrs.

Course Description: This course is intended to prepare students to use mathematical foundations in many areas of computer science like algorithms, computer networks, cryptography, data science, machine learning, artificial intelligence etc. To develop mathematical thinking in solving computer science related problems

Course Prerequisite: Basic Mathematics (Algebra, Arithmetic)

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Visualize vector operations and linear transformations	U, A
2	Apply matrices to real-world computer science problems	A
3	Ability to relate calculus concepts to practical CS applications	A
4	Builds foundational knowledge for cryptography & security.	U, A
5	Understand and apply core statistical and probability principles	U, A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	3			2		2
CO 2	2	3	2	2	2	2	2
CO 3	3	3	2	2	2	2	2
CO 4	3	3	2	2	2	2	2
CO 5	3	3	3	2	2	2	2

COURSE CONTENTS

Contents for Classroom Transaction:

NB: Concepts must be delivered by relating it to the scenarios in Computer Application

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Linear Algebra & Calculus		12
	1	Vector: Definition of vectors, Real coordinate spaces, Magnitude of a vector	
	2	Vector operations: addition, scalar multiplication, visualisation	
	3	Linear Independence, bases, dimension	
	4	Vector dot products, Interpretation of dot product	
	5	Orthogonal vectors	
	6	Matrix, Linear transformations, Eigen value, Eigen Vector (Geometric interpretation of concept)	
	7	Differentiation: Concept, Rate of change, slope of a function, Visualisation	
	8	Partial Derivatives: Concept, Derivatives w.r.t. one variable in multivariable functions, Gradient	
	9	Maxima and Minima (Optimization), Using first and second derivative	
	10	Integration: concept, Definite Integral, indefinite Integral, Area under curve	
	11	Examples to illustrate applications in Computer Science	
2	MODULE TITLE: Algebraic Structure & Number Theory		10
	1	Group: Closure, Associativity, Identity, Inverse Abelian Group, Cyclic Group	
	2	Modular Arithmetic, Addition and Multiplication modulo, Modular Inverse, Euclidean Algorithm, Extended Euclidean Algorithm Primality Testing	
3	3	Examples to illustrate applications in Cryptography	
	MODULE TITLE: Probability		12
	1	Experiment and sample space, events and operations with events (Union, Intersection, Complement of Events)	
3	2	Exclusive and exhaustive events, equally likely events with examples.	
	3	Probability of an event, basic probability rules, conditional probability.	

	4	Bayes' Theorem	
	5	Random Variables: Discrete and continuous random variable Probability distribution of a random variable, probability mass function, probability density function	
	6	Expectation and variance of a random variable	
	7	Standard Probability Distributions: Binomial probability distribution, Poisson probability distribution, Normal probability distribution, Measures of shapes	
(Focus on problem sets involving real-world applications of probability. Real-world scenario of Binomial, Poisson and Normal Distribution.)			
4	MODULE TITLE: Statistics		
	1	Basic concepts of Statistics, qualitative and quantitative data, classification of data: Nominal, ordinal, interval, ratio Construction of frequency distribution, diagrammatic representation of data : Histograms, Boxplots, Bar charts, pie chart Measures of Central Tendency: mean, median and mode—their properties Measures of Dispersion: Range, mean deviation, quartile deviation, variance and standard deviation. Highlight the use of Measures Mean, Median and Mode in Real-World Scenarios and the significance of measures of Dispersion in Data Analysis. Identify the inferences from these measures)	11
	2	STATISTICAL INFERENCE AND REGRESSION ANALYSIS Correlation: Definition, scatter diagram, Pearson's Coefficient of Correlation and Rank Correlation Regression: Linear regression-fitting by least square method and interpretation.	
5	Lab		
		Explore fundamental vector operations—addition, scalar multiplication, dot product, and transformation—using GeoGebra. $\vec{u}=(3,2)$ $\vec{v}=(1,4)$	30

	Using GeoGebra: 1. Visualize both vectors from the origin. 2. Perform and display vector addition 3. Apply scalar multiplication 4. Compute the dot product of $\vec{u}$ and $\vec{v}$ and interpret the result geometrically. 5. Apply a matrix transformation to rotate vector $\vec{u}$ by 90 degrees using matrix multiplication Interpret the visual and numerical results of each operation Visualize the derivative (slope of a tangent) of a function using GeoGebra. https://www.geogebra.org/classic Visualize and compute partial derivatives of a multivariable function using GeoGebra's 3D Graphing environment and understand the geometric meaning of partial derivatives. https://www.geogebra.org/3d Visualizing the Area Under a Curve in Geogebra https://www.geogebra.org/graphing • Plot a function ($f(x)$), e.g., ($f(x) = x^2$) over interval $[0, 5]$. • Shade the area under the curve. • Show how the definite integral corresponds to this shaded area. • Use tools like GeoGebra for visualization. Visualize probability distribution in Geogebra https://www.geogebra.org/classic	
--	---	--

Set up Lab for Statistics

Jamovi is a free, open-source, user-friendly statistical software

Download from the official website:
<https://www.jamovi.org>

Analyze the study habits of students by examining their daily study hours and corresponding exam scores. Understand the central tendency, variability, and distribution of these variables.

Also interpret Output (Average study hours and exam scores, Middle value when data is sorted, How spread out the data is, Whether data is symmetric or skewed, Visualization Insights: **Histogram:** Visualize distribution shape (normal, skewed).

Boxplot: Identify median, spread, and outliers.

Hint : Prepare Your Data with two columns Study_Hours, Exam_Score

Sample Data

Study_Hours	Exam_Score
2	55
4	70
3	65
.....	

By varying the values of study hours and corresponding exam scores for a group of students, visualize different types of data distributions (such as symmetric, right-skewed, and left-skewed) using histograms in Jamovi, and analyze how changes in data affect the shape, central tendency, and spread of the distribution.

		Analyze the number of daily emails received by employees in a software company to understand communication load. Use descriptive statistics and histograms in Jamovi to determine the central tendency, variation, and distribution shape of email traffic, and interpret the implications for system design or user productivity. Comment on the shape of the distribution and central tendency. Determine the probability that more than 20 emails are received on a given day. Either either data or import .csv Sample Data <table><tr><th>Day</th><th>Emails_Received</th></tr><tr><td>1</td><td>15</td></tr><tr><td>2</td><td>12</td></tr><tr><td>3</td><td>18</td></tr><tr><td>.....</td><td>.....</td></tr></table>	Day	Emails_Received	1	15	2	12	3	18									
Day	Emails_Received																		
1	15																		
2	12																		
3	18																		
.....																			
		Exploratory Data Analysis on Web User Activity Data Explore the distribution and patterns in user session durations based on age group and device type to gain insights for web optimization. Compare average session times between age groups and also analyze device influence. Sample data. Either manually type in Jamovi or Load .csv file <table><tr><th>User_ID</th><th>Session_Duration</th><th>Age</th><th>Device_Type</th></tr><tr><td>1</td><td>5</td><td>18-25</td><td>Mobile</td></tr><tr><td>2</td><td>10</td><td>26-35</td><td>Desktop</td></tr><tr><td>3</td><td>12</td><td>18-25</td><td>Mobile</td></tr></table>	User_ID	Session_Duration	Age	Device_Type	1	5	18-25	Mobile	2	10	26-35	Desktop	3	12	18-25	Mobile	
User_ID	Session_Duration	Age	Device_Type																
1	5	18-25	Mobile																
2	10	26-35	Desktop																
3	12	18-25	Mobile																

		4	8	36-45	Tablet																																	
		5	15	26-35	Desktop																																	
		6	20	18-25	Mobile																																	
		7	18	26-35	Desktop																																	
		8	25	36-45	Tablet																																	
		9	6	18-25	Mobile																																	
		10	35	26-35	Desktop																																	
		Correlation Analysis of Coding Practice vs. Exam Scores																																				
		A computer science instructor wants to understand the relationship between the number of hours students practice coding and their final exam scores in a programming course. Additionally, the instructor wants to analyze if the rank order of students by coding hours correlates with their rank order by exam scores. Plot Scatter Diagram. Interpret Scatter Diagram, Pearson's r, Spearman's rho																																				
		Sample dataset																																				
		<table><tr><th>Student_ID</th><th>Coding_Hours</th><th>Exam_Score</th></tr><tr><td>1</td><td>5</td><td>55</td></tr><tr><td>2</td><td>12</td><td>78</td></tr><tr><td>3</td><td>8</td><td>65</td></tr><tr><td>4</td><td>15</td><td>85</td></tr><tr><td>5</td><td>7</td><td>62</td></tr><tr><td>6</td><td>10</td><td>70</td></tr><tr><td>7</td><td>3</td><td>50</td></tr><tr><td>8</td><td>20</td><td>90</td></tr><tr><td>9</td><td>6</td><td>58</td></tr><tr><td>10</td><td>11</td><td>75</td></tr></table>				Student_ID	Coding_Hours	Exam_Score	1	5	55	2	12	78	3	8	65	4	15	85	5	7	62	6	10	70	7	3	50	8	20	90	9	6	58	10	11	75
		Student_ID	Coding_Hours	Exam_Score																																		
		1	5	55																																		
		2	12	78																																		
		3	8	65																																		
		4	15	85																																		
		5	7	62																																		
6	10	70																																				
7	3	50																																				
8	20	90																																				
9	6	58																																				
10	11	75																																				

		Predicting Student’s Programming Exam Score Based on Coding Practice Hours Using Linear Regression For the above problem to find a linear relationship between practice hours (independent variable) and exam score (dependent variable). Predict the daily temperature using weather historical data. Key influencing factors considered are: ● Humidity ● Wind Speed ● Atmospheric Pressure Build a multivariate linear regression model using Jamovi to estimate the temperature and interpret the results. Sample Data <table><tr><th>Day</th><th>Humidity</th><th>Wind_Speed</th><th>Pressure</th><th>Temperature</th></tr><tr><td>1</td><td>60</td><td>15</td><td>1012</td><td>32</td></tr><tr><td>2</td><td>70</td><td>12</td><td>1010</td><td>30</td></tr><tr><td>3</td><td>55</td><td>20</td><td>1015</td><td>34</td></tr><tr><td>4</td><td>80</td><td>10</td><td>1008</td><td>28</td></tr><tr><td>5</td><td>65</td><td>18</td><td>1011</td><td>31</td></tr><tr><td>6</td><td>50</td><td>22</td><td>1016</td><td>35</td></tr></table>	Day	Humidity	Wind_Speed	Pressure	Temperature	1	60	15	1012	32	2	70	12	1010	30	3	55	20	1015	34	4	80	10	1008	28	5	65	18	1011	31	6	50	22	1016	35
Day	Humidity	Wind_Speed	Pressure	Temperature																																	
1	60	15	1012	32																																	
2	70	12	1010	30																																	
3	55	20	1015	34																																	
4	80	10	1008	28																																	
5	65	18	1011	31																																	
6	50	22	1016	35																																	

Sample Questions

Module 1

Let the vector $\vec{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

A transformation is applied using the matrix

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- Find the result of the transformation A. $\vec{v}$
 - Describe what happens to the direction and length of the vector after transformation.
- How does the **sign** (positive, zero, or negative) of the dot product relate to the **angle between vectors**? Why is dot product important when comparing two directions, such as the direction of movement of an object and the direction of a force applied to it?
 - How does scalar multiplication affect the **magnitude** and **direction** of a vector?
 - A car's position over time is given by a graph. The slope at any point on the graph represents its speed. What does a **positive**, **zero**, or **negative** slope indicate about the car's motion?
 - $f(x,y)=x^2+y^2$ represents the elevation of a surface. What does $\partial f/\partial x$ represent in this case?
 - The graph of a function shows the rate at which water flows into a tank. What does the **area under the curve** represent over a time interval?

Module 2

- In modular arithmetic, $17 \equiv 5 \pmod{12}$. What does this mean in simple terms?
- Use the Euclidean Algorithm to find $\gcd(48,18)$. What does the result tell you about whether a modular inverse exists for these numbers?

Module 3

- Explain what it means for two events to be mutually exclusive with an example.

- 2) Are the events rolling a 2 and rolling an even number mutually exclusive? Why or why not?
- 3) Why is **variance** important in understanding data spread?
- 4) Describe one real-life scenario where the **binomial distribution** is useful.

Module 4

1. You are given the following dataset representing the scores of 12 students in a programming test:

56, 72, 68, 45, 89, 77, 54, 62, 70, 80, 66, 75, 56, 72, 68, 45, 89, 77, 54, 62, 70, 80, 66, 75, 56, 72, 68, 45, 89, 77, 54, 62, 70, 80, 66, 75

Calculate the mean, median, and mode of the data.

Find the range, variance, and standard deviation of the scores.

Interpret the results:

- d) What does the mean score tell you about the overall performance?
- e) How does the median compare with the mean, and what might that indicate about the distribution?
- f) What does the standard deviation tell you about the consistency of student scores?
- g) Are there any signs of skewness or outliers from your calculations or frequency distribution?

- 2) You have collected data on the number of hours studied and the marks obtained by 8 students in an exam:

Student	Hours-Studied (X)	Marks-Obtained (Y)
1	2	50
2	3	55
3	5	65
4	4	60
5	6	70
6	8	85
7	7	75

8	9	90
---	---	----

Plot a scatter diagram of the data (Hours Studied vs Marks Obtained).

Based on the scatter plot, describe the nature of the relationship between hours studied and marks obtained.

Without calculating the exact value, would you expect the Pearson correlation coefficient to be close to +1, 0, or -1? Explain why.

2) For the given dataset, Regression Line is $Y=105+15X$

Day (X)	No of UsersVisiting a Website (Y)
1	120
2	135
3	150
4	165
5	180
6	195
7	210

a) What is the meaning of the slope $b=15$ in this context?

b) Interpret the intercept $a=105$ what does it represent in this scenario?

c) Using the regression line, predict the number of users visiting the website on day 8.

d) If the website team wants to reach 300 users, on which day does the regression model predict this will happen?

e) How would you assess whether this linear model is a good fit for the data?

Essential Readings:

1. Introduction to Linear Algebra, Gilbert Strang, Wellesley-Cambridge Press, 6th Edition
2. An Introduction to Mathematical Cryptography by Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman
3. Manish Sharma, Amit Gupta, The Practice of Business Statistics, Khanna Book Publishing Company, 2010
4. Ross Sheldon M., Introduction to Probability and Statistics for Engineers and Scientists, 6th Edition, Elsevier, 2021.

Suggested Readings:

1. Mathematics for Machine Learning, Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong, Cambridge University Press., 2020
2. Pal Nabendu and Sarkar Sahadeb, Statistics: Concepts and Applications, Second Edition, PHI, 2013

Web Resources

1. <https://nptel.ac.in/courses/111106112>
2. <https://nptel.ac.in/courses/111105041>
3. <https://www.youtube.com/playlist?list=PLblh5JKOoLUK0FLuzwntyYI10UQFUhsY9>
4. <https://www.youtube.com/playlist?list=PLblh5JKOoLUK0FLuzwntyYI10UQFUhsY9>
5. <https://www.khanacademy.org/math/linear-algebra>
6. <https://www.3blue1brown.com>

Assessment Rubrics:

Evaluation Type	Marks
End Semester Evaluation	65
Theory	50
Practical	15
CCA	35

Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU3DSCCAP205: LINUX SYSTEM ADMINISTRATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200-299	KU3DSCCAP205	4 (3T+ 1 P)	75

Learning Approach (Hours/Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1.5 Hrs

Course Description: The primary benefit of Linux is its open-source distribution, which allows programmers to create their own unique distribution of Linux OS. C Programming language is used to write the majority of Linux code. The majority of web servers, smartphones, laptops, supercomputers, and cloud servers are powered by Linux due to its excellent security, reliability, and open-source nature. A Linux system administrator has a very critical role in managing and maintaining Linux-based systems

Course Prerequisite: KU1DSCCAP101 : Foundations of Computers and Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
--------	------------------	-----------------

1	To learn basic Linux commands and understand the file system structure	A
2	To understand the Boot loaders and the configuration files	U, An
3	To learn different system services, maintenance and configuring these	U
4	Understand and develop Shell Scripting	A, C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	2	2					2
CO2	2	2					
CO3	2	2					2
CO4	3	2	2		2	3	2

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE: Introduction to Linux OS and commands		
	1	Intro to Open Source free Software, History & Features of Linux	11
	2	Basic Linux Commands: Currently logged-in user: whoami, w, logname, who, id, uname. Managing Files & Directories: ls, touch, cat, rm, mv,cp, mkdir,cd, pwd. Other commands: man, grep, more, sort, wc, du, df, cal, date, bc	
	3	Files and Directories, Types of files, directory tree, relative and absolute pathnames, referring home directory	
	4	File permissions: user,group,ls (long listing), changing file permissions (chmod and umask commands)	

	5	Types of users in Linux, creating a user account (adduser or useradd)	
II	MODULE TITLE: Shell Scripting		
	1	Introduction to Shell and Types of shell	12
	2	Basic shell configuration files for Bourne and bash shell: /etc/profile, /etc/bashrc, ~/.bash_profile, ~/.bashrc, ~/.bash_history.	
	3	Bourne shell scripts, script execution, variables and parameters, shell environment variables	
	4	Control structures - if then else, if then elif, for loop, while loop, until loop , case	
III	MODULE TITLE: Linux Operating System Management		
	1	Boot process: LILO - boot process, /etc/lilo.conf file, GRUB - /etc/grub.conf file	12
	2	Brief Introduction to Run levels.	
	3	Mounting: mounting file systems, structure of /etc/fstab, mount, umount commands	
	4	Periodic command execution: at and cron, crontab file	
	5	Starting and stopping different services – service command and systemctl command	
IV	MODULE TITLE: System Maintenance		
	1	Backup and Restore: types of backup - full, differential, incremental	10
	2	Backup and Restore commands: cpio, tar commands.	
	3	Linux OS Installation: Partitioning - SWAP, root, boot, EFI partitions and mount points	
	4	Installation and uninstallation of software packages in a Linux system (any distribution of Linux)	

V	LAB EXPERIMENTS		
	LINUX ADMINISTRATION - Linux installation, upgradation and rescue. - Boot loader configuration using GRUB - The service command - Managing process- viewing status, killing, restarting etc using ps. - Adding and deleting user accounts, changing passwords		30

	6. Scheduling jobs using cron 7. Mounting and unmounting external file systems 8. Setting the value of umask, changing the permissions, owner and groups 9. Installation and removal of packages 10. Archiving and Backup using tar. Restoring backup 11. Compressing and decompressing files using any one tool SHELL Programming 1. Get a name and number from the user, create a file with that name and number. Also display the contents of the file. - o If the name is XXX and number is 2 the filename must be XXX_2. use cat command to create a file. Create the file with 10 different lines, then display the first 5 lines of the file using head command. 2. Write a program to greet a user by 'Good Morning', 'Good Afternoon' or 'Good Evening' based on time - o get the system time using the 'date' command. Read the name from the user, if the name is 'XXX' then greet with 'Hello XXX, Good Morning! ' 3. Write a shell program to check whether a number is positive, negative or zero 4. A program to create 10 users. Use loop structure, get usernames from the user and assign same password to all the users 5. A demo program to test different file operators. Read filename from the user. Check if the file exists, if it exists then display the contents, otherwise create the file. Check whether the size of the file is zero, check whether the file is having read, write and execute permission	
--	---	--

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

1. Yashavant Kanetkar, UNIX Shell Programming, BPB
2. Eleen Frisch, Essential System Administration, 3rd Edition, O'Reilly Media

Assessment Rubrics:

Evaluation Type	Marks
End Semester Evaluation	65
Theory	50
Practical	15
CCA	35

Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU3DSCCAP206: MOBILE APPLICATION DEVELOPMENT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200-299	KU3DSCCAP206	4 (3T+ 1 P)	75

Learning Approach (Hours/Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/Internship	Tutorial	CE	ESE	Total	1.5 Hrs
3	2	-	35	65	100	

Course Description: Mobile App Development is a course designed to provide students with a comprehensive understanding of the tools, and techniques involved in creating mobile applications for android platforms. The course aims to equip students with both the theoretical knowledge and practical skills necessary to embark on a career in app development. The curriculum is structured to cover key aspects of the app development lifecycle, user interface (UI) design, simple app designs and implementation

Course Prerequisite: KU1DSCCAP101: Foundations of Computers and Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domain
1	Understand Mobile App Development Concepts	U
2	Develop Programming Proficiency and Explore Mobile App Architecture	C

3	Design User-Friendly Interfaces configuring these Apps	C
4	Manage Data in Mobile Apps	A

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO1	2	2	3	2	2		2
CO2	3	3	3	3	3	2	2
CO3	3	3	3	3	3	2	2
CO4	3	3	3	3	3	2	2

COURSE CONTENTS

MODULE	UNIT	DESCRIPTION	HOURS
I	MODULE TITLE: Introduction to Mobile App development and Getting an Overview of Android		
	1	Overview of mobile app development	11
	2	Understanding different mobile platforms (iOS, Android)	
	3	Introducing Android Studio: Android APIs, Android Architecture Application Framework, Application components	
	4	The Manifest File, ANDROID Environment Setup: Steps for setting the environment, installing the SDK.	
II	MODULE TITLE: Using Activities, Fragments, and Intents, Layouts in Android		
	1	Working with Activities: Creating and Starting an activity, Lifecycle of an activity, Applying themes and Styles on Activity, Displaying a dialog, hiding the title of the activity	12
	2	Using Intents: Exploring Intent objects, resolution and filters. Linking the activities using Intent. Passing the data using an intent object.	

	3	Fragments: Fragments Implementation. Finding Fragments, Adding, removing and replacing fragments. Finding Activity using fragment	
	4	Layouts: Linear Layout, Relative Layout, Scrollview Layout, Table Layout, Frame Layout and TabLayout	
III	MODULE TITLE: Working with the User Interface Using Views and View Groups		
	1	TextView, EditText View	12
	2	Button, Radio Button, CheckBox, ImageButton, ToggleButton, Rating Bar	
	3	ListView, Gallery View, AutoText Complete	
	4	User interaction with Views and Activities	
IV	MODULE TITLE: Handling Pictures and Menus with Views and data storage		
	1	Creating Menus: options Menu, Context Menu, Sub menu	10
	2	Displaying Images in the Gallery and Grid View	
	3	Using the Image switcher view	
	4	Notifying the user: Toast, Status Bar, Dialog Notification	
V	LAB EXPERIMENTS		
	1. Android: Environment Setup, Implementation of Android feature in real time application 2. Demonstration of Android – UI Layouts, UI Controls & Event Handling 3. Create two activities. Launch the second activity from the first using a button click. Apply a custom theme to an activity and hide its title bar. 4. Create two activities, First activity collects user data with views, Second activity receives and displays data 5. Show an alert dialog with Yes/No buttons when a button is clicked. Send user input from Activity A to Activity B and display it. 6. Create a fragment and add it to an activity. Replace it with another fragment using a button. 7. Create a form with TextView labels, EditText input fields (name, email), and a Submit Button. On clicking the button, show a Toast with entered data. Add the features where the user		30

	selects gender (RadioButton), hobbies (CheckBox), enable/disable notifications (ToggleButton), and gives a rating (RatingBar). 8. Use buttons to show a simple Toast, an AlertDialog with Yes/No, a system Notification in the status bar 9. Load and display a set of drawable images in a Gallery and a GridView. 10. Execution of Android – adding pictures and menus and using notifications	
--	---	--

Essential Readings (Books, Journals, E-sources Websites/ weblinks)

1. Android Application Development (With Kitkat Support), Black Book Kindle Edition by Pradeep Kothari
2. Jerome DiMarzio, Beginning Android Programming with Android Studio, 4th Edition.
3. Android Studio 4.0 Development Essentials – Kotlin Edition by Neil Smyth:
4. <https://developer.android.com/guide>
5. <https://developer.android.com/codelabs/basic-android-kotlin-compose-first-app#0>
6. Android Programming: The Big Nerd Ranch Guide by Bill Phillips and Chris Stewart

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5

c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU4DSCCAP207: SOFTWARE ENGINEERING

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
4	DSC	200 – 299	KU4DSCCAP207		4(4T+0P)	60
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	--	--	30	70	100	2

Course Description: This course targets to expose the students to the challenges of large-scale software development and would familiarise them as to how to overcome those. Starting with basic life cycle model concepts, it would discuss requirements, specification, design specifically object-oriented design, testing issues and Software Quality standards.

Course Prerequisite: KU1DSCCAP101: Foundations of Computers and Programming
KU2DSCCAP106 : Programming with C and C++

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic processes in the software development lifecycle.	U
2	Familiarise yourself with different SDLC models and their significance.	U
3	Gather, analyse and specify software requirements effectively	A
4	Design software systems using modular design principles and object-oriented approaches.	A
5	Develop and execute effective testing strategies to ensure software quality.	A, C
6	Familiarise SQA and important Quality Standards	U

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)***

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1				2			
CO 2	2	2	2				
CO 3	2	2		2		2	
CO4							
CO5					2		
CO 6		2				2	

COURSE CONTENTS

Contents for Classroom Transaction:

MODULE	UNIT	DESCRIPTION	HOURS
1	MODULE TITLE -Introduction to Software Engineering		

	1	a)Software Engineering Definition ,Evolution,	12
		b)Software Development Projects	
		c)Exploratory style of software development	
		d) Emergence of software Engineering	
		e) Computer systems engineering	
		f) program versus software, software process, software characteristics.	
	2	a)Software Life Cycle Models- Basic Concepts	
		b)Waterfall model and its extensions	
	3	a)Rapid Application Development Model	
		b)Spiral model	
	4	a)Agile development Models	
		b)Basic Idea of Extreme programming Model	
		c)Basic Idea of Scrum Model	
		d)Comparison of Different life cycle models	
		e) Selecting an appropriate life cycle model for a project CASE STUDY - PART1	
2	MODULE TITLE- Requirements Analysis and Specification and Basics of Software Design		12
	1	a) Overview -Requirements Engineering b) Requirements gathering /elicitation c) Requirements Analysis	
	2	h) Software Requirements Specification (upto 4.2.10)	
	3	a) Software Design- Overview b) How to characterize a good software design c) Cohesion and Coupling d) Layered arrangement of modules e) Approaches to software design	

	4	a) Function Oriented Software Design -Introduction, Overview of SA/SD Methodology b) Structured analysis c) Structured Design d) Detailed Design, Design Review. CASE STUDY- Part 2	
3	MODULE TITLE- UML and Object Oriented Software Development		
	1	a) Object Modelling using UML-Basic object Oriented Concepts a) UML- Origin of UML, Evolution, b) UML Diagrams	12
	2	a) Use Case Model (Upto 7.4.3) b) Class Diagrams, Object diagrams, c) Interaction diagrams, Activity Diagrams, State Chart Diagrams, d) Package, Component and Deployment Diagrams, UML 2.0	
	3	a) Object Oriented software Development- OOA Vs.OOD b) Patterns c) Some common design patterns- Model View Controller Pattern, Publish-Subscribe Pattern, Intermediary pattern	
	4	a) OOAD Methodology- Unified process Model b) Applications of Analysis and Design Process c) OOD Goodness Criteria CASE STUDY -Part 3	
MODULE TITLE- Coding, Testing and SQA			
	1	a) Coding b) Coding Standards and Guidelines c) Code Review d) Software Documentation	

	2	d) Testing - Basic Concepts and Terminologies, Verification Vs. Validation, Testing Activities, Why design Test Cases, e) Testing in the Large Vs. Small, Unit Testing, BlackBox Testing, White Box testing, f) Debugging, Program analysis Tools, Integration Testing, g) Testing Object oriented Programs, System Testing -- Some General Issues Associated with testing - Testing Documentation, Regression Testing	
	3	a) Software Reliability b) Statistical Testing	
	4	1) Software Quality 2) Software Quality Management System 3) ISO 9000(Upto 11.5.4), SEI CMM, CMMI 4) Few Other Important Quality Standards,Six Sigma 5) CASE STUDY- Part 4 & Presentation	
5	Teacher Specific Module		
	Familiarise Tools for Software Engineering		12
	Emerging Trends in Software Engineering		

Guidelines to conduct Case Study

1. Choose any one of the case studies (Examples are given below)

1. Student Marks Analysis System
2. Library Management System
3. E-Commerce Website
3. Inventory Control System
4. Food Delivery Management system
5. Logistics Management System

2. Do the following exercises for that Case Study. Use appropriate software tools for each one.
 1. Write the complete problem statement
 2. Write the software requirements specification document
 3. Draw the entity relationship diagram
 4. Draw the data flow diagrams
 5. Draw use case diagrams
 6. Draw activity diagrams for all use cases
 7. Draw sequence diagrams for all use cases
 8. Draw collaboration diagram
 9. Assign objects in sequence diagrams to classes and make class diagram.
 10. Write sample code.
 11. Write test cases and test results after unit testing.

Essential Readings:

1. Rajib Mall, Fundamentals of Software Engineering, Fourth edition, PHI Learning Private Limited (For all modules except given in Essential readings:No. 2)
2. K. K. Aggarwal, Yogesh Singh, Software Engineering, 3rd Ed, New Age, International Publication (For Modules 1.1.b, 2.1.a, 2.1.b and 2.2.d only)

Suggested Readings:

1. Ian Sommerville, Software Engineering, 10th Ed, Pearson
2. Roger S Pressman, Software Engineering: A Practitioner's Approach, 6th Ed, TMH

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Case study	10
Total		100

KU4DSCCAP208: DATABASE MANAGEMENT SYSTEM

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSC	200-299	KU4DSCCAP208	4 (3T+ 1 P)	75

Learning Approach (Hours/ Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/Internship	Tutorial	CE	ESE	Total	1.5 Hrs
3	2	-	35	65	100	

Course Description: Databases are the backbone of almost all the digital services and e-governance solutions. Modern businesses and financial systems heavily depend on databases systems and transaction processing for their successful operation. This course introduces the students to the various theoretical and practical principles involved in the design and use of database systems with the help of database management systems (DBMS) and SQL.

Course Prerequisite: KU1DSCCAP101: Foundations of Computers and Programming ,
KU3DSCCAP201: Discrete Structures

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand fundamental database concepts and architecture	U
2	Ability to construct an ER model and derive the relational schemas from the model.	U, C
3	Apply normalization techniques to design efficient database schemas	A, C
4	Construct efficient SQL queries to retrieve and manipulate data as required	An, C
5	To apply procedural programming concepts by developing PL/SQL blocks	A, C
6	Introduce advanced DBMS topics and trends	U

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©***

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1					2		
CO 2		2		2		2	
CO 3	2					2	2
CO 4		2			2	2	
CO 5		2		2	2		
CO 6				2			

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Introduction & ER Model		12
	1	Overview of DBMS, Characteristics of Database system, Database Users	
	2	Data Models and Schema	
	3	Three-Schema-architecture, Data Independence	
	4	Database Languages	

	5	Database architecture	
	6	ER model: basic concepts, entity set & attributes, notations	
		Relationships and constraints – cardinality, participation, notations	
		Weak entities	
	7	Extended ER diagram	
2	MODULE TITLE: Relational Database Design		11
	1	Structure of relational Databases, Integrity Constraints	
	2	ER to Schema synthesis	
	3	Need for normalisation, database anomaly	
	4	Functional Dependencies, Types, Closure, Finding keys	
	5	Desirable Properties of Decomposition, Dependency Preserving, Lossless Decomposition (Basic idea only)	
	6	First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Fourth Normal Form	
3	MODULE TITLE : Query Language		12
	1	Relational algebra: select, project, SET operations	
	2	Join - Equi-join, Natural join	
	3	Introduction to SQL, data types	
	4	Constraints: Primary key, Foreign Key, Unique, Not Null, Check	
	5	DDL: create, alter, drop, use of auto increment	
	6	DML: Insert, delete, update, select	
	7	Aggregate Functions	

	8	GROUP BY, HAVING Clauses	
	9	Data Sorting and Filtering, ORDER BY, GROUP BY, LIKE, IN, BETWEEN	
	10	SET operations	
	11	Joins: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN	
	12	Subqueries: Single-row and Multi-row	
4	MODULE TITLE: Advanced SQL, PL/SQL Programming and NoSQL Databases		10
	1	Views: Creating and Using Views	
	2	Introduction to Transactions: COMMIT, ROLLBACK	
	3	Creating user, Access control, DCL, grant, revoke	
	4	Introduction to SQL security: SQL Injection (Basic Idea only)	
	5	Introduction to PL/SQL, Declaration	
	6	Procedures and Functions, IN, OUT, IN OUT parameters	
	7	Cursors	
	8	Triggers	
	9	Exception Handling	
	10	Structured, Unstructured, semi structured data	
	11	NoSQL Databases: Introduction, properties of NoSQL Databases, types of NoSQL database	
	Teacher Specific Module		
5	Lab Experiments		30
	Set up Lab: Install MySQL		

	Case Study: A group activity, a group of maximum 3 can be made. Identify a DBMS case study. Write the problem statement. Draw the E-R diagram. Convert ER Model to Relational Model. Apply normalization to remove the redundancies and anomalies in the above relational tables, normalize up to Third Normal Form.	
	Frame and execute suitable SQL queries 1. Create table Department with fields Dept_no (primary key), Dept_Name, (unique), location (Check) 2. Create table Employee with fields Emp_no (primary key), Emp_Name, Salary default 5000, Dept_ID refer Dept_No in Department table and on delete set null, Dob not null 3. Create table Dependent with fields Dep_id, Dept_Name and Emp_id with references Emp_no in Employee and on delete cascade Create the above tables. Define the PRIMARY KEY, FOREIGN KEY, NOT NULL, DEFAULT, CHECK and UNIQUE constraints wherever appropriate. 4. Display the description of Employee table 5. Insert records by satisfying constraints 6. Select all records from the Employee table 7. Delete the records of all employees in Dept_No 6. Delete records by satisfying constraints 8. Add attributes HireDate, Address, Designation in Employee table 9. Delete the attribute Address 10. Update designation details of Employees 11. List employees whose salary greater than 30000 12. Display annual salary of all employees 13. Rename HireDate to Hire_Date 14. Update the Employee table by giving an increment of 500 to manager 15. Write a query to change the salary of an employee to 80000 whose ID is 105, if the existing salary is less than 50000.	

	Implementation of various aggregate functions in SQL 1. Find the number of employees 2. Find the average salary of all employees. 3. Find the highest salary of all employees. 4. Find the minimum salary. 5. Display total salary of all employees 6. Count the distinct employee designations	
	Implementation of Group By, Having, ORDER BY, LIKE, IN, BETWEEN operators 1. Find the number of employees in each department 2. Display the total number of dependents for each employee 3. Display average employee salary by department 4. Find the dept_no where the average salary of all employees is more than 1500. 5. Display the total number of dependents for each employee 6. Display the total number of dependents for each employee for employees who have at least two dependents. 7. Display employee names in descending order. 8. Find employees whose name start with N 9. Find employees whose name contains thra 10. Display employee name, salary where salary is between 10000 and 20000	
	Implementation of Join 1. List employees with their department names 2. Show employees with their dependents 3. Count of employees in each department 11. Display the names of employees in CS department	
	Implementation of SUBQUERIES 1. Find employees who earn more than the average salary 2. List employees who have dependents	

	- Find employees who work in the same department as employee with emp_id = 101 - Find employees whose salary is greater than salary of employee named 'Mizhi - Display departments with no employees'	
	6. Implement SET operations	
	Creation of Views - Create a view showing only employees with salary greater than 5000 - Create a view with average salary per department - Create a view with Employee name and department name	
	Creating user, Access control, DCL, grant, revoke	
	PL/SQL Programming - Create a function to count employees in a given department - Create a MySQL function named check_salary_status that takes an employee ID as input and returns a string: - High: if the employee's salary is greater than 5000, - Medium: if the salary is between 3000 and 5000 (inclusive), - Low: if the salary is less than 3000 Create a MySQL AFTER UPDATE trigger on the employee table to log changes into an employee_audit table. Whenever an employee's record is updated, log the ID, name, salary, and the time of update into employee_audit	
	3. Install and Configure MongoDB to execute NoSQL Commands.	
	Frame and execute suitable SQL queries for your case study selected at the beginning.	

Essential Readings:

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan, 7th Edition, McGraw-Hill Education, 2010.
2. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition

Suggested Readings:

1. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, Third Edition, McGraw-Hill
2. https://onlinecourses.nptel.ac.in/noc22_cs91/preview

Assessment rubrics

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case Study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU4DSCCAP209: DATA STRUCTURES AND ALGORITHMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSC	200-299	KU4DSCCAP209	4 (3T+ 1 P)	75

Learning Approach (Hours/Week)		Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/Internship	Tutorial	CE	ESE	Total	1.5 Hrs
3	2	-	35	65	100	

Course Description: The objective of the course is to introduce the fundamentals of Data Structures, Abstract concepts and how these concepts are useful in problem solving. This course provides a basic understanding of algorithms and techniques to compare different algorithms that solve the same problem. The course also introduces fundamental data structures used in computer science related problems. The algorithms introduced in the course may be implemented in the lab using C.

Course Prerequisite: KU1DSCCAP101, KU2DSCCAP106, KU3DSCCAP111

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of Data Structures and their applications.	U
2	To introduce the basic concepts of algorithms	U
3	Implement Data Structures using C/ C++ programming language	A, C
4	Use the appropriate data structure in context of solution of given problem	An, C
5	Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs	A, C

6	Understanding various searching & sorting techniques	U, A, An
---	--	----------

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1					2		
CO 2					2		
CO 3	2	2				2	
CO 4				2	2	2	
CO 5	2		2				2
CO 6				2	2		

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	INTRODUCTION		11
	1	Data types – primitive and non-primitive, Definition of data structure, data structure operations. Types of Data Structures- Linear & Non-Linear Data Structures.	
	2	Algorithms : Complexity, Time Space tradeoff -- Average, best and worst case of algorithms. Complexity of Algorithms, Asymptotic Notations for Complexity of Algorithms	

	3	Arrays: Definition and Classification of Arrays, Representation of Linear Arrays in Memory, Operations on Linear Arrays: Traversing, Inserting, Deleting, Searching, Sorting and Merging	
	4	Two-Dimensional Arrays, Representation of Two-Dimensional Arrays in Memory, Matrices and Sparse Matrices, Multi-Dimensional Arrays.	
	5	Searching: Linear Search and Binary Search, Comparison of Methods.	
2	SORTING, LINKED LISTS AND HASHING		12
	1	Sorting: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort	
	2	Linked Lists: Definition, Comparison with Arrays, Representation, Types of Linked lists, Traversing, Inserting, Deleting and Searching in Singly Linked List, Doubly Linked List and Circular Linked List. Applications of Linked Lists: Addition of Polynomials.	
	3	Hashing and Collision: Hashing, Hash Tables, Types of Hash Functions, Collision, Collision Resolution with Open Addressing and Chaining	
3	STACK AND QUEUE		12
	1	Stacks: Definition, Representation of Stacks using Arrays and Linked List Operations on Stacks using Arrays and Linked List Application of Stacks: Arithmetic Expressions, Polish Notation, Conversion of Infix Expression to Postfix Expression, Evaluation of Postfix Expression.	
	2	Recursion: Definition, Recursive Notation, Runtime Stack, Applications of Recursion: Factorial of Number, GCD, Fibonacci Series and Towers of Hanoi.	
	3	Queues: Definition, Representation of Queues using Array and Linked List Types of Queue: Simple Queue, Circular Queue, Double-Ended queue, Priority Queue,	

		Operations on Simple Queues and Circular Queues using Array and Linked List, Applications of Queues.	
4	GRAPHS AND TREES		10
	1	Graphs: Definition, Terminology, Representation, Traversal—DFS and BFS	
	2	Trees: Definition, Terminology, Binary Trees, Traversal of Binary Tree, Binary Search Tree, Inserting, Deleting and Searching in Binary Search Tree Height Balanced Trees: AVL Trees, Insertion and Deletion in AVL Tree.	
5	TEACHER-SPECIFIC MODULE: LAB EXPERIMENTS USING C		30
	1. Write a program for insertion and deletion operations in an array. 2. Write a program to search for an element in an array using Linear Search and Binary Search. 3. Write a program to sort an array using Bubble Sort, Selection Sort and Insertion Sort. 4. Write a program to perform Quick Sort and Merge Sort. 5. Write a program to add and subtract two matrices. 6. Write a program to multiply two matrices. 7. Write a program to insert an element into a Singly Linked List: i) At the beginning ii) At the end iii) At a specified position 8. Write a program to delete an element from a Singly Linked List: i) At the beginning ii) At the end iii) A specified element 9. Write a program to perform the following operations in a Doubly Linked List: i) Create a DLL and Search for an element		

	10. Write a program to perform the following operations in a Circular Linked List: i) Create ii) Delete an element from the end 11. Write a program to implement stack operations using an array. 12. Write a program to implement stack operations using a linked list. 13. Write a program to add two polynomials using linked lists. 14. Write a program to evaluate a postfix expression using a stack. 15. Write a program to perform the following using recursion: i) Find the factorial of a number ii) Find the GCD of two numbers 16. Write a program to implement simple queue operations using an array. 17. Write a program to implement circular queue operations using an array. 18. Write a program to implement circular queue operations using a linked list. 19. Write a program to do the traversal operations on a binary search tree. i) Preorder Traversal ii) Inorder Traversal iii) Postorder Traversal 20. Write a program to perform insertion operations in a binary search tree.	
--	--	--

Essential Readings:

1. Debasis Samanta, Classic Data Structures, 2nd Ed, PHI
2. G. A. V. Pai, Data Structures and Algorithms: Concepts, Techniques and Applications, 1st Ed, TMH

Suggested Readings:

1. Yashavant Kanetkar, Data Structures Through C, 4th Edition, BPB Publications, 2022.
2. Ellis Horowitz, Sartaj Sahni and Dinesh Mehta, Fundamentals of Data Structures in C++, 2nd Ed, Universities Press

Assessment rubrics

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU5DSCCAP301: PYTHON PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSC	300-399	KU5DSCCAP301	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description:

This course provides a comprehensive introduction to Python programming and its applications in modern software development. The course begins with the fundamentals of Python, including syntax, data types, operators, control structures, functions, modules, and built-in data structures. It further explores object-oriented programming concepts, exception handling, file handling, and database connectivity for developing reliable and structured applications.

The course also introduces scientific computing and data visualization using NumPy and Matplotlib. Students gain practical exposure to developing interactive graphical user interfaces and lightweight web applications using Streamlit and Flask with database integration and session management. In addition, the course emphasizes software development practices such as modular programming, virtual environments, and version control using Git and GitHub.

By the end of the course, students will be able to design, develop, and deploy Python-based applications for real-world problem-solving and project-oriented software development.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand and apply the fundamentals of Python programming including data types, control structures, functions, modules, and data structures to develop basic programs.	A
2	Apply object-oriented programming concepts, exception handling, file handling, and database connectivity to develop structured and reliable Python applications.	A/An
3	Perform numerical computation, data analysis, and visualization using NumPy and Matplotlib.	An
4	Develop interactive GUI and lightweight web applications using Streamlit and Flask integrated with databases and session management.	C
5	Apply software development practices including modular coding, virtual environments, and version control using Git and GitHub for project-oriented software development.	C

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)***

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	3	2	1	1	1	2
CO 2	3	3	3	2	1	1	2
CO 3	2	2	3	2	2	1	2
CO 4	3	3	3	2	2	2	3
CO 5	3	3	2	3	3	2	3

COURSE CONTENTS

MODU LE	U NI T	DESCRIPTION	HOURS
1	MODULE TITLE: Basics of Python		11
	1	Introduction to Python, features, Installing Python and setting up development environment, use of pip, Running python programs	
	2	Basics of Python-Variables and Data Types, Keywords, Mutable and Immutable objects, Operators and Expressions,Using Comments, docstring, Input and Output in Python, Working with strings , Formatting output,fstring.	
	3	Data Structures in Python-working with Lists, Tuples, Sets and Dictionaries-operations and methods.	
	4	Conditional statements, Loops, List comprehension, Dictionary comprehension. Range and enumerate functions.	
2	MODULE TITLE-Functions, OOP and other Concepts		10
	1	Functions- define and call functions, return statements, different types of arguments and scope and lifetime, Lambda functions, Modules and packages.	
	2	Object-Oriented Programming (OOP) Basics-Classes and objects, Constructors and destructors,OOP Concepts - Inheritance, Polymorphism, Encapsulation and abstraction, Method overriding, Operator overloading	
	3	Exception Handling- Types of errors, Built-in Exceptions -IndexError, OverflowError, ZeroDivisionError,RuntimeError, try, except, else, finally, raise statement, User-defined exceptions,Assertions.	
	4	File handling-Text files,CSV files-Open, Close, Read and write only, with statement, File Handling exceptions.	

3	MODULE TITLE -Python for Data Analysis and Visualization		12
	1	Data Processing using NumPy- ndarray, dtype, create arrays(array, zeros, ones, empty, linspace, arrange, random), Two-Dimensional Array, Indexing, Slicing, Shape Manipulation (reshape, transpose, resize), Arithmetic Operations on Arrays.	
	2	GUI Programming with streamlit-Installing and running streamlit programs. Widgets- text_input() ,number_input() ,button() ,selectbox() ,radio() ,checkbox() ,date_input() ,Password Input ,Messages-Success, Error, Warning, Displaying Tables ,Simple Log in form, Student Registration Form.	
	3	Data Visualization -Matplotlib module,pyplot, figure(), axes, subplot(), plot(),formatting a plot -title(),text(), xlabel(), ylabel(), legend(), grid(),savefig(), show(), grid(),Line chart,Bar chart,Scatter chart, Pie chart,Histogram,Plotting Simple Mathematical Functions (sin x, x2).	
4	MODULE TITLE- Software development using Python		12
	1	Database Connectivity- SQLite Or MySQL,CRUD operations.	
	2	Introduction to Flask as a Lightweight Web Framework for Python- Flask Architecture- Application Structure, Request Lifecycle, Application Context, Setting Up a Basic Flask Application, Configuring a Virtual Environment for Flask, Create Simple Application, Integration with sqlite, Manage sessions	
	3	Introduction to Software Development Practices -Modular Coding- importing modules, code reuse and project organisation,Virtual Environments,Git and GitHub Basics	
5	Teacher Specific Module :		30
	1. Write a program to count vowels, consonants, digits, and special characters in a string given as input. 2. Write a program to find the second largest element in a list. 3. Write a program to count word frequency using a dictionary. 4. Write a program to sort dictionary by keys and the by values		

	5. Write a program using list comprehension to generate prime numbers. 6. Write a program to generate Fibonacci series using loops. 7. Sort strings by length using lambda function. 8. Sort dictionary using multiple values and lambda function. 9. Write a program to demonstrate Inheritance and polymorphism. 10. Write a program to read and write text files .Use exceptions. 11. Plot simple mathematical functions. 12. Create simple login form using streamlit 13. Create student registration form using streamlit 14. Form Handling using Flask. 15. Flask and SQLite Integration -to store student data using Flask and SQLite. 16. To perform Create, Read, Update, Delete operations using Flask and SQLite. 17. Login System with Session Management . 18. Sample program to use Git and Github	
--	---	--

Essential Readings:

- 1) Reema Thareja, *Python Programming Using Problem Solving Approach*, Oxford University Press (India).
- 2) Mark Summerfield, *Programming in Python 3 : A Complete Introduction to the Python Language*, Second Edition
- 3) THE NORTHERN HIMALAYAS, *Mastering Python: A Journey Through Programming and Beyond*, THE NORTHERN HIMALAYAS
- 4) Sheetal Taneja & Naveen kumar: *Python Programming a Modular approach – A Modular approach with Graphics, Database, Mobile and Web applications*,
- 5) Venkatesh, Nagaraju Y, *Introduction to Python Programming*, Khanna Publishing House, 2021.

Suggested Readings:

- 1) Introduction to Computation and Programming using Python, by John Guttag, 2nd edition, 2016, PHI India.
- 2) An introduction to Python for absolute beginners, by Bob Dowling, Cambridge Univ.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar/Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSC	300-399	KU5DSCCAP302	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	0	30	70	100	2 Hours

Course Description: This course introduces the fundamental concepts of computer networks, including network architecture, reference models (OSI and TCP/IP), physical and data link layer functions, routing and transport protocols, and network performance metrics. It further extends to network security concepts such as secure communication protocols, cryptography basics, firewalls, intrusion detection systems, and secure network practices.

Course Prerequisite: Basic Computer Knowledge

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic concept of the physical layer and its role in data communication.	U
2	Analyze data link layer mechanisms, including error detection/correction and flow control protocols.	U,An
3	Apply network layer concepts such as routing algorithms, IP addressing, congestion control, and switching techniques.	A
4	Understand transport layer protocols (TCP/UDP) and their differences in communication.	U,An
5	Describe network security principles, including encryption, firewalls, and secure communication protocols.	U,An

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
--	-------	-------	-------	-------	-------	-------	-------

CO 1	2	1	1	2	1	2	2
CO 2	3	2	2	2	1	2	2
CO 3	3	3	3	2	2	2	2
CO 4	2	2	2	2	1	2	2
CO 5	3	2	3	3	2	3	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Fundamentals of Computer Networks		12
	1	Introduction – Uses of computer networks, Network hardware, Network software.	
	2	Reference models – The OSI reference model, The TCP/IP reference model, Comparison of OSI and TCP/IP reference models.	
	3	Physical Layer –Basics of digital & analog signals, Modes of communication, Physical topologies, Signal encoding, Repeaters and hub, Transmission media overview.	
	4	Performance indicators – Bandwidth, Throughput, Latency, Queuing time, Bandwidth	

2	MODULE TITLE : Datalink Layer		12
	1	Data link layer services, error-detection Types of errors, Single bit error and Burst error, Cyclic Redundancy Check, Check sum Error correction - Single bit error correction, Hamming code	
	2	Stop-and-Wait Protocol, Sliding Window Protocol : Go-Back- N and Selective Repeat.	
	3	MAC addressing, Address Resolution Protocol and Reverse Address Resolution Protocol	
	4	IEEE Standards- Wireless LANS, Ethernet, Bluetooth (Concept only)	

3	MODULE TITLE: Network and Transport Layer		12
	1	Services to the transport layer, Switching Techniques	
	2	Routing algorithms- adaptive, non-adaptive algorithms, Dijkstra's shortest path routing algorithm, distance vector routing, flow based routing, hierarchical routing	
	3	Congestion Control : Leaky Bucket, Algorithm and Token Bucket Algorithm.	
	4	IPv4 Addressing, Public vs Private IP Addresses, Subnet Masks, Default Gateway, Basic Subnetting Concepts, IP Classes & Classful Routing, Classless Inter-Domain Routing, Introduction to IPv6	
	5	Transport Layer Protocols : TCP and UDP, structure of header, Comparison of TCP and UDP.	
	6	Switching Techniques - Circuit, Packet, Message Switching.	

4	MODULE TITLE: Fundamentals of Network Security		12
	1	HTTP and HTTPS, DNS, DHCP, FTP, SMTP, POP3 and IMAP, SSH and Telnet, SNMP (Concepts only)	
	2	Definition of Network Security, Importance of Network Security, Security Goals, Threats and Vulnerabilities, Security Policies, Defense in Depth	
	3	Encryption Basics: Symmetric and Asymmetric encryption (Overview), Message Integrity and Digital Signatures (overview)	
	4	Firewalls, Packet Filtering, Stateful Inspection, HTTPS and SSL/TLS, IPsec and Virtual Private Networks	
	5	Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Log Monitoring Basics, Security Awareness, Safe Network Usage Practices	

5	Teacher Specific Module :		12
	Illustration of Networking Hardware components		

	Configure IP address of a system. Illustration of ipconfig, ping Configuring network host - setting hostname - assigning IP address Setup a Wired LAN with more than two systems Setup a Wireless LAN with more than two systems Secure wifi configuration	
--	---	--

Essential Readings:

1. Tanenbaum, A.S. & Wethrall, D.J.. Computer Networks, 5th edition, Pearson Education, 2012.
2. Forouzan, B. A.. Data Communication and Networking, 4th edition, McGraw-Hill Education, 2017.
3. William Stallings, Data and Computer Communications, VIIth Edition, Pearson Education

Suggested Readings:

- 1) Comer, D. E.. Computer Networks and Internet, 6th edition, Pearson education, 2015.
- 2) Achyut S. Godbole and AtulKahate, Data communication and Networks, 2nd Ed, McGraw Hill

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Case study/Seminar/Viva	10
Total		100

KU5DSCCAP303: WEB TECHNOLOGY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSC	300-399	KU5DSCCAP303	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course provides a comprehensive introduction to web development covering front-end and back-end technologies. It includes HTML5 for structuring web pages, CSS for styling and responsive design, JavaScript for client-side interactivity, and PHP with MySQL for server-side programming and database integration. The course emphasizes practical implementation through hands-on labs, mini-projects, and a final case study to develop full-stack web applications. Students will gain experience in designing dynamic, interactive, and data-driven websites following industry practices.

Course Prerequisite: Database Management System

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Develop structured and semantic web pages using HTML5.	C
2	Apply CSS to design visually appealing and responsive web interfaces.	A
3	Implement client-side interactivity using JavaScript, including DOM manipulation and form validation.	A,C

4	Build dynamic web applications using PHP for server-side processing and MySQL	A,C
5	Develop problem-solving skills through real-world web application projects.	A,C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs							
	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	3	1	1		1	2
CO 2	2	3	1	1		1	2
CO 3	3	3	2	2	1	1	2
CO 4	3	3	3	2	1	2	2
CO 5	3	3	2	3	2	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

MODU LE	U NI T	DESCRIPTION	HOUR S
1	MODULE TITLE : HTML5 Fundamentals		12
	1	HTML5 Structure & Boilerplate	
	2	HTML tags and elements, Attributes	
	3	Semantic & Non-Semantic Tags	
	4	Text Formatting & Lists	
	5	Hyperlinks and navigation, image, audio, video integration	
	6	Tables	

	7	Forms & Input Types, HTML Form Validation	
	8	Iframe	
	9	HTMLBest Practices.	
2	MODULE TITLE : CSS		12
	1	CSS Basics, CSS Selectors, Id/Class/Universal Selectors	
	2	Colors, Units & Typography	
	3	Box Model	
	4	Margin, Padding & Border	
	5	Positioning, Backgrounds, Hover Effects	
	6	Responsive styling Overview : Viewport Meta Tag, Media Queries, Responsive Units, Flexbox Basics, Responsive Images	
3	MODULE TITLE : Introduction to JavaScript		11
	1	Introduction to JavaScript, Role of JavaScript in web development	
	2	Variables & Data Types	
	3	Operators, Conditional Statements, Loops	
	4	Arrays & Objects	
	5	Introduction to DOM Manipulation	
	6	Event Handling Basics: Mouse and keyboard events, Event handlers	
	7	JavaScript Form Validation Basics	
4	MODULE TITLE : PHP and MySQL		10
	1	Introduction to PHP: Uses of PHP	
	2	General Syntactic Characteristics - Primitives, Operations, and Expressions	

	3	Control Statements, Arrays, Functions, Pattern Matching	
	4	Form Handling	
	5	Cookies, Session Tracking	
	6	Basic MySQL Database Connectivity	
5	Teacher Specific Module :		30
	<i>Directions</i>		
	1. Version Control & Collaboration : Git Fundamentals, Repository Initialization, Staging, Commit, Branching, Merging, Resolving Merge Conflicts, GitHub Repository Creation, Push and Pull Operations, README Creation. 2. Create a webpage demonstrating text formatting tags and lists. 3. Design a webpage with hyperlinks, images, audio and video integration. 4. Create a student mark list using HTML tables. 5. Design a student registration form using different input types. 6. Implement HTML form validation using validation attributes. 7. Embed webpages or videos using iframe. 8. Create a personal portfolio webpage with: Header and navigation. About section. Skills list. Image integration, Contact information 9. Develop a simple landing page using semantic HTML tags. 10. Design a registration form using: Text fields, Email and password inputs, Radio buttons and checkboxes, Form validation attributes 11. Basic Webpage Styling using Selectors 12. Design a structured content page using text styling and box model. 13. Create structured layouts using CSS positioning and Flexbox. 14. Enhance UI using backgrounds and interactivity. 15. Design a web page to demonstrate Text alignment and Border colours using internal CSS 16. Create a simple responsive card layout using Flexbox and media queries. 17. Create a webpage and display messages using JavaScript. 18. Write a JavaScript program to check whether a student is eligible for admission based on marks. 19. Write a JavaScript program to store details of multiple products using an array of objects and display all product details using a		

	loop. Also display whether each product is Available or Not Available based on stock quantity. 20. Design a web page with a form to accept student details and calculate the result. The system should validate input fields and display whether the student has Passed or Failed based on marks entered. 21. Create a webpage where a button changes the text of a paragraph dynamically. 22. Create a login form and validate user inputs before submission. 23. Write a PHP program to perform arithmetic operations and display results. 24. Practice arrays and conditional statements in PHP. 25. Write a PHP program to store and display a username using cookies or session. 26. Write a PHP program to connect to a MySQL database and insert student records. 27. Simple Student Registration System - Form input (name, age, marks) - Store data in MySQL - Display stored records - Use sessions for login message 28. Case study : Develop a web application using HTML, CSS, JavaScript, PHP, and MySQL with form validation, data storage, and record display.	
--	---	--

Essential Readings:

1. P. J. Deitel, H.M. Deitel, Internet & World Wide Web How To Program, 4/e, Pearson International Edition 2010.
2. Robert W Sebesta, Programming the World Wide Web, 7/e, Pearson Education Inc., 2014.
3. Bill Kennedy, Chuck Musciano, HTML: The Definitive Guide, 3rd Ed, O'Reilly Media
4. Flanagan David, JavaScript: The Definitive Guide, 6th Ed, O'Reilly Media
5. RasmusLerdorf, Kevin Tatroe, Peter MacIntyre, Programming PHP, 3rd Ed, O'Reilly Media

Suggested Readings:

1. Bob Boiko, Content Management Bible, 2nd Edition, Wiley Publishers.
2. Chris Bates, Web Programming Building Internet Applications, 3/e, Wiley India Edition 2009.
3. Dream Tech, Web Technologies: HTML, JS, PHP, Java, JSP, ASP.NET, XML, AJAX,
4. Jeffrey C Jackson, Web Technologies A Computer Science Perspective, Pearson Education Inc. 2009.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSC	300-399	KU5DSCCAP304	4 (3T+1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces the basic concepts and working principles of Artificial Neural Networks inspired by the human brain. It covers neuron models, activation functions, network architectures, learning rules, and training methods such as backpropagation. The course also provides an overview of different neural network types and their applications in classification, prediction, and pattern recognition. It builds foundational knowledge for further studies in artificial intelligence and machine learning.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand neural network architecture	U
2	To understand different learning rules and training methods.	U
3	To apply ANN models for simple tasks such as pattern recognition and classification.	A
4	To expose students to modern neural-network-based AI ideas such as deep learning and their applications in real-world problems.	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©*

Mapping of Course Outcomes to PSOs							
	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	2	1	2	1	1	2
CO 2	2	2	1	2	2	1	2
CO 3	3	3	2	2	2	1	2
CO 4	3	3	2	3	3	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

MODU LE	U NI T	DESCRIPTION	HOURS
1	MODULE TITLE Introduction		10
	1	Definition and need for Artificial Neural Networks, comparison with traditional computing. Evolution of Artificial Neural Networks	
	2	Biological Inspiration: Structure of biological neurons (Axons, Dendrites, Synapses) vs. the Artificial Neuron model.	
		McCulloch-Pitts neuron model	
	3	Building Blocks: Input layer, hidden layers, output layer, weights, bias, activation function	
	4	Types of activation functions: Step, Linear, Sigmoid, Tanh, and ReLU functions.	
	5	Learning rules: Hebbian learning, Perceptron learning rule	
2	MODULE TITLE Single Layer Networks		10
	1	Single Layer Perceptron: Architecture, training algorithm, and limitations, error, loss function	

	2	Linearly separable and non-separable problems	
	3	Adaline concepts	
	4	Simple pattern classification applications	
3	MODULE TITLE Multi-Layer Networks and Backpropagation		13
	1	Multilayer Feedforward Neural Network	
	2	Backpropagation algorithm, Loss function: MSE, Cross-Entropy	
		Gradient descent and error minimization, delta rule	
	3	Practical Issues: Learning rate, momentum, and overfitting/underfitting.	
	4	Basic regularization ideas.	
	5	Applications in classification and prediction	
4	MODULE TITLE Advanced Neural Networks and Applications		12
	1	Brief idea of associative memories, Hopfield networks, and self-organizing maps (SOM).	
	2	Radial Basis Function (RBF) Networks	
	3	Introduction to Deep Learning concepts: Convolutional Neural Networks and Recurrent Neural Networks (Basic idea)	
	4	Application in NLP, Computer vision, Forecasting and Prediction	
5	Teacher Specific Module : Practice Hands-on: Build simple neural networks using PyTorch/TensorFlow Experimentation with activation functions and optimization methods		30
	1. Introduction to Python for ANN a. Basics of Python for ML (NumPy, Pandas, Matplotlib) b. Loading and exploring datasets c. Write a program to implement AND, OR, and NOT gates using a Single Layer Perceptron.		

	2. Implement and plot activation functions :Step, Sigmoid, ReLU, Tanh 3. Build a simple Multi-Layer Perceptron. Train on small dataset (Iris dataset) Analyse effect of learning rate and epochs a. Run the same MLP with different learning rates and numbers of epochs. b. Observe training vs validation loss and show overfitting / underfitting. c. Regularization: weight decay and early stopping d. Add L2 regularization (weight decay) and enable early stopping in your MLP. e. Compare results with and without regularization. f. Dropout in a neural network g. Implement a simple network with dropout layers. h. Show how dropout reduces overfitting and stabilizes performance. 4. Handwritten Digit Recognition: Use the MNIST dataset to build a network that can identify numbers (0-9) from images. 5. Apply ANN model for classification (e.g., Iris dataset) 6. Complete a case study based on a selected application. Should choose one application. The study should include a brief description of the selected application, its objective, real-world use cases, and a simple explanation of how the system works. Students are required to implement a basic working model using Python.	
--	---	--

Essential Readings:

1. Haykin S. (2009). *Neural Networks and Learning Machines*, Third Edition, PHI Learning.
2. Dive into Deep Learning [Dive into Deep Learning](#), Aston Zhang, Zachary C. Lipton, MuLi, and Alexander J. Smola
3. Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press
4. Artificial Neural Networks, B. Yegnanarayana

5. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
Aurélien Géron

Suggested Readings:

1. Martin T. Hagan, Howard B. Demuth, Mark Beale *Neural Network Design*, PWS Publishing
2. Goodfellow, Yoshua Bengio, Aaron Courville *Deep Learning*, MIT Press

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU5INTERN301:INTERNSHIP

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	Internship		KU5INTERN301	2	120

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
-	-	-	15	35	50	-

All students should undergo Internship of 2-credits during the fourth semester summer break in a firm, industry or organization, or training in labs with faculty and researchers of their own institution or other Higher Educational Institutions (HEIs) or research institutions.

Internship can be for enhancing the employability of the student or for developing the research aptitude. Internship can involve hands-on training on a particular skill/ equipment/ software. It can be a short project on a specific problem or area. Attending seminars or workshops related to an area of learning or skill can be a component of Internship.

The following guidelines are issued for students undertaking their internship as part of the BCA programme. Students are expected to adhere to these guidelines to ensure successful completion and academic credit.

1. Internship Duration and Completion

- a. Students may secure internships through the Internship Kerala Portal, managed by the Kerala State Higher Education Council (KSHEC). Internships offered through Central/State Govt. portals for offering internship/ apprenticeship/ skill development/ training programmes shall also be considered as equivalent to internship courses
- b. Private Sector Requirements: If the Internship Provider Organization (IPO) is from the private sector, the following conditions must be met.
 - i. Experience: Private organizations must have a minimum of 10 years of operational experience in their respective fields.
 - ii. Documentation: To ensure academic standards, colleges are advised to establish a Memorandum of Understanding (MoU) with these organizations, clearly outlining the scope and details of the internship.

- c. Following conditions shall be followed for research-focused internship
 - i. Host Institution: Research internships may be undertaken at any recognized higher education institution (not their parent institute), university department (not their parent department), research laboratory, or industry facility, provided the host possesses the requisite infrastructure for the proposed study.
 - ii. Students are also permitted to utilize Centers of Excellence or laboratories facilitated by State or Central Government agencies, including those within their parent institution.
- d. There should be a minimum 120 hrs. of engagement for the students of BCA in the Internship.
- e. Summer vacations and other holidays can be used for completing the Internship.
- f. In the BCA program, institute/ industry visit or study tour can be part of Internship. Visit to national research institutes, research laboratories and places of scientific importance should be part of the study tour. A brief report of the study tour has to be submitted with photos.
- g. Extensions and Deadline Compliance: While the standard requirement is to complete the internship before the end of the Fifth Semester, the following exceptions apply
 - i. Extensions: In unavoidable circumstances, a student may be permitted to complete the internship before the end of the Sixth Semester, subject to prior formal permission from the University.
 - ii. Mandatory Registration: Regardless of when the internship is physically completed, all students must register for the internship course during the Fifth Semester course registration period. Failure to register during this window may result in a delay in credit awarding.

2. Nature and Continuity of Work

- a. The internship should be relevant to the field of Computer Applications / IT. The institution at which the Internship will be carried out should be prior-approved by the Department Council of the College where the student has enrolled for the UG program.
- b. Students are encouraged to select work that can be extended into their final year project.
- c. The work undertaken should demonstrate practical application of technical skills such as programming, data analysis, web development, cybersecurity, etc.
- d. If deemed necessary, the Internship Supervisor is authorized to visit the internship sites to physically verify the nature of the work and monitor student performance.

3. Weekly Reporting and Guidance

- a. A faculty member in the department shall be assigned as an Internship Supervisor for each student at the beginning of the academic year by the Department for monitoring and supervising the student during the internship.

In exceptional cases, a full time Post Doctoral Fellow in the department can also be the Internship Supervisor. Their duties include:

- i. Mentorship: Guiding students in identifying interest areas and suitable internship placements.
 - ii. Monitoring: Supervise and monitor student progress throughout the internship period.
 - iii. Documentation: Maintaining records of the organization's consent, the Internship Mentor's evaluation report, and the student's final report.
 - iv. Evaluation: Conducting internal assessments and arranging examiners for the Viva Voce examination.
 - v. Coordination: It is recommended that the Supervisor conduct at least one formal meeting (online or offline) with the external Internship Mentor during the program
- b. Each student must report to their assigned faculty guide on a weekly basis to update on the progress of the work. Updates should include:- skills/tools learnt/implemented, tasks completed during the week, challenges faced and learning outcomes. Regular interaction with the guide is mandatory to ensure proper direction and progress

4. Work Documentation (Work Register)

- a. There shall be a designated Internship Mentor at the institute/organisation where the Internship is offered, who should monitor the regularity and performance of the student.
- b. The students should make regular and detailed entries, Daily activities, Hours worked Skills acquired etc., into a personal work book throughout the period of Internship. The workbook must be regularly signed and verified by the internship mentor/ authority at the organization.
- c. The work book will be a record of the progress of the Internship and the time spent on the work, and it will be useful in writing the final report. Refer the appendix for the format of the register
- d. Refer Appendix for Work Register Format and Acceptance Letter

5. On completion of the internship

- a. At the end of the internship, students must have tangible work to demonstrate, such as a software application/module, website or system prototype. data analysis report or dashboard or any technical implementation relevant to their domain
- b. The outcome should clearly reflect the learning and contribution of the student.

6. Final Submission Requirements

- a. Students must submit the following upon completion:
 - i. Internship Report (well-documented and structured)
 1. The report must include the following components:
 - a. Cover Page: Title of the internship, Student's Name, University Register Number, Academic Year, and the names of the College and Department.

- b. Declaration: A signed statement of authenticity by the student.
- c. Official Certificate: The original certificate signed by the Head of the Internship Institution.
- d. Content Body: A detailed account of the work performed, total duration of the engagement, and the specific learning outcomes achieved.
- ii. Internship Workbook register (day-wise) duly signed by the authority
- iii. Completion Certificate (provided by the organization)
- iv. Presentation and Demonstration of the developed work/project

7. Evaluation Criteria

- a. The internship will be evaluated based on:
 - i. Completion of required hours
 - ii. Quality and relevance of work
 - iii. Regular reporting and discipline
 - iv. Documentation and record maintenance
 - v. Report and work/project presentation and demonstration
- b. The credits and marks for the Internship will be awarded only at the end of fifth semester
- c. The evaluation of Internship shall be done internally through continuous assessment mode by a committee internally constituted by the Department Council of the college where the student has enrolled for the UG program.
- d. There shall be no provision for improvement of marks once the internship evaluation is finalized
- e. To be valid for university records, the Internship Report must be formally signed by the Internship Supervisor and the Head of the Department.
- f. The internship shall be evaluated internally by a committee constituted by the Head of the Department (HoD), employing assessment methods appropriate to the nature of the internship.
- g. The Internship supervisor shall provide a performance rating on a scale of A (Excellent), B (Good), C (Average), and D (Poor), which will serve as the basis for awarding marks and credits. Marks and credits shall be assigned based on this grading scale. The official award of credits and marks shall be recorded only upon the conclusion of the Fifth Semester.

8. General Instructions

- a. Students must maintain professional conduct throughout the internship and adhere to the rules and policies of the organization.
- b. Must ensure originality in work, plagiarism or misconduct will lead to disqualification.
- c. Any issues or changes must be immediately reported to the internal supervisor.

The scheme of evaluation of CCA and ESE is given below:

Components of Evaluation of Internship	Weightage	Marks
CCA	30%	15
ESE	70%	35

Evaluation Type		Marks
ESE		35
a)	Presentation	20
b)	Viva	10
c)	Report	05
CCA		15
a)	Punctuality & Work book	05
b)	Acquisition of skill (Presentation/Viva/Practical Test)	10
Total		50

KU6DSCCAP306 : THEORY OF COMPUTATION AND APPLICATIONS IN COMPILER DESIGN

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP306	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2 Hours

Course Description:

This course introduces the fundamental concepts of automata theory, formal languages, and compiler design. Students learn finite automata, regular expressions, context-free grammars, parsing techniques, syntax-directed translation, intermediate code generation, code optimization, and compiler construction tools. The course helps students understand the theoretical foundations of programming languages and the internal working of compilers.

Course Prerequisite: NIL**Course Outcomes:**

CO No.	Expected Outcome	Learning Domains
1	Explain formal languages, regular expressions, DFA and NFA.	U
2	Apply parsing techniques and compiler phases in syntax analysis.	A
3	Construct syntax-directed translations and intermediate code representations.	A,An
4	Analyze type checking, code optimization, and runtime environments	An
5	Design basic compiler components and apply	C

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)***

Mapping of Course Outcomes to PSOs							
	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	2	1	1	1	1	2
CO 2	3	3	1	2	1	1	2
CO 3	3	3	2	2	2	1	2
CO 4	3	3	2	3	2	1	2
CO 5	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOUR S
1	MODULE 1: Formal Languages and Finite Automata		12
	1	Symbols, Alphabets, Strings, Languages	
	2	Regular Expressions and Regular Languages	
	3	Finite Automata – DFA and NFA, Conversion of Regular Expression to NFA, Real world Use Cases of Automata	
	4	Conversion of NFA to DFA	
	5	Applications of DFA : pattern matching and token recognition, Applications of Regular Expressions :text searching, input validation	
	6	Context Free Grammar and Parse Trees, Eliminating Ambiguity	
	7	Applications of CFG: syntax analysis and programming language design, Applications of Parse Trees: expression evaluation and compiler construction	
2	MODULE 2: Compiler Fundamentals and Parsing Techniques		12
	1	Analysis of the source program - Analysis and synthesis phases Introduction to Compiler and Phases of Compilation	
	2	Lexical Analysis - Role of Lexical Analyser, Input Buffering, Specification of Tokens, Recognition of Tokens.	

	3	Role of the Syntax Analyser, Basic parsing approaches - Eliminating left recursion, left factoring. Top-Down Parsing : Predictive Parsing, LL(1) Grammars, Error Recovery	
	4	Bottom-Up Parsing, Handle Pruning and Shift Reduce Parsing	
	5	SLR parsing	
	6	Ambiguous Grammar	
3	MODULE 3: Syntax Directed Translation and Intermediate Code		12
	1	Syntax Directed Definitions (SDD)	
	2	S-Attributed and L-Attributed Grammars	
	3	Annotated Parse Trees	
	4	Intermediate Code Generation , Abstract Syntax Trees and Three Address Code, Quadruples, Triples.	
	5	Translation of Assignment Statements	
4	MODULE 4:Code Optimization, Code Generation		12
	1	Code Optimization - Principal sources of optimization	
	2	Machine dependent and machine independent optimizations	
	3	Basic blocks, Local and global optimizations	
	4	Code generation – issues in the design of a code generator – the target language – a simple target machine model	
5	Teacher Specific Module :		12
	<i>Directions:</i>		
	Design a DFA to validate identifiers and keywords used in programming languages. Convert a given Regular Expression into NFA and then into DFA. Write a simple Lexical Analyzer to identify tokens in a source program. Construct LL(1) parsing table for a given grammar and perform predictive parsing. Generate Three Address Code for arithmetic expressions and assignment statements. Implement simple code optimization techniques such as constant folding and dead code elimination.		

Essential Readings:

1. Michael Sipser, Introduction to the Theory of Computation.
2. Alfred V. Aho, Ravi Sethi, Jeffrey Ullman, Compilers: Principles, Techniques and Tools.
3. John E. Hopcroft and Jeffrey Ullman, Introduction to Automata Theory, Languages and Computation.

Suggested Readings:

1. Kenneth C. Loudon, Compiler Construction: Principles and Practice.
2. Andrew W. Appel, Modern Compiler Implementation.
3. D. Grune et al., Modern Compiler Design.
4. Kamala Krithivasan and Rama R., Introduction to Formal Languages and Automata Theory.

Assessment Rubrics:

End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation (Theory)		30
a)	Test Paper-1	15
b)	Assignment	10
c)	Viva/seminar/Case study	5
Total		100

- **Employability for the Course:** Language Processing Engineer. Software Developer, System Programmer, Research Assistant in Computing Theory

KU6DSCCAP307: PRINCIPLES OF OPERATING SYSTEMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP307	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2 Hours

Course Description: This course introduces the basic concepts and functions of operating systems. It covers topics such as process management, memory management, CPU scheduling, file systems, device management, and system security. Students will learn how operating systems manage computer hardware and software resources and provide an interface between users and the computer system. The course helps students build a foundation for system programming, software development, and computer system administration.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic concepts, structure and functions of operating systems.	U
2	Understand the principles behind the techniques in resource management	U
3	Knowledge about the basic design of the OS	U

4	Understand the concepts of OS in Linux OS	An, A
---	---	-------

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	2	1	1	2
CO2	3	2	3	2	1	1	2
CO3	3	3	2	2	2	1	2
CO4	3	2	3	2	1	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

MODU LE	U NI T	DESCRIPTION	HOURS
1	OPERATING SYSTEMS OVERVIEW		12
	1	Operating System Structure, Operating system operations, Process Management, Memory Management, Storage Management, Protection and Security	
	2	Operating System Services, User and Operating-System Interface, System Calls, Types of System Calls , Operating-System Design and Implementation	
	3	Introduction to Linux OS, Features, Linux History, Design Principles	
2	PROCESS MANAGEMENT		12
	1	Process Concept , Process States , Process control block , Process Scheduling, Operations on Processes , Interprocess Communication, Communication in Client Server Systems	

	2	CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms	
	3	Process synchronization- Race conditions – Critical section problem, Synchronization problem - Producer Consumer, Semaphore (Concept)	
	4	Deadlocks: Deadlock Characterization, Necessary conditions, Resource allocation graphs, Methods for Handling Deadlocks	
	5	Deadlock Prevention, Deadlock Avoidance- Banker's algorithms , Deadlock Detection, Recovery from Deadlock	
3	MEMORY MANAGEMENT		12
	1	Main Memory : Background , Swapping , Contiguous Memory Allocation	
	2	Main Memory : Segmentation, Paging , Structure of the Page Table	
	3	VirtualMemory : Background, Demand Paging , Page Replacement, Thrashing	
	4	Introduction to Shell Programming: Types of Linux Shells, Basic Commands: ls, cd, pwd, mkdir, rm, cp, mv, chmod, touch Input/Output Commands: read, echo, Text Processing Commands: grep, cat Arithmetic, Logical and Relational Operator Iterative & Conditional Commands: if, while, for, break, continue, case Command line arguments, Shell variables	
4	STORAGE MANAGEMENT and I/O SYSTEMS		12
	1	Mass Storage Structure : Overview, Disk Scheduling , Disk Management.	
	2	File System Storage: File Concept, Directory and Disk Structure, File Sharing, Protection.	
	3	File System Implementation: File System & Directory Structure & Implementation, Allocation Methods, Free Space Management.	

	4	I/O Systems : Overview, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O Requests to Hardware Operations	
5	Teacher Specific Module : Implement the given exercise and instruct students to submit a report.		12
	Case Study - Linux Systems: Process Management, Scheduling, Memory Management, File Systems 1. Write a shell script to get the filename from the user and for a menu driven program to perform file management (create a file, display content of a file, delete the file, write content to a file). 2. Write a shell script to copy the contents of the file to another. Input file names through command line. The copy should not be allowed if the second file exists. 3. Write a shell script to display current date and time, list all user account names, count of logged in user accounts, list all logged in user accounts with login time. 4. Write a shell script to display your system details (number of users, current processes, memory usage, system running time) 5. Write a shell script to implement and examine the effectiveness of the First Come First Serve CPU Scheduling algorithm. Find the average waiting time and turnaround time.		

Essential Readings:

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, 9th Edition, John Wiley and Sons Inc.,2012
2. <https://www.cse.iitb.ac.in/~mythili/os/>
3. Operating Systems, D M Dhamdhere, 2nd Edition, Tata McGraw Hill

Suggested Readings:

1. System Programming , D M Dhamdhere, Tata McGraw Hill
2. [Operating System Tutorial](#), accessed on 20/4/2026
3. <https://nptel.ac.in/courses/106106144>, accessed on 20/4/2026
4. <https://homecse.iitd.ac.in/>, accessed on 20/4/2026

Assessment Rubrics:

End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation (Theory)		30
a)	Test Paper-1	15
b)	Assignment	5
c)	Shell Program Report	5
d)	Viva/seminar/Case study	5
Total		100

KU6DSCCAP308: DESIGN AND ANALYSIS OF ALGORITHMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP308	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2 Hours

Course Description:

To introduce the fundamental concepts of algorithm design and computational complexity analysis. To develop problem-solving skills using efficient sorting, searching, divide and conquer, and greedy techniques. To enable students to analyze and implement graph algorithms and dynamic programming approaches for real-world computational problems. To familiarize students with advanced algorithmic strategies such as backtracking and branch and bound methods. To provide an understanding of computational complexity classes including P, NP, and NP-Complete problems. To enhance logical thinking, analytical ability, and algorithmic efficiency through practical problem-solving exercises and performance evaluation techniques.

Course Prerequisite: Data Structures and Algorithms**Course Outcomes:**

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of algorithms, asymptotic	U

	notations, and complexity analysis.	
2	Apply suitable sorting, searching, and greedy algorithms to solve computational problems.	A
3	Analyze divide and conquer, graph traversal, and dynamic programming techniques for efficient problem solving.	AN
4	Develop algorithmic solutions using backtracking and branch and bound methods.	A,C
5	Evaluate computational complexity and classify problems based on P, NP, and NP-Completeness concepts.	E

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©*

Mapping of Course Outcomes to PSOs							
	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	2	1				
CO 2	3	3	2	1			
CO 3	3	3	3	2	1		
CO 4	3	2	3	3	2	1	
CO 5	2	2	3	2	2	1	1

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOUR S
1	MODULE 1: Introduction to Algorithms and Complexity		12
	1	Fundamentals of algorithms, Time and space complexity	
	2	Best, average and worst case analysis, Asymptotic notations	
	3	Recursion and recurrence relations, probabilistic analysis	
	4	Frequency Count Method, Substitution Method, Recursion Tree	

		Method, Master Theorem	
2	MODULE 2: Algorithm Design Techniques		12
	1	DIVIDE AND CONQUER (Algorithm and Analysis): General method, applications-analysis of binary search, quick sort, merge sort, AND OR Graphs	
	2	GREEDY METHOD (Algorithm and Analysis): General method, Applications-job sequencing with deadlines, Fractional knapsack problem, minimum cost spanning trees, Single source shortest path problem	
	3	Comparison of sorting techniques	
3	MODULE 3: Graph Algorithms and Dynamic Programming		12
	1	GRAPHS (Algorithm and Analysis): Breadth first search and traversal, Depth first search and traversal, Spanning trees, connected components and bi-connected components	
	2	DYNAMIC PROGRAMMING (Algorithm and Analysis): General method, applications—Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design	
4	MODULE 4: Back Tracking and Branch and Bound		12
	1	BACKTRACKING (Algorithm and Analysis): General method, Applications- n-queen problem, Graph coloring and Hamiltonian cycles, 0/1 knapsack problem	
	2	Branch and bound solution, FIFO branch and bound solution	
	3	P, NP and NP-Complete problems	
5	Teacher Specific Module :		12
	<i>Directions</i> 1. Introduction to randomized algorithms 2. Analysis of Hashing techniques, Priority queues 3. Visual representation of sorting/searching 1. Step-by-step execution tracing 2. Greedy and dynamic programming examples 3. Las Vegas and Monte Carlo algorithms 4. Vertex cover approximation 5. Parallel sorting concepts		

	6. Algorithms in social media and search engines	
	7. Performance comparison studies of different algorithms for the same problem.	

Essential Readings:

1. Introduction to Algorithms – Thomas H. Cormen et al.
2. Ellis Horowitz, Satraj Sahni, Rajasekharam (2007), Fundamentals of Computer Algorithms, 2nd edition, University Press, New Delhi
3. Fundamentals of Computer Algorithms – Ellis Horowitz and Sartaj Sahni
4. Algorithm Design – Jon Kleinberg and Éva Tardos
5. Data Structures and Algorithm Analysis in C – Mark Allen Weiss

Assessment Rubrics:

End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation (Theory)		30
a)	Test Paper-1	15
b)	Assignment	10
c)	Viva/seminar/Case study	5
Total		100

Employability for the Course:

The course equips students with modern frontend development skills required for careers such as React Developer, Frontend Web Developer, UI Developer, Full Stack Developer, and JavaScript Application Developer in the software industry.

KU6DSCCAP309: APPLIED COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP309	4(0+4)	120

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
0	8	0	40	60	100	2

Course Description: Applied Computing focuses on the practical application of computing concepts, tools, and technologies to solve real-world problems. The course provides students with opportunities to apply programming, computational thinking, system design, data analysis, and software development techniques in interdisciplinary and industry-oriented contexts. It encourages analytical thinking, innovation, experimentation, and research-oriented problem solving through practical assignments, case studies, computational investigations, and project-based activities.

Course Prerequisite: Knowledge about Discipline Specific Courses

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Application of theoretical knowledge and practical skills acquired throughout the program to solve real-world problems.	A,C
2	Development of critical thinking and problem-solving abilities through tackling challenges	A,C
3	Improved technical skills in software tools, programming languages, and techniques	A,C

4	Work independently or collaboratively in planning, execution, and management of computing projects.	A,C
5	Ability to contribute to the domain of computer science through research, innovation, and practical implementation.	A,C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	3	2	2	2	2	2
CO 2	3	2	1	3	2	2	2
CO 3	2	3	3	2	1	1	3
CO 4	2	3	2	2	2	3	3
CO 5	3	3	2	3	3	2	3

GUIDELINES

Applied Computing is a **mandatory** Discipline Specific Course offered in the VI Semester of the BCA programme. The course is designed to cultivate students' research, analytical, computational, and software development skills through practical and application-oriented work.

It serves as a capstone experience that enables students to integrate and apply the theoretical knowledge acquired throughout the programme to real-world problems and computing challenges. Individual projects/case study are encouraged; however, students may work as a group with a maximum of three members per group.

The duration of the project extends over one academic year, commencing in the fifth semester and concluding with submission and evaluation at the end of the sixth semester.

The sessions shall focus on planning, guidance, design methodology, review, and evaluation activities. Also shall be utilized for implementation, experimentation, software development, testing, and related practical work.

Proposal

Students shall submit a proposal outlining the problem statement, objectives, scope of work, methodology, implementation strategy, and expected outcomes. The proposal should demonstrate relevance to contemporary trends, emerging technologies, or real-world challenges in the field of computing.

Literature Review

Students shall conduct a comprehensive literature review to establish the theoretical and technical foundation. The review should identify existing approaches, research gaps, limitations, and opportunities for innovation or improvement.

Execution Plan

Students shall prepare a systematic execution plan including project phases, milestones, timelines, tools and technologies to be used, resource requirements, and evaluation strategies.

Version Control

Students shall use version control and collaborative software development practices using tools such as Git and GitHub for managing projects, and team-based development activities throughout the course.

Implementation and Analysis

The project may involve experimental, theoretical, computational, or application-oriented work such as software development, algorithm implementation, simulations, analytical studies, or research investigations. Appropriate testing, validation, experimentation, and analysis shall be carried out wherever applicable.

Presentation

Students shall present the progress and outcomes of the project through seminars, demonstrations, and viva voce examinations in a structured and professional manner.

Report

Students shall prepare a comprehensive report including introduction, objectives, literature review/ground Study, methodology, system design or experimental setup, implementation

details, results, analysis, testing, screen shots, sample code (if permitted), conclusions, limitations, and future scope or recommendations, References.

FORMAT

Font- Times New Roman

PROJECT REPORT (12 Regular- all capitals)

TITLE (16 Bold all capitals – Times New Roman)

submitted by (12 Regular)

NAME: (14 Bold- all capitals)

Reg. No (12 Bold)

Department Name (14 Bold, leading capitals)

Name of college (12 Regular)

Place

Month, Year (14 Regular)

Chapter Titles- 16 Bold – All capitals – No underlining

-Main headings - 14 Bold - All capitals – No underlining

Main headings shall be numbered as 1.1,1.2,1.3,..... 2.1,2.2,2.3..... etc.

-Sub headings- 12 Bold- Leading capitals- No underlining

Sub headings shall be numbered as 1.1.1,1.1.2,.....2.1.1,2.1.2.....etc

Figures, sketches, equations and tables shall be serially numbered chapter wise (Eg. 2.1,2.2..... 3.1,3.2.....).

-All figures, sketches, photos and tables shall be titled.

-Figure no. and title (12 Regular) shall be given below the figure.

-Table no and title (12 Regular) shall be given above the table

Text (Content under each heading) – 12 Regular, Times New Roman, Alignment- Justified.

-Figures, Tables, Sketches and Equations shall be centre justified. Figures, sketches, tables shall be placed immediately after the paragraph in which they are referred.

Begin each chapter in a new page.

-Begin paragraph in the line next to the heading.

Start paragraphs indenting 10 character space in the beginning

Do not begin a new section at the end of a page. Minimum two lines must follow a main

heading/sub heading in a page

All references listed must be cited in the text, all cited references shall be listed in the REFERENCES.

Page numbering : numerals (12 Regular font) – bottom centred

DECLARATION (use double or 1.5 line spacing with 12 Regular font)

CERTIFICATE PAGE (use double or 1.5 line spacing with 12 Regular font)

ACKNOWLEDGEMENT (use double or 1.5 line spacing with 12 Regular font)

ABSTRACT (Do not exceed 2 pages – use double or 1.5 line spacing with 12 Regular font) (No figures, sketches, tables shall be there)

LIST OF TABLES (No. , Title and Page No shall be given)

LIST OF FIGURES (No. , Title and Page No shall be given)

ABBREVIATIONS: List all abbreviations and their expansion in Alphabetical order

Publication and Dissemination (Not mandatory)

Students are encouraged to publish their work in journals indexed in UGC-CARE or present their findings at national/international conferences conducted by recognized professional bodies or reputable publishers.

EVALUATION

Evaluation of the course shall be done at two stages:

1. **Internal Assessment** : The internal evaluation of the course shall be done through continuous assessment mode by the course coordinator throughout the semester. 30% of the weightage shall be given through this mode.

Project Proposal and Planning

- Clarity of goals and objectives.
- Scope of the System
- Feasibility of the chosen approach.
- System study/literature review and proposed methodology.
- Clarity of project schedule and division of tasks within the team.

Project Progress and Implementation

- Regular reviews and adoption of feedback provided by the supervisor.
- Completion of milestones as planned.

Presentations

- Effectiveness of communication and presentation skills.
- Clarity of technical details and progress made.
- Ability to answer questions about the work effectively.
- Interim and Final Presentations

2. **External evaluation:** External evaluation shall be Conducted by an internal examiner appointed by the Department Council and the course coordinator. This will take place at the end of the VI th semester which may include the project report, demonstration or presentation, and viva voce examination. 70% of the weightage shall be given.

Assessment Rubrics:

Evaluation Type		Marks
ESE		60
a)	Presentation (Demonstration of the project or present the research findings)	15
b)	Viva voce based on presentation	15
c)	Novelty, Relevance & Contribution	10
d)	Version control using Git and GitHub	10
e)	Report	10
CCA		40
a)	Proposal and Planning	5
b)	Punctuality & Work book	5
c)	Presentation & Viva-Voce (Minimum 3)	30
Total		100

Employability for the Course: Applied Computing enhances employability by developing practical programming, problem-solving, software development, and computational skills required for industry-oriented IT and technology careers.

KU6DSCCAP310 : INTRODUCTION TO DEEP LEARNING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP310	4 (3T+1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces fundamental and advanced concepts of neural networks and deep learning. It covers biological inspiration, perceptron models, activation functions, multilayer perceptrons, training methods, and optimization techniques. Students gain hands-on experience in building, training, and evaluating deep learning models using frameworks such as Keras and TensorFlow.

Course Prerequisite: Artificial Neural Network

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	To introduce the fundamentals of Deep Learning and its role in image processing tasks	U
2	Apply Regularization methods Early stopping, data augmentation, dropout etc. for optimization.	A
3	To develop understanding of Convolutional Neural Networks for feature extraction and pattern recognition	U

4	To understand the fundamentals of sequence models using RNNs	U
5	To develop problem-solving skills in real-world computer vision applications	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	2	1	2	1	1	2
CO 2	3	3	2	2	2	1	2
CO 3	3	3	2	2	2	1	2
CO 4	2	2	2	2	2	1	2
CO 5	3	3	2	3	3	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Review of Neural Networks		10
	1	Review of Neural Networks: Model of a biological neuron, McCulloch Pitts Neuron,	
	2	Activation Functions : Sigmoid, Tanh, ReLU, Softmax	
	3	Single Layer Perceptron, Multi-Layer Perceptron (MLP)	
	5	Loss Functions, Training of MLPs, Gradient Descent, Backpropagation algorithm	
	6	Practical Issues in Neural Network Training, Overfitting, Vanishing and Exploding Gradient Problem, Difficulties in Convergence, Local and Spurious Optima, Computational Challenges	
2	MODULE TITLE Deep Learning		11

	1	Deep Feedforward Networks, History of Deep Learning and Differences from Traditional Machine Learning.	
	2	Training Deep Models : Gradient Descent, Stochastic Gradient Descent	
	3	Optimization Techniques : Momentum, Adam	
	4	Regularization Techniques : L1 and L2 Regularization, Dropout, Early Stopping	
	5	Parameter Initialization (concept: Xavier, He)	
3	MODULE TITLE : Convolutional Neural Network		12
	1	Introduction to CNNs	
	2	Convolution operation, filters, feature maps	
		Pooling : max pooling, average pooling	
	3	Fully connected layer	
	4	CNN architecture and layers	
	5	Architecture of Lenet, Alexnet (Overview only)	
4	MODULE TITLE : Sequential Modeling, Application		12
	1	Sequence modeling using RNNs (Basic concept)	
	2	Vanishing gradient problem	
	3	LSTM, GRU (Introduce the purpose)	
	4	Applications of CNNs in image classification, object detection, segmentation, healthcare, and autonomous systems.	
5	Teacher Specific Module :		30
	1. Implement basic functions in Keras and Tensorflow. 2. Perform preprocessing of image data for CNN including resizing, normalization, handling corrupted or inconsistent images, label encoding, and applying data augmentation techniques. 3. Build a deep feedforward neural network to classify data using a standard dataset (example: Iris dataset or MNIST flattened). 4. Design and train a simple neural network and systematically study how different choices affect training and performance. Analyse the impact of different initialization, optimization, dropout and regularization strategies and visualise the change in performance.		5

	5. Digit classification using CNN for MNIST dataset using LeNet-5. Analyse the model structure and understand the output. a. Environment Setup and Data Loading b. Data Preprocessing c. Defining the LeNet-5 Architecture d. Training e. Evaluation and Visualization 6. Implement simple RNN using Keras, Dataset: sequence prediction (toy dataset) 7. Simple RNN for IMDB Review Classification 8. Case study: Select a simple, real-world application and explain its concept, working, and use cases.	
--	---	--

Essential Readings:

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press 2016
2. Francois Chollet. *Deep Learning with Python*. Manning Publications, 2021.
3. Dive into Deep Learning [Dive into Deep Learning](#), Aston Zhang, Zachary C. Lipton, MuLi, and Alexander J. Smola

Suggested Readings:

1. Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.
2. Aurelien Geron, .Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Third Edition, October 2022

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU6DSCCAP311: INTRODUCTION TO COMPILER DESIGN

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSC	300-399	KU6DSCCAP311	4 (4T+0P)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2 Hours

Course Description:

This course covers the fundamental concepts of different phases of compilation such as lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization and code generation. Students can apply this knowledge in design and development of compilers.

Course Prerequisite: Foundations of Computers and Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Identify different phases in the compilation process.	R
2	Understand in detail how lexical, syntax and semantic analysis works.	U
3	Learn the need of intermediate code generation and its various forms.	An

4	Model a Lexical Analyser and Syntax Analyser.	A
5	Illustrate code optimization and code generation techniques in compilation	U

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	2	1	1	1	1	2
CO 2	2	3	1	2	1	1	2
CO 3	3	3	2	2	2	1	2
CO 4	3	3	2	2	2	1	2
CO 5	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE: Introduction, phases of compiler and lexical analysis.		12
	1	Introduction to compiler: definition of compiler, Classification of Compiler: Single pass, Multi pass, Load and Go, Parts of Compilation: Analysis and Synthesis.	
	2	The phases of a compiler: Lexical Analyser, Syntax Analyser, Semantic Analyser, Intermediate code generator, Code optimizer, Target Program, Symbol table manager.	
	3	Compiler construction tools.	
	4	Lexical analysis: Programming language basics - lexical analysis – role of lexical analyzer – input buffering.	
	5	Specification of tokens: strings, languages, regular expressions-recognition of tokens using finite automata.	
	6	Activity: Create a simple lexical analyser that recognizes numbers and arithmetic + and minus (-) operators for addition and subtraction using	

		Lex.	
--	--	------	--

2	MODULE: Syntax Analysis		12
	1	Syntax analysis – role of parser – error handling and recovery – definitions of parsing,	
	2	Context free grammars – derivations-leftmost and rightmost derivation, reductions, Eliminating Ambiguity	
	3	Parsing- parse tree – types of parsing-Top-down parsing and bottom-up parsing,	
	4	Basic parsing approaches - Eliminating left recursion, left factoring. Top-Down Parsing : Predictive Parsing, LL(1) Grammars.	
	5	Bottom-Up Parsing, Handle Pruning and Shift Reduce Parsing	
	6	SLR parsing	
	7	Activity: Create a simple Syntax analyser for verifying addition and subtraction expressions, using YACC. Use it with a lexical analyser to print the result.	

3	MODULE:(Syntax directed translation and Intermediate code generation		12
	1	Syntax Directed Definitions (SDD)	
	2	S-Attributed and L-Attributed Grammars	
	3	Annotated Parse Trees	
	4	Bottom-up evaluation of S-attributed definitions.	
	5	Intermediate code generation – Directed Acyclic Graphs (DAG)	
	6	Three address code – addresses and instructions – quadruples – triples – Static Simple Assignment form	

4	MODULE: Code Optimization and Generation		12
	1	Run time environments – storage optimization – static Vs dynamic allocation	
	2	Stack allocation of space - activation trees and records – calling sequences.	
	3	Code optimization - the principal sources of optimization, Machine	

		dependent and machine independent optimizations, Basic blocks, Local and global optimizations	
	4	Code generation – issues in the design of a code generator – the target language – a simple target machine model.	

	Teacher Specific Module :	12
	<i>Input specifications to Lex and steps for compilation and usage.</i> <i>Input specifications to Yacc and steps for compilation and usage.</i> <i>Concept of JIT compilation</i> <i>Overview of modern compiler frameworks</i> <i>Modular compiler architecture</i> <i>Compilers for embedded systems</i> <i>Recent trends in compiler construction.</i>	

Essential Readings:

1. V Aho A, Ravi Sethi, D Ullman J, Compilers Principles, Techniques and Tools, 2nd Edition, Pearson Education Singapore Pte Ltd, ISBN: 8131721019.
2. Principles of Compiler Design by MG Durga and TG Manikumar. ISBN: 978-81- 8094-161-0

Suggested Readings:

1. W Appel and Andrew, Modern Compiler Implementation in C, 1st Edition, Cambridge University Press, ISBN: 817596071X.
2. Allen I Holub, Compiler Design in C, 1st Edition, PHI Learning Pvt Ltd, ISBN: 812030778X.

Assessment Rubrics:

End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation (Theory)		30
a)	Test Paper-1	15

b)	Assignment	10
c)	Viva/seminar/Case study	5
Total		100

DISCIPLINE SPECIFIC ELECTIVE COURSES With SPECIALIZATION

Data Science

KU5DSECAP301: STATISTICAL METHODS FOR DATA SCIENCE USING PYTHON

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP301	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: This course provides a foundational understanding of statistics, probability, and data science concepts using the Python programming language. It introduces students to data collection, organization, and analysis techniques, along with the basics of probability theory and statistical thinking. The course also covers inferential statistics, hypothesis testing, correlation, and regression analysis for drawing meaningful conclusions from data. Students will learn to create effective data visualizations and interpret analytical results for real-world decision-making applications.

Course Prerequisite: 1. Basics of Mathematics, 2. Fundamentals of Programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand fundamental concepts of statistics, probability, descriptive statistics, statistical inference, sampling methods, probability distributions, and hypothesis testing used in real-world data analysis.	U
2	Apply statistical techniques using Python libraries such as NumPy, Pandas, SciPy, Matplotlib, and Seaborn for real-world data analysis problems.	A

3	Use Python libraries such as NumPy, Pandas, SciPy, Matplotlib, Seaborn, and Statsmodels for data analysis, statistical computation, simulation, visualization, and interpretation of real-world datasets.	C
4	Visualize and summarize data effectively using graphical techniques such as histograms, box plots, scatter plots,	C
5	Develop analytical thinking, statistical reasoning, and problem-solving skills required for advanced studies and applications in data science, machine learning, artificial intelligence, business analytics, and research.	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	2	2	1	2
CO2	3	3	3	2	2	1	3
CO3	3	3	3	3	3	1	3
CO4	3	2	2	3	2	1	2
CO5	3	2	2	3	3	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
		<i>All topics in this course should be taught using real-world computing and data-driven examples to help students understand how statistical concepts are used for data analysis rather than purely abstract formulations</i>	
1	MODULE TITLE-Introduction to Statistics & Python		11
	1	Introduction to Statistics -Definition, Types of Statistics -Descriptive , Inferential, Applications of statistics,Methods of collection, Population, Sample, Types of data -Qualitative(Categorical), Quantitative(Numerical), Types of Quantitative data- Discrete, Continuous.	
	2	Scales of Measurement -Nominal Scale , Ordinal Scale ,Interval Scale ,Ratio Scale , Concept of primary and secondary data, Population and sample , Parameters and statistics, Importance of sampling, Sampling methods- Random and Nonrandom Sampling , Types of sampling,	

		Sample size, Sample Bias, Data collection methods. Real world applications and solutions.	
	3	Descriptive Statistics- Measures of Central Tendency-Mean, Median, Mode, Measures of Dispersion-Range, Variance, Standard Deviation, Coefficient of variation, Quartiles and Percentiles, Skewness, Kurtosis, Correlation, Covariance, Z-Score, Real world applications and solutions.	
	4*	Introduction to Python -Introduction to Python for Data Science, Installing Python and Jupyter Notebook, Introduction to Google Colab Python Variables and Data Types, Collections in Python-Lists, Tuples, Sets, Dictionaries(Basic methods only), Modules and Packages.	

	MODULE TITLE-Basics of Probability and Data Science		12
2	1	Basics of Probability- Probability, Properties, Types, Sample Space and Events, Conditional Probability, Bayes' Theorem, Random Variables, Discrete and Continuous Variables, Expected Value and Variance, Probability Rules, Probability Mass Function (PMF), Probability Density Function (PDF), Cumulative Distribution Function (CDF), Real world applications and solutions.	
	2	Probability Distributions- Discrete Distributions-Bernoulli, Binomial, Poisson, Continuous Distributions- Normal (Gaussian), t-distribution, Exponential Distributions, Real world applications and solutions.	
	3*	Python-Random Numbers, Generating Random Numbers Using the random package-random() randint(), choice(), sample(), shuffle(), random.uniform(),	
	4*	Python Numpy Module- ndarray, Creating arrays, basic operations, statistical methods for descriptive statistics and probability. Applications and real world problems.	

	MODULE TITLE-Inferential statistics, Hypothesis Testing		12
3	1	Statistical Inference: Estimation Theory- Statistic, Parameter, Estimator, Point Estimation, Standard error, Maximum Likelihood Estimation (MLE), Method of moments, Confidence intervals for population Parameters, Central Limit Theorem, Law of Large Numbers. Real world applications and solutions.	
	2	Testing of Hypothesis: Concept of a hypothesis testing General principles of testing, Two types of errors . Types of Testing: Z- test, T-Test, ANOVA-Test, Chi-square test, Real world applications and solutions.	

	3	Introduction to Pandas-DataFrames and Series, Basic Data Inspection, Creating datasets, Loading CSV files, Data type identification, Sample vs population examples. Handling Missing Values, Statistical Functions, GroupBy Operations, Data Aggregation, Sorting and Filtering Data	
	4*	Python Packages and methods - scipy.stats, stats.sem() , stats.t.interval() , poisson, statsmodels.stats.weightstats.ztest(), z_stat, p_value, ttest_rel(), f_oneway() , chi2_contingency()	

4	MODULE TITLE-Correlation, Regression Analysis and and Data Visualisation		10
	1	Correlation: Definition, scatter diagram, types of correlation, measures—Karl Pearson’s correlation coefficient and Spearman’s rank correlation coefficient. Difference Between Karl Pearson Correlation and Spearman Rank Correlation , Real world applications and solutions.	
	2	Regression: Linear regression-fitting by least square method and interpretation, Multiple Linear regression, Prediction using Regression, Real world applications and solutions.	
	3*	Python Packages (Only necessary ones)-scipy.stats , sklearn.linear_model , sklearn.metrics , Methods- pearsonr() , spearmanr() , LinearRegression() , fit(), predict(), LogisticRegression(), mean_absolute_error(), mean_squared_error(), r2_score()	
	4*	Data Visualization-matplotlib module, pyplot, formatting a plot , different types of plots -Histograms, Box Plots, Scatter Plots, Bar Charts, Line Graphs, Heatmaps.	

* (This section is for lab only and not for theory examination)

5	Teacher Specific Module :		30
	<i>Directions</i>		
	1. A. A weather simulation system needs random humidity values between 0 and 1. Generate 5 random floating-point values. B. Create a virtual dice simulator for a board game application. Generate 10 random dice values. C. An online learning platform randomly selects questions for each student. Choose one random question from a list. D. Generate 6 unique lottery numbers between 1 and 50.		

	E. A teacher wants a random presentation order for students. Shuffle roll numbers randomly. F. G. A climate monitoring system generates random temperatures between 20°C and 40°C. Generate a list of 10 temperatures. G. Generate random password characters for the security system. 2. A. Write a Python program to calculate the arithmetic mean of student marks using NumPy. Consider the marks scored by students as: [78, 85, 90, 67, 88]. Interpret the result obtained. B. Write a Python program to calculate the median waiting time of patients using NumPy. One patient experienced an unusually long waiting time. Consider the waiting times: [10, 12, 15, 18, 100]. Explain why median is preferred in this situation. C. Write a Python program to determine the mode of shirt sizes using SciPy. Use the following data: ["M", "L", "M", "S", "M", "L"]. Interpret the result. D. Write a Python program to calculate the range of temperature readings using NumPy. Use: [25, 28, 32, 35, 40]. Interpret the result. E. Write a Python program to calculate variance of machine-produced product weights using NumPy. Consider: [50, 52, 49, 51, 48]. Interpret the result. F. Write a Python program to calculate standard deviation of stock prices using NumPy. Use: [100, 102, 98, 105, 95]. Explain the significance of standard deviation. 3. A. Write a Python program to calculate coefficient of variation for investment returns using NumPy. Consider: [10,12,11,9,13]. Interpret the result. B. Write a Python program to calculate quartiles and 90th percentiles using NumPy. Use: [50,60,70,80,90]. Interpret the quartiles obtained. C. Write a Python program to calculate skewness of income distribution using SciPy. Use:	
--	---	--

	[20000,22000,25000,30000,100000]. Interpret the skewness value. D. Write a Python program to calculate kurtosis of stock returns using SciPy. Use: [2,3,4,5,100]. Interpret the result. E. Write a Python program to calculate z-scores of student marks using SciPy. Use: [50,55,60,65,95]. Identify whether any outlier exists. F. A bank wants to identify the top 25% high-value customers based on account balance as [1000, 2000, 3000, 4000, 5000] . Find 75th percentile. 4. A. Generate heights of students using Normal Distribution. B. factory produces products with 10% defect probability. Simulate defective products for 10 production batches.(Hint: Use Binomial distribution) C. A city traffic department observes average 4 accidents daily. Simulate accident counts for 15 days.(Use poisson distribution). 5. A. Generate machine temperature readings every second. Print Time and Temperature sequence. B. Analyze watch times of users on streaming platforms. Print mean and standard deviation. 6. A. A school wants to visualize marks scored by a student across 5 exams. Create a line graph to analyze performance trends, consistency, and improvement patterns. B. A retail company wants to compare monthly sales. Create a bar chart for sales analysis and describe your findings. C. A shopping mall wants to analyze customer age distribution. Create a histogram for customer ages and describe your findings. D. A social media company wants to study the relationship between number of posts and likes received. posts = [1,2,3,4,5] , likes = [100,200,300,400,500] Create scatter plot and describe your findings. E. An HR department wants to analyze employee salary spread and identify outliers given salary = [25000,30000,32000,35000,40000,100000] .Create a box plot and describe your findings. F. A university wants to analyze relationships among subject marks given the following data { "Math": [80,85,90], "Physics": [78,88,92],	
--	--	--

	"Chemistry": [82,86,91] }. Create a heatmap and describe your findings. 7. A. A university collected student probability exam scores. data = {"Name": ["A", "B", "C", "D"], "Marks": [78, 85, 90, 67], "Probability_Grade": ["B", "A", "A", "C"]} .Perform dataset exploration to: - inspect data, - identify data types, - and summarize statistics. B. A probability experiment records coin toss results as data = {"Result": ["Head", "Tail", "Head", "Head", "Tail", "Tail", "Head", "Head"]}. Analyze frequency of Heads and tails and give your interpretation C. A healthcare company wants random patient samples for probability Analysis. Given the data = { "Patient": [1,2,3,4,5], "Age": [22,25, 30, 35, 40]}, select random rows. D. Analyze average marks for each grade category given the following data = { "Grade": ["A","A","B","B","C"], "Marks": [90,85,75,70,60] } E. A delivery company records: success = 1, failure = 0. Given the data = { "Delivery": [1,1,0,1,1,0,1] } , Analyze probability of successful Deliveries.Justify your answer.(Hint: Use Bernoulli distribution). F. Generate normally distributed student marks and analyze statistics. 8. A. A climate research team wants to analyze the relationship between: Temperature = [30, 32, 35, 38] and electricity consumption = [200, 220, 260, 300] . Find correlation coefficient. B. A researcher studies the relationship between study hours and probability exam marks. data = { "Hours": [1,2,3,4,5], "Marks": [40,50,60,70,80] }. Find correlation matrix. 9. A. A food delivery company wants to estimate average delivery time in a city. Delivery times (minutes): [25, 28, 30, 35, 27].Find:sample mean,standard deviation and standard error.Justify the significance of standard error. B. A logistics company wants to estimate average delivery time for parcels given delivery time in days = [3,4,5,2,6,4,5,3,4,5] Sample Mean, Standard Deviation, Standard Error, and 95% Confidence Interval. Describe your findings. (Description example : Mean estimates average delivery time,Standard error measures reliability of sample mean, Confidence interval estimates population mean range.Small SE means more reliable estimate.)	
--	---	--

	10. A. A factory records machine breakdown counts per week. Given data = [3,4,2,5,3,4,6,2,3,5]. Find average machine failures/week. (Hint: Use MLE) B. Given the following data battery-life = [9.8,10.2,9.7,10.1, 10.3,9.9,10.0, 10.2], the company claims the battery lasts 10 hours. Test the hypothesis and draw conclusion(Hint : find Z-static and p-Value) C. Given the student score before and after coaching before = [45,50,48,47, 49] after = [55,58,52,54,57] compare them. (Hint : Use t-test and find t-static and p-value) 11. Given the details of sales in three regions of a company, north = [25,28,30,27,29] south = [32,35,31,30,34], east = [20,22,19,24,21], compare the data and find conclusion using ANOVA test.(Hint: use f-oneway()) Load and display a CSV dataset. 12. A retail company wants to check whether gender and product preference are related. Perform Chi-square test and draw conclusion. Two product categories are there viz., Product A = Smart Watches and Product B = Wireless Earbuds. Given data = {"Gender": ["Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female", "Male", "Female"], "Preferred_Product": ["Smart Watch", "Wireless Earbuds", "Smart Watch", "Wireless Earbuds", "Smart Watch", "Wireless Earbuds", "Wireless Earbuds", "Wireless Earbuds", "Smart Watch", "Wireless Earbuds"] } 13. A. Analyze relationship between advertising expense and sales given the data advertising = [10,15,20,25,30,35,40], sales = [20,25,28,35,40,45,50] . draw a scatter plot and explain your findings. B. Find correlation between hours studied and exam marks given the data hours = [1,2,3,4,5,6,7,8] and marks = [35,40,50,55,60,65,70,80] . (Hint: Pearson) C. Compare rankings given by two judges in a dance competition. Given the data judge1 = [1,2,3,4,5], judge2 = [2,1,3,5,4] . Find Spearman correlation and p-value and explain the findings. 14. Predict salary based on years of experience given the data experience = [1,2,3,4,5] and salary = [25000,30000,35000,40000,45000]. Find slope, intercept and predicted salary for 6 years. Justify the answer. 15. Predict future sales using advertising budget.given the data ads = [5,10,15,20,25]), sales = [50,60,70,85,95] 16. Predict house price using: Given the data for Area, Bedrooms and Age of house as data = { "Area": [1000,1200,1500,1800,2000], "Bedroom": [2,2,3,3,4], "Age": [10,8,6,5,3], "Price": [50,55, 65,72,80] } , predict price where the Area=1600, Bedrooms=3, Age=4.	
--	---	--

Essential Readings:

1. Peter Bruce & Andrew Bruce ,Practical Statistics for Data Scientists -50 Essential Concepts, O'Reilly Media, Inc.
2. Laura Igual, Santi Seguí, Introduction to Data Science A Python Approach to Concepts, Techniques and Applications Second Edition

3. Peter C. Bruce, Peter Gedeck and Janet Dobbins ,Statistics for Data Science and Analytics, WILEY Publications.
4. Jesús Rogel-Salazar, Statistics and Data Visualisation with Python ,CRC Press, Taylor and Francis Group
5. Miller I Freund J E, Probability and Statistics for Engineers,Prentice Hall of India.
6. Gupta, S.P. Statistical Methods. Sultan Chand and Sons: New Delhi.

Web Resources:

1. <https://www.w3schools.com>
2. <https://docs.scipy.org/doc/>
3. <https://sklearn.org/stable/>
4. <https://pandas.pydata.org/docs/>

Sample questions for Theory examination:

1. Define statistics. Explain its importance in real-world applications.
2. Differentiate between descriptive statistics and inferential statistics with examples.
3. Explain population and sample with suitable examples.
4. Why is random sampling important in statistics?
5. Explain mean, median, and mode. When should each be used?
6. Explain quartiles and percentiles with applications.
7. What is skewness? Explain positive and negative skewness with suitable examples.
8. What is regression analysis?
9. A battery manufacturing company claims that its batteries last 10 hours on average. Explain how a Z-test can be used to verify the claim.
10. A retail company compares sales performance in three different regions. Explain how ANOVA helps determine whether significant differences exist.
11. A real-estate company predicts house price using: area,number of bedrooms, age of house. Explain how multiple linear regression can be applied in this scenario.
12. Two judges rank participants in a dance competition differently. Explain the importance of Spearman rank correlation in ranking systems.

13. A bank wants to identify the top-performing customers based on account balance. Explain the role of quartiles and percentiles in this analysis.
14. A spam-filtering system predicts whether an email is spam based on previous data. Explain how Bayes' theorem can help in this prediction.
15. Explain the difference between PMF and PDF using the following examples:
 - Number of defective products
 - Daily rainfall amount

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

Employability for the Course: High, in Data Science

KU5DSECAP302: INTRODUCTION TO DATA SCIENCE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP302	4 (3T+ 1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description:

The course **Introduction to Data Science** provides fundamental knowledge of data collection, processing, analysis, and visualization techniques used in modern computing environments. The course introduces students to the concepts of Data Science, data analytics, statistical analysis, data preprocessing, and visualization methods. It also familiarizes students with essential tools and technologies such as Python, NumPy, Pandas, spreadsheets, and Tableau Public. The course emphasizes practical applications of Data Science through simple programming exercises, statistical techniques, and visualization methods.

Course Prerequisite: Foundations of Computers and Programming

KU4DSCCAP208 Database Management System

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts, applications, and ethical issues related to Data Science and identify various types and sources of data.	U
2	Apply data collection, preprocessing, cleaning, transformation, and reduction techniques using basic tools and Python libraries such as NumPy and Pandas.	A

3	Perform basic statistical analysis and data analytics operations including measures of central tendency, dispersion, correlation, and spreadsheet-based analytics.	A
4	Create effective data visualizations using charts, graphs, spreadsheets, and Python visualization libraries such as Matplotlib and Seaborn.	An
5	Develop simple data-driven applications and analytical solutions using basic Data Science tools and techniques for real-world problem solving.	C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	1	1	3	1	2	2
CO 2	3	3	3	2	1	1	2
CO 3	3	2	3	2	1	1	2
CO 4	3	3	2	2	1	2	2
CO 5	3	3	3	2	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Fundamentals of Data Science		10
	1	Introduction to Data Science -Evolution and Importance of Data Science -Data Science Lifecycle -Responsibilities of a Data Scientist -Types of Data: Structured Data, Semi-structured Data, Unstructured Data.	
	2	Sources of Data, Big Data, Basic Applications of Data Science	

	3	Fundamental Areas of Data Science(Basic idea only): Machine Learning, Deep Learning, Natural Language Processing (NLP) Statistical Data Analysis, Knowledge Discovery and Data Mining ,Text Mining,Recommender Systems, Data Visualization.	
	4	Data Science Tools(basic idea only)- Python Programming, R Programming, SAS, Spreadsheet Tools, Tableau Public. Ethical Issues in Data Science- Data Privacy, Data Security, Bias and Fairness Responsible Use of Data .	
2	MODULE TITLE: Data Collection and Preprocessing		11
	1	Introduction to Data Collection, Data Types and Forms, Possible Data Error Types .	
	2	Data Collection Methods: Surveys, Sensors, Databases, Web Data, CSV Files. Data Preprocessing Operations: Data Cleaning, Handling Missing Values, Smoothing Noisy Data, Detecting and Removing Outliers.	
	3	Data Integration: Virtual Integration ,Physical Data Integration, Application-Based Integration ,Manual Integration,Middleware Data Integration .	
	4	Data Transformation: Rescaling Data, Normalizing Data, Binarizing Data, Standardizing Data .	
	5	Data Reduction Techniques: Dimensionality Reduction, Data Cube Aggregation, Numerosity Reduction Discretization: Top-Down Discretization, Bottom-Up Discretization Python Libraries for Data Processing: Introduction to NumPy , Introduction to Pandas, Importing and Exporting Data, Basic Data Manipulation using Python	
3	MODULE TITLE: Data Analytics and Statistical Analysis		12
	1	Introduction to Data Analytics,Descriptive Statistics: Mean, Median, Mode ,Standard Deviation, Variance, Skewness, Kurtosis, Correlation Statistics .	

	2	Spreadsheet-Based Analytics: Introduction to Worksheets,Worksheet creation and Formatting ,Ranges and Tables ,Data Cleaning with Text Functions ,Working with Date and Time Values ,Conditional Formatting, Sorting and Filtering ,Subtotals with Ranges ,Creating Macros ,Pivot Tables Lookup Functions: LOOKUP ,VLOOKUP ,HLOOKUP .	
	3	Types of Jobs in Data Analytics: Data Analyst ,Data Scientist ,Data Engineer, Database Administrator ,Data Architect ,Analytics Manager .	
4	MODULE TITLE: Data Plotting and Visualization		12
	1	Introduction to Data Visualization, Importance of Data Visualization Basic Data Visualization Tools: Charts and Graphs: Bar Chart ,Pie Chart,Histogram ,Line Graph ,Scatter Plot ,Area Plot ,Box Plot	
	2	Types of Data Visualization: Temporal Data Visualization ,Hierarchical Data Visualization ,Network Data Visualization,Multi-dimensional Data Visualization ,Geospatial Data Visualization ,Multivariate Data Visualization	
	3	Advanced Visualization Techniques: Band Chart ,Thermometer Chart ,Gantt Chart,Waterfall Chart ,Pivot Charts Visualization using Python: Introduction to Matplotlib, Introduction to Seaborn	
5	Teacher Specific Module :		30
	<i>Recommended Lab Exercise</i>		
	1. Install Python and Jupyter Notebook/Google Colab. Write simple Python programs: - Program to input and Display Student Details - Program to get a number and Generate Multiplication Table of it - Program to Find Sum of Elements in a List(with and without using built in function) - Program to Sort Numbers in Ascending Order(with and without using built in function) 2. Download a CSV dataset from Kaggle/UCI, Load the dataset using Pandas, Display first 10 rows, Sort Dataset 3. Program to Create and Read Data from a CSV File 4. Program to Display Selected Columns from Dataset, Filter Rows Based on Conditions		

	5. Program to Normalize and standardize a dataset column using Python. 6. Program to Create NumPy Arrays , Array Addition and Multiplication 7. Program to Find Maximum and Minimum in an Array 8. Calculate statistical measures.Mean, Median, Mode, Standard deviation and Variance using a student marks dataset 9. Creating Worksheets and Formatting Tables, Applying Conditional Formatting, Sorting and Filtering Data , Using LOOKUP, VLOOKUP, and HLOOKUP Functions , Creating Pivot Tables and Pivot Charts 10. Create a Bar Chart, Pie Chart, Line Graph, Histogram, Scatter Plot for Student Performance Data created in a spreadsheet 11. Case study a. On Data privacy and Responsible use of data b. A short report on applications of Data Science in: Healthcare, Banking, Education, Social Media, E-commerce	
--	--	--

Essential Readings:

1. Introduction to Data Science — Laura Igual & Santi Seguí — Springer Publications
2. Data Science for Business — Foster Provost & Tom Fawcett — O'Reilly Media
3. Python for Data Analysis — Wes McKinney — O'Reilly Media
4. Data Mining: Concepts and Techniques — Jiawei Han, Micheline Kamber & Jian Pei — Morgan Kaufmann Publishers
5. Practical Statistics for Data Scientists — Peter Bruce & Andrew Bruce — O'Reilly Media
6. Data Analytics Using Excel — Conrad Carlberg — Pearson Education
7. Fundamentals of Data Visualization — Claus O. Wilke — O'Reilly Media
8. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow — Aurélien Géron — O'Reilly Media

Suggested Readings:

- 1.Data Science: A Beginner's Guide-C. Raju-Penguin Random House India
- 2.Data Science and Analytics (with Python, R and SPSS Programming)Dr. V.K. Jain-Khanna Publishing House.
- 3.Introduction to Artificial Intelligence & Data Science Puneet Kumar & Namrata Chawla Kalyani Publishers.
- 4.Fundamentals of Data Science for BCA Chitra Ravi & K. Murugan Sunstar Publisher

Assessment Rubrics:

Evaluation Type	Marks
-----------------	-------

End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	4
c)	Viva/Seminar	5
d)	Case study	4
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- **Employability for the Course:** Data Analyst, Data Scientist, ML Engineer etc.,

KU6DSECAP303: DATA MINING CONCEPTS AND ALGORITHMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP303	4 (4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	–	30	70	100	2

Course Description: This course provides a comprehensive introduction to the concepts, techniques, and applications of Data Mining along with a foundational understanding of Data Warehousing. It focuses on the process of discovering meaningful patterns, trends, and knowledge from large volumes of data to support effective decision-making. It further explores key data mining functionalities such as data preprocessing, data cleaning, data transformation, and data reduction, core data mining techniques including classification, clustering, and association rule mining

Course Prerequisite: Introduction to Data Science

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understanding the importance of data mining and data warehousing.	U
2	Understand the data management aspects data preprocessing model	U
3	Familiarise the Data mining Algorithms of different Data mining techniques	U,An
4	Understand the basics of Data warehousing	U

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	1	3	2	2	1	2
CO2	3	2	3	2	2	1	2
CO3	3	3	2	2	2	1	2
CO4	2	1	3	2	1	1	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	INTRODUCTION TO DATA MINING AND DATA WAREHOUSING		12
	1	Data mining – Definitions, KDD vs data mining, DBMS vs data mining	
	2	DM Techniques, Applications of Data Mining	
	3	Data warehousing – What is Data Warehousing, Characteristics, Multidimensional data model, OLAP operations	
	4	Data warehouse schema, Data warehousing Architecture, Data warehouse Backend Processes.	

2	ASSOCIATION RULE MINING		12
	1	Association rules – What is Association Rules, Terminologies - Frequent Itemsets, Closed Itemsets, and Association Rule, Support, Confidence, Lift,	
	2	Properties of Frequent Itemset, A priori algorithm, partition algorithm(Concept only),	
	3	Pincer search algorithm(Concept only), FP-tree growth algorithm	
	4	Generating Association Rules from Frequent Itemsets, Pattern Mining in Multilevel, Multidimensional Space	

3	CLASSIFICATION		12
	1	Basic Concepts: What is Classification? General Approach to Classification	
	2	Decision Tree Induction, Attribute Selection Measures, Tree Pruning	
	3	Data Preprocessing: An Overview- Data Quality: Why Preprocess the Data? Major Tasks in Data Preprocessing	
	4	Data Cleaning: Missing Value, Noisy Data,	

4	CLUSTERING		12
	1	Cluster Analysis : What Is Cluster Analysis, Requirements for Cluster Analysis, Overview of Basic Clustering Methods	
	2	Partitioning Methods : k-Means: A Centroid-Based Technique, k-Medoids: A Representative Object-Based Technique	
	3	Hierarchical Methods: Agglomerative versus Divisive Hierarchical Clustering	
	4	Density-Based Methods : DBSCAN: Density-Based Clustering Based on Connected Regions with High Density	

5	Teacher Specific Module :		12
	<i>Activities for Assessment</i>		
	Task 1: Design a Data Warehouse (Basic): - Choose a simple domain (e.g., College, Supermarket, Hospital) - Identify: - Fact table (e.g., Sales, Student Performance) - Dimension tables (e.g., Time, Product, Student) - Draw a Star Schema Expected Output: - Diagram (hand-drawn or using tools), Short explanation Task 2: Data Preprocessing Exercise: - Given a dataset (with missing or inconsistent values):		

	○ Identify missing data ○ Remove duplicates ○ Normalize or format data Tools: Excel / Python (optional) Expected Output: ● Before & after dataset, Explanation of changes Task 3: Classification Task ● Use a simple dataset (e.g., student pass/fail) ● Use DT and the splitting index values to show the best splitting ● Explain how classification works Expected Output: ● Steps followed, Result and interpretation Task 4: Association Rule Mining ● Use sample transaction data (e.g., supermarket items) ● Find: ○ Frequent itemsets ○ Association rules (e.g., Bread → Butter) Expected Output: ● Rules generated, Support and confidence (basic level) Task 5: Data Visualization Task ● Use dataset (sales/student data) ● Create: ○ Bar chart, Pie chart, Line graph Tools: Excel / Tableau / Python Expected Output: ● Graphs, Insights from data	
--	--	--

Essential Readings:

1. Arun K. Pujari, Data Mining Techniques, 2nd Ed, University Press
2. Jiawei Han, Micheline Kamber and Jian Pei, Data Mining: Concepts and Techniques, 3rd Ed, Morgan Kaufmann
3. Data Mining and Data Warehousing: Principles and Practical Techniques, By Parteek Bhatia, Cambridge
4. <https://www.geeksforgeeks.org/data-science/data-mining/>

Suggested Readings:

1. Margaret H. Dunham, Data Mining - Introductory and Advanced Topics, Pearson
2. <https://www.nptel.ac.in/courses/106106853>
3. <https://cse.iitkgp.ac.in/~pabitra/course/cs698v.html>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	5
d)	Activities	5
Total		100

o Employability for the Course:

- o After completing this course, students can pursue entry-level roles such as Data Analyst, Junior Data Scientist, Business Analyst and Market Research Analyst etc.,
- o Students gain skills applicable in real-world domains like E-commerce → Product recommendation, customer behavior analysis, Healthcare → Disease prediction, patient data analysis and Education → Student performance prediction etc.,

KU6DSECAP304: MACHINE LEARNING USING PYTHON

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP304	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces fundamental concepts of machine learning, including regression, classification, and clustering techniques. It focuses on building basic predictive models and analyzing data using Python and suitable examples.

Course Prerequisite: Basic knowledge in Statistics

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of Machine Learning, including types of learning (supervised, unsupervised), applications, and basic workflow of ML systems.	U
2	Apply regression techniques to build predictive models and analyze relationships between variables for real-world datasets	A
3	Implement classification algorithms to categorize data and evaluate model performance using appropriate metrics.	C
4	Analyze and apply clustering techniques to group similar data points and interpret patterns in unlabeled datasets.	An

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
------------	------	------	------	------	------	------	------

CO1	3	2	1	2	2	1	3
CO2	3	3	2	2	2	1	2
CO3	3	3	2	3	2	1	2
CO4	3	2	2	3	3	1	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	INTRODUCTION TO MACHINE LEARNING		10
	1	What is Machine Learning, Machine Learning Vs. Traditional Programming	
	2	How Machine Learning Works? Applications of Machine Learning	
	3	Types of Learning, Supervised Learning, Unsupervised Learning, Semi-supervised Learning	
	4	Reinforcement Learning, Active Learning, Challenges in Machine Learning	
	5	Machine Learning Pipeline overview	
	6	Introduction to Data, Types of Data - Numerical Data, Categorical Data Data Collection Methods, Introduction to Datasets, Importing Datasets using Python Data Cleaning Basics, Handling Missing Values, Data Visualization Basics Splitting Training and Testing Data	
2	REGRESSION, CLASSIFICATION		11
	1	Introduction to regression, classification, real world use cases	

	2	Types of Regression	
	3	Error and Mean Squared Error for Regression	
	4	Linear Regression, Gradient descent (Concept only), Multiple Linear Regression	
	5	Non-Linear Regression (Polynomial Regression)	
	6	Error in classification, Evaluation Measures : Accuracy, Precision, Recall, F1-Score, Confusion Matrix	
	7	K-Nearest Neighbours (KNN algorithm)	
	8	Logistic Regression	
3	CLASSIFICATION		12
	1	Naïve Bayes Classification, Multinomial Naïve Bayes Classification	
	2	Decision Trees, ID3 algorithm	
	3	Artificial Neural Network, Biological Neuron and Artificial Neuron	
	4	Building Blocks: Input layer, hidden layers, output layer, weights, bias, activation functions: Step, Linear, Sigmoid, Tanh, and ReLU functions, Illustration of Simple Neural Network.	
	5	Single Layer Perceptron	
	6	Multilayer neural network, Loss function	
	7	Importance of Backpropagation (concepts only)	
4	CLUSTER ANALYSIS		12
	1	Introduction, Requirements of Clustering, Categorization of Major Clustering Methods	
	2	Types of Data in Cluster Analysis: Interval-Scaled Variables, Binary Variables, Categorical Variables, Ordinal Variables, Ratio-Scaled Variables, Variables of Mixed Types	
	3	Partitioning Methods: K-means, K-medoids, CLARANS	
	4	Hierarchical Methods : Agglomerative Clustering, Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)	
	5	Density-based Methods: DBSCAN	
5	Teacher Specific Module : Lab experiments		30
	<i>Directions</i>		

	1. To check whether linear relationship exists between two variables, [Refer Textbook 2] a. Stock_Index_Price & Interest_Rate b. Stock_Index_Price & Unemployment_Rate c. Use Pandas for data and Matplotlib for visualization 2. Consider a stock market data and implement multiple linear regression [Refer Textbook 2] a. Use Pandas for data and sklearn and statsmodel for insights 3. Polynomial Regression using Python a. Aim: To implement Polynomial Regression for salary prediction based on experience. Logistic Regression using Python b. Aim : To implement Logistic Regression for predicting student pass/fail status. 4. Implement K-nearest neighbour algorithm and show accuracy on the dataset 5. Use the dataset 'diabetes.csv' and predict whether or not a patient has diabetes based on certain diagnostic measurements given in the dataset. [Refer Textbook 2] 6. For the given dataset [age, income, student, credit_rating, buys_computer] , using Naive Bayes Model predict the class of a new data. [Refer Textbook 2] 7. Using a classifier, classify the 'iris' flowers, use the well-known data set iris.csv [Refer Textbook 2] 8. Build a simple Multi-Layer Perceptron using the library Scikit-learn. Train on small dataset (Iris dataset) 9. Implement the ID3 algorithm using Decision Tree Classifier with entropy. Classify weather conditions for playing outdoor games 10. Implement DBSCAN Clustering to cluster a set of random numbers generated 11. Implement K-Means Clustering to cluster a set of data, a simple customer dataset containing Age & Annual Income	
--	---	--

Essential Readings:

1. Introduction to Machine Learning with Python, Andreas C. Müller & Sarah Guido, O'Reilly
2. Introduction to Machine Learning using Python, Dr. Jeeva Jose, Khanna books
3. Introduction to Machine Learning, Second Edition, Ethem Alpaydm, The MIT Press Cambridge

Suggested Readings:

1. https://python-course.eu/books/bernd_klein_python_and_machine_learning_a4.pdf
2. https://www.w3schools.com/python/python_ml_getting_started.asp
3. <https://nptel.ac.in/courses/106105152>
4. <https://www.eicta.iitk.ac.in/courses/machine-learning-with-python>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- **Employability for the Course:**

- Students completing this course can work as Junior Machine Learning Developer, Python Programmer, AI/ML Intern, Data Science Intern, Research Assistant in AI/ML Projects because the course helps in developing Industry-Relevant Skills such as Python programming for data analysis, Data preprocessing and visualization, Building regression and classification models, Using machine learning libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn, Problem-solving and analytical thinking

Artificial Intelligence & Machine Learning

KU5DSECAP305: MATHEMATICAL FOUNDATIONS OF MACHINE LEARNING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP305	4 (3T+1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1Hr 30 Minutes

Course Description: This course introduces the basic mathematical concepts required for AI and Machine Learning, including linear algebra, probability, statistics, and basic calculus. It focuses on developing conceptual understanding through real-world computing and data-based examples, emphasizing how mathematical methods are used for data representation, analysis, and problem-solving in computer systems.

Course Prerequisite: KU3DSCCAP201 Discrete Mathematics, Basic mathematics

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	To understand mathematical structures used in computer systems for representing, analyzing, transforming, and interpreting data.	U
2	Apply statistical and probabilistic concepts for analyzing uncertainty and relationships in data	A
3	Interpret functions, derivatives, and gradients in computational systems	An
4	Analyze optimization and prediction concepts used in machine learning models	An
5	Apply mathematical reasoning to simple data-driven computing problems	A

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	2	2	1	2
CO2	3	2	3	2	2	1	2
CO3	3	2	2	2	2	1	2
CO4	3	3	2	3	3	1	3
CO5	3	2	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
		All topics in this course should be taught using real-world computing and data-driven examples to help students understand how mathematical concepts are used for data representation, uncertainty modeling, change analysis, and decision-making rather than purely abstract formulations	
		Linear Algebra and Calculus for Computing	11
	1	Linear Algebra for Data Representation : Scalars, vectors, matrices : How data is represented	
	2	Vector operations (dot product, magnitude) :Computation of relationships such as similarity and strength between data points	
	3	Matrix operations (addition, multiplication, transpose)	
	5	Identity and inverse (conceptual)	
	6	Eigen value, Eigen Vector : geometric interpretation of how data directions behave under transformations	
	7	System of linear equations: modeling and solving relationships between multiple variables	
	8	Calculus for Modeling and Optimization	
	9	Functions and graphs, limits (intuitive idea only)	

	10	Differentiation: Concept, Rate of change, slope of a function, Visualisation	
	11	Partial Derivatives: Concept, Derivatives w.r.t. one variable in multivariable functions, Gradient	
	12	Chain rule (conceptual idea): how change in one variable propagates through related variables	
2	Data Handling and Descriptive Statistics		12
	1	Types of data (qualitative, quantitative)	
	2	Data classification (nominal, ordinal, interval, ratio)	
	3	Frequency distribution	
	4	Graphical representation (bar, histogram, pie chart, boxplot)	
	5	Measures of central tendency (mean, median, mode)	
	6	Measures of dispersion (variance, standard deviation)	
	7	Measures of shape (skewness, symmetry)	
	8	Covariance : how two variables vary together	
	9	Correlation : strength and direction of relationships between variables	
	10	Correlation coefficient and interpretation	
	11	Scatter plots and trend visualization	
3	Probability for Uncertainty Modeling		12
	1	Sample space and events, Union, intersection, complement	
	2	Mutually exclusive and exhaustive events	
	3	Basic probability rules	
	4	Conditional probability, Bayes theorem	
	5	Random variables (discrete & continuous), PMF and PDF	
	6	Probability Distributions (Normal, Binomial, Poisson)	
	7	Hypothesis testing (concept only)	
4	Optimization		10
	1	Regression concept: fitting relationships between variables	
	2	Cost/Loss functions (MSE, Cross-entropy)	

	3	Linear regression intuition using error minimization	
	4	Minimization and maximization, Iterative improvement	
	5	Gradient descent : How does error change when we slightly change values?	
	6	Learning rate	
	7	Overfitting vs underfitting mathematical intuition of model complexity and generalization	
	8	Regularization (L1, L2 basic idea): controlling model complexity to reduce overfitting	
5	Teacher Specific Module :		30
	Directions : Use GeoGebra, Jamovi		
	Explore fundamental vector operations—addition, scalar multiplication, dot product, and transformation—using GeoGebra.		
	$\vec{u}=(3,2)$		
	$\vec{v}=(1,4)$		
	Using GeoGebra:		
	1. Visualize both vectors from the origin. 2. Perform and display vector addition 3. Apply scalar multiplication 4. Compute the dot product of $\vec{u}$ and $\vec{v}$ and interpret the result geometrically. 5. Apply a matrix transformation to rotate vector $\vec{u}$ by 90 degrees using matrix multiplication		
	Interpret the visual and numerical results of each operation		
	Understand the geometric meaning of eigenvalues and eigenvectors.		
	1. Apply matrix transformation to vectors 2. Observe stretching and rotation 3. Identify eigenvector directions visually 4. Interpret eigenvalues geometrically		
	Study behavior of functions graphically.		

1. Plot and analyze basic functions graphically.
2. Observe slope changes
3. Explore intuitive idea of limits

Visualize the derivative (slope of a tangent) of a function using GeoGebra.

<https://www.geogebra.org/classic>

Visualize and compute partial derivatives of a multivariable function using GeoGebra's 3D Graphing environment and understand the geometric meaning of partial derivatives.

<https://www.geogebra.org/3d>

Visualize probability distribution in Geogebra Plot

1. Normal distribution
2. Binomial distribution
3. Poisson distribution

Observe shape and spread

<https://www.geogebra.org/classic>

Set up Lab for Statistics

Install Jamovi. Jamovi is a free, open-source, user-friendly statistical software

Download from the official website:

<https://www.jamovi.org>

Analyze the study habits of students by examining their daily study hours and corresponding exam scores. Understand the central tendency, variability, and distribution of these variables.

Also interpret Output (Average study hours and exam scores, Middle value when data is sorted, How spread out the data is, Whether data is symmetric or skewed, Visualization Insights: **Histogram**: Visualize distribution shape (normal, skewed).

Boxplot: Identify median, spread, and outliers.

Hint : Prepare Your Data with two columns Study_Hours, Exam_Score

Sample Data

Study_Hours	Exam_Score
2	55
4	70
3	65
.....	

By varying the values of study hours and corresponding exam scores for a group of students, visualize different types of data distributions (such as symmetric, right-skewed, and left-skewed) using histograms in Jamovi, and analyze how changes in data affect the shape, central tendency, and spread of the distribution.

Analyze the number of daily emails received by employees in a software company to understand communication load. Use descriptive statistics and histograms in Jamovi to determine the central tendency, variation, and distribution shape of email traffic, and interpret the implications for system design or user productivity. Comment on the shape of the distribution and central tendency. Determine the probability that more than 20 emails are received on a given day.

Either either data or import .csv

Sample Data

Day	Emails_Received
1	15
2	12
3	18
.....	

Exploratory Data Analysis on Web User Activity Data

Explore the distribution and patterns in user session durations based on age group and device type to gain insights for web optimization. Compare average session times between age groups and also analyze device influence.

Sample data. Either manually type in Jamovi or Load .csv file

User_ID	Session_Duration	Age_Group	Device_Type
-----	-----	-----	-----
U001	5	18–25	Mobile
U002	10	26–35	Desktop
U003	12	18–25	Mobile
U004	8	36–45	Tablet
U005	15	26–35	Desktop
U006	20	18–25	Mobile
U007	18	26–35	Desktop
U008	25	36–45	Tablet
U009	6	18–25	Mobile
U010	35	26–35	Desktop

Correlation Analysis of Coding Practice vs. Exam Scores

A computer science instructor wants to understand the relationship between the number of hours students practice coding and their final exam scores in a programming course. Additionally, the instructor wants to analyze if the rank order of students by coding hours correlates with their rank order by exam scores. Plot Scatter Diagram. Interpret Scatter Diagram, Pearson's

Sample dataset

Student_ID	Coding_Hours	Exam_Score
-----	-----	-----
S001	5	55
S002	12	78
S003	8	65
S004	15	85
S005	7	62
S006	10	70
S007	3	50

S008	20	90	
S009	6	58	
S010	11	75	

Predicting Student's Programming Exam Score Based on Coding Practice Hours Using Linear Regression

For the above problem to find a linear relationship between practice hours (independent variable) and exam score (dependent variable).

Predict the **daily temperature** using weather historical data. Key influencing factors considered are:

- **Humidity**
- **Wind Speed**
- **Atmospheric Pressure**

Build a multivariate linear regression model using Jamovi to estimate the temperature and interpret the results.

Sample Data

Day	Humidity	Wind_Speed (km/h)	Pressure	Max_Temperature	
---	-----	-----	-----	-----	
1	60	15	1012	32	
2	70	12	1010	30	
3	55	20	1015	34	
4	80	10	1008	28	
5	65	18	1011	31	
6	50	22	1016	35	

Sample Question

- 1) Let the vector $\vec{u} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

A transformation is applied using the matrix

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- a) Find the result of the transformation A. $\vec{u}$
b) Describe what happens to the direction and length of the vector after transformation.

2)

How does the **sign** (positive, zero, or negative) of the dot product relate to the **angle between vectors**?

3) Why is dot product important when comparing two directions, such as the direction of movement of an object and the direction of a force applied to it?

4) How does scalar multiplication affect the **magnitude** and **direction** of a vector?

5) A car's position over time is given by a graph. The slope at any point on the graph represents its speed. What does a **positive**, **zero**, or **negative** slope indicate about the car's motion?

5) $f(x,y)=x^2+y^2$ represents the "elevation" of a surface. What does $\partial f/\partial x$ represent in this case?

7) For the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Find the eigenvalues and interpret the dominant eigenvector geometrically.

5) Explain what it means for two events to be mutually exclusive with an example.

Are the events "rolling a 2" and "rolling an even number" mutually exclusive? Why or why not?

6) Why is **variance** important in understanding data spread?

7) Describe one real-life scenario where the **binomial distribution** is useful.

8) You are given the following dataset representing the scores of 12 students in a programming test:

56,72,68,45,89,77,54,62,70,80,66,75
56, 72, 68, 45, 89, 77, 54, 62, 70, 80, 66, 75

Calculate the mean, median, and mode of the data.

Find the range, variance, and standard deviation of the scores.

Interpret the results:

- What does the mean score tell you about the overall performance?
- How does the median compare with the mean, and what might that indicate about the distribution?

- What does the standard deviation tell you about the consistency of student scores?
- Are there any signs of skewness or outliers from your calculations or frequency distribution?

9) You have collected data on the number of hours studied and the marks obtained by 8 students in an exam:

Student	Hours	Marks
	Studied (X)	Obtained (Y)
1	2	50
2	3	55
3	5	65
4	4	60
5	6	70
6	8	85
7	7	75
8	9	90

Plot a scatter diagram of the data (Hours Studied vs Marks Obtained).

Based on the scatter plot, describe the nature of the relationship between hours studied and marks obtained.

10) For the given dataset, Regression Line is $Y=105+15X$

Day (X)	Number of Users
	Visiting a Website (Y)
1	120
2	135
3	150

	4	165	
	5	180	
	6	195	
	7	210	
	a) What is the meaning of the slope =15 in this context? b) What does intercept=105 represent in this scenario? c) Using the regression line, predict the number of users visiting the website on day 8. d) If the website team wants to reach 300 users, on which day does the regression model predict this will happen? e) How would you assess whether this linear model is a good fit for the data?		

Essential Readings:

1. Mathematics for Machine Learning, Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong, Cambridge University Press., 2020
2. Pal Nabendu and Sarkar Sahadeb, Statistics: Concepts and Applications, Second Edition, PHI, 2013

Web Resources

1. <https://nptel.ac.in/courses/111106112>
2. <https://nptel.ac.in/courses/111105041>
3. <https://www.youtube.com/playlist?list=PLblh5JKOoLUK0FLuzwntyYI10UQFUhsY9>
4. <https://www.youtube.com/playlist?list=PLblh5JKOoLUK0FLuzwntyYI10UQFUhsY9>
5. <https://www.khanacademy.org/math/linear-algebra>
6. <https://www.3blue1brown.com/>

Assessment Rubrics:

Evaluation Type	Marks
End Semester Evaluation	65
Theory	50

Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case Study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU5DSECAP306: ARTIFICIAL INTELLIGENCE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP306	4 (4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	0	30	70	100	2 Hours

Course Description: This course introduces the fundamental concepts of Artificial Intelligence, including intelligent agents, knowledge representation, and problem-solving techniques. It covers key AI domains such as machine learning, natural language processing, and computer vision, along with emerging areas like generative AI. The course emphasizes real-world applications, basic reasoning methods, and ethical considerations, providing a foundation for understanding and applying AI in practical scenarios

Course Prerequisite: KU3DSCCAP201 Discrete Mathematics

Course Outcomes:

CO No	Expected Outcome	Learning Domains
1	Understand fundamental concepts of intelligent systems.	U
2	Formulate problems as a state space and apply various search algorithms to identify optimal solutions.	A
3	Understand effective knowledge representation ways.	U
4	Ability to design and develop expert systems	An C
5	Obtain a basic understanding of the AI domains and their applications and examine the legal and ethical issues of AI	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	1	2	2	1	3
CO2	3	3	2	2	2	1	3
CO3	3	2	2	3	2	1	2
CO4	3	3	2	3	3	2	3
CO5	3	2	1	3	2	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE Introduction to AI		12
	1	Artificial Intelligence: Definition	
	2	Concept of Artificial General Intelligence (AGI)	
	3	History of Artificial Intelligence, State of the Art	
	4	Domains in AI: Machine Learning, Computer Vision, Robotics, Natural Language Processing, Deep Neural Networks, and their Applications using real-world use cases	
	5	Introduction to Generative AI (concept only): overview and real-world use cases	
	6	Structure of Agents: Agent Programs, Simple Reflex Agent, Model Based Reflex Agent, Goal Based Agent, Utility Based Agent, Learning Agent, Knowledge based agents (Concept Only, use real-world use cases)	
	7	Case Study: Agent Problem Solving using Wumpus World	
2	MODULE TITLE: Knowledge Representation		12

	1	Logic: Propositional logic, First-order predicate logic, Propositional versus first-order inference	
	2	Unification.	
	3	Application: Logic based Financial Advisor (Refer Essential Readings (1)	
	4	Knowledge Representation Structures: Frames, Scripts, Semantic networks, Ontology, Knowledge graphs (Examples and use cases such as search engines, chatbots, and expert systems)	
3	MODULE TITLE Problem Solving as Search		12
	1	Production systems, Control strategies, Problem characteristics, Production system characteristics, illustration using Water Jug problem	
	2	Informed Search, Introduction to Heuristics, Admissible heuristics, Best First Search, A*	
		Adversarial Search & Games: Two-player zero-sum games, Minimax Search, Alpha-Beta pruning.	
	3	Constraints and Constraint Satisfaction Problems : Crypt arithmetic problem, Map Coloring, N-Queens Problem, Task Scheduling (Illustration only, algorithms not required)	
4	MODULE TITLE Planning, Expert Systems, Responsible AI		12
	1	Planning, Overview, components of a planning system, Goal stack planning	
	2	Architecture of expert systems, Rule Based Expert Systems	
	3	Weak AI and Strong AI: concept, differences, and examples of current AI systems	
	4	AI Ethics and Responsible AI	
5	Teacher Specific Module :		12
	Emerging Topics Awareness Guidance to students for Case study Case study guidelines Ask students to select a simple, real-world AI application and explain its concept, working, and use cases. Select a case study like - Intelligent Agents (Robot navigation) - Expert Systems (e.g., medical or financial advisor) - Chatbots / NLP systems		

	● Recommendation systems ● Constraint satisfaction problems (e.g., scheduling, map coloring) ● Real-world AI applications (e.g., Google Search, self-driving cars) Evaluation Criteria ● Understanding of AI concepts ● Relevance of case study ● Clarity of explanation ● Real-world application understanding ● Presentation and organization	
--	---	--

Essential Readings:

1. George F Luger, Artificial Intelligence – Structures and Strategies for Complex problem solving, 5thEdn, Pearson.
2. Russell, S and Norvig, P, 2015, Artificial Intelligence – A Modern Approach, 3rd Edition, Prentice Hall.
3. Elaine Rich, Kevin Knight, Shivashankar B Nair., Artificial Intelligence”, 3rd Edition, McGraw Hill Education, 201

Suggested Readings:

1. Nilsson N.J., Artificial Intelligence - A New Synthesis, Harcourt Asia Pvt. Ltd.
2. M.C. Trivedi, A Classical Approach to Artificial Intelligence, Khanna Book Publishing Company, 2024
3. Dan W Patterson, Introduction to Artificial Intelligence & Expert Systems, PHI Learning 2010.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Case study	10

Total	100
-------	-----

- **Employability for the Course:** The course enhances employability by providing foundational knowledge of AI concepts and applications, enabling students to pursue entry-level roles and higher studies in AI and data-driven technologies

KU6DSECAP307: MACHINE LEARNING USING PYTHON

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP307	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1:30

Course Description: *This course introduces fundamental concepts of machine learning, including regression, classification, and clustering techniques. It focuses on building basic predictive models and analyzing data using Python and suitable examples.*

Course Prerequisite: Statistical Methods for Data Science using Python

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of Machine Learning, including types of learning (supervised, unsupervised), applications, and basic workflow of ML systems.	U
2	Apply regression techniques to build predictive models and analyze relationships between variables for real-world datasets	A
3	Implement classification algorithms to categorize data and evaluate model performance using appropriate metrics.	C
4	Analyze and apply clustering techniques to group similar data points and interpret patterns in unlabeled datasets.	An

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7

CO1	3	2	1	2	2	1	3
CO2	3	3	2	2	2	1	2
CO3	3	3	2	3	2	1	2
CO4	3	2	2	3	3	1	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	INTRODUCTION TO MACHINE LEARNING		10
	1	What is Machine Learning, Machine Learning Vs. Traditional Programming	
	2	How Machine Learning Works? Applications of Machine Learning	
	3	Types of Learning, Supervised Learning, Unsupervised Learning, Semi-supervised Learning	
	4	Reinforcement Learning, Active Learning, Challenges in Machine Learning	
	5	Machine Learning Pipeline overview	
	6	Introduction to Data, Types of Data - Numerical Data, Categorical Data Data Collection Methods, Introduction to Datasets, Importing Datasets using Python Data Cleaning Basics, Handling Missing Values, Data Visualization Basics Splitting Training and Testing Data	
2	REGRESSION, CLASSIFICATION		12
	1	Introduction to regression, classification, real world use cases	

	2	Types of Regression	
	3	Error and Mean Squared Error for Regression	
	4	Linear Regression, Gradient descent (Concept only), Multiple Linear Regression	
	5	Non-Linear Regression (Polynomial Regression)	
	6	Error in classification, Evaluation Measures : Accuracy, Precision, Recall, F1-Score, Confusion Matrix	
	7	K-Nearest Neighbours (KNN algorithm)	
	8	Logistic Regression	
3	CLASSIFICATION		12
	1	Naïve Bayes Classification, Multinomial Naïve Bayes Classification	
	2	Decision Trees, ID3 algorithm	
	3	Artificial Neural Network, Biological Neuron and Artificial Neuron	
	4	Building Blocks: Input layer, hidden layers, output layer, weights, bias, activation functions: Step, Linear, Sigmoid, Tanh, and ReLU functions, Illustration of Simple Neural Network.	
	5	Single Layer Perceptron	
	6	Multilayer neural network, Loss function	
	7	Importance of Backpropagation (concepts only)	
4	CLUSTER ANALYSIS		11
	1	Introduction, Requirements of Clustering, Categorization of Major Clustering Methods	
	2	Types of Data in Cluster Analysis: Interval-Scaled Variables, Binary Variables, Categorical Variables, Ordinal Variables, Ratio-Scaled Variables, Variables of Mixed Types	
	3	Partitioning Methods: K-means, K-medoids, CLARANS	
	4	Hierarchical Methods : Agglomerative Clustering, Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)	
	5	Density-based Methods: DBSCAN	
5	Teacher Specific Module : Lab experiments		30
	<i>Directions</i>		

	1. To check whether linear relationship exists between two variables, [Refer Textbook 2] a. Stock_Index_Price & Interest_Rate b. Stock_Index_Price & Unemployment_Rate c. Use Pandas for data and Matplotlib for visualization 2. Consider a stock market data and implement multiple linear regression [Refer Textbook 2] a. Use Pandas for data and sklearn and statsmodel for insights 3. Polynomial Regression using Python a. Aim: To implement Polynomial Regression for salary prediction based on experience. Logistic Regression using Python b. Aim : To implement Logistic Regression for predicting student pass/fail status. 4. Implement K-nearest neighbour algorithm and show accuracy on the dataset 5. Use the dataset 'diabetes.csv' and predict whether or not a patient has diabetes based on certain diagnostic measurements given in the dataset. [Refer Textbook 2] 6. For the given dataset [age, income, student, credit_rating, buys_computer] , using Naive Bayes Model predict the class of a new data. [Refer Textbook 2] 7. Using a classifier, classify the 'iris' flowers, use the well-known data set iris.csv [Refer Textbook 2] 8. Build a simple Multi-Layer Perceptron using the library Scikit-learn. Train on small dataset (Iris dataset) 9. Implement the ID3 algorithm using Decision Tree Classifier with entropy. Classify weather conditions for playing outdoor games 10. Implement DBSCAN Clustering to cluster a set of random numbers generated 11. Implement K-Means Clustering to cluster a set of data, a simple customer dataset containing Age & Annual Income	
--	---	--

Essential Readings:

4. Introduction to Machine Learning with Python, Andreas C. Müller & Sarah Guido, O'Reilly
5. Introduction to Machine Learning using Python, Dr. Jeeva Jose, Khanna books
6. Introduction to Machine Learning, Second Edition, Ethem Alpaydın, The MIT Press Cambridge

Suggested Readings:

5. https://python-course.eu/books/bernd_klein_python_and_machine_learning_a4.pdf
6. https://www.w3schools.com/python/python_ml_getting_started.asp
7. <https://nptel.ac.in/courses/106105152>
8. <https://www.eicta.iitk.ac.in/courses/machine-learning-with-python>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

o Employability for the Course:

- o Students completing this course can work as Junior Machine Learning Developer, Python Programmer, AI/ML Intern, Data Science Intern, Research Assistant in AI/ML Projects because the course helps in developing

Industry-Relevant Skills such as Python programming for data analysis, Data preprocessing and visualization, Building regression and classification models, Using machine learning libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn, Problem-solving and analytical thinking

KU6DSECAP308: NATURAL LANGUAGE PROCESSING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP308	4 (3T+1P)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minute

Course Description: This course provides a foundational understanding of Natural Language Processing (NLP), focusing on how computers process, analyze, and understand human language. It covers key concepts such as text preprocessing, feature representation, language modelling, syntax, and knowledge representation. The course also introduces important NLP tasks and applications including text classification, sentiment analysis, information retrieval, machine translation, chatbots, text summarization, and information extraction. Emphasis is placed on rule-based approaches, basic Python implementation, and practical understanding of how NLP techniques are applied in real-world systems.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic concepts of Natural language processing.	U
2	Understand approaches to syntax and semantics in NLP.	U
3	Understand effective knowledge representation ways	U
4	Analyze real-world NLP applications such as text classification, sentiment analysis, chatbots, and information retrieval.	An

5	Understand modern NLP trends including word embeddings, transformers, large language models, and evaluate ethical issues in NLP systems.	U, E
---	--	------

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	1	2	2	1	3
CO2	3	2	2	2	2	1	2
CO3	3	2	2	3	2	1	2
CO4	3	3	2	3	3	2	3
CO5	3	3	2	3	3	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Introduction		10
	1	Introduction to Natural Language Processing	
	2	Corpus, NLP Libraries	
	3	Why is NLP Challenging? Ambiguity, Non-standard English, Segmentation Issues, Idioms / Multiword. Neologisms, World Knowledge,	
	4	NLP phases	
	5	Natural Language Understanding, Natural Language Generation	

2	MODULE TITLE Text Preprocessing and Representation		11
	1	Regular expressions	
	2	Text preprocessing: tokenization, text cleaning, stop-word removal, stemming and lemmatization (concept only, algorithm not required)	
	3	Lexicon	
	4	Morphological Parsing	
	5	POS Tagging (concept and applications)	
	6	Feature representation: Bag of Words, TF-IDF	
3	MODULE TITLE Language Modelling, Syntax, and Knowledge Representation		12
	1	N-grams	
	2	Probabilistic language models, Markov assumption (concept only)	
		Named Entity Recognition (concept)	
	3	Word sense disambiguation (concept)	
	4	Syntactic Analysis: Context-free Grammar (basic idea)	
	5	Knowledge Representation in NLP: case grammar (concept), knowledge graphs, and ontology (overview and applications)	
4	MODULE TITLE Applications and Emerging Trends		
	1	Applications of NLP, Concept, basic working, and use cases of: Text classification, sentiment analysis, spelling correction, chatbots, machine translation, information retrieval, text summarization, and information extraction.	12
	2	Introduction to modern NLP: word embeddings (concept), transformers and large language models (overview only). Use cases	
	3	Ethical issues in NLP (bias, privacy, misinformation)	
5	Teacher Specific Module : Focus on rule-based methods, preprocessing, and basic NLP tasks only. Use NLTK		30
	1. Basics of Python - Variables, data types (list, tuple, dictionary) - Basic input/output		

	• Control structures (if, loops) • Functions 2. Basic Text Processing • Read text input • Convert to lowercase • Remove punctuation and special characters • Split text into words and sentences • Stop-word Removal • Stemming and Lemmatization 3. Word Frequency Analysis • Count frequency of words • Display most frequent words 4. Given a small set of documents, compute the TF-IDF values and identify important words in each document. 5. Regular Expressions : Extract patterns (emails, numbers, dates) from text 6. POS Tagging (Using Library) 7. Named Entity Recognition (Using Library) 8. Sentiment Analysis 9. Text Classification 10. Text Summarization 11. Information Extraction 12. Build a machine learning model that classifies text as positive or negative sentiment 13. To build a machine learning model that classifies emails as Spam or Ham 14. Case study : Complete a case study based on a selected Natural Language Processing application. Should choose one application .The study should include a brief description of the selected application, its objective, real-world use cases, and a simple explanation of how the system works. Students are required to implement a basic working model using Python.	
--	---	--

Essential Readings:

1. Dan Jurafsky and James Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, Second Edition, 2009.
Some draft chapters of the third edition are available online:
<https://web.stanford.edu/~jurafsky/slp3/>
2. James Allen, Natural Language Processing with Python”, First Edition, O’Reilly Media, 2009.
3. Chris Manning and Hinrich Schütze. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA: May 1999.

Suggested Readings:

1. Daniel and James H. Martin Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition”, Second Edition, Prentice Hall of India, 2008.
2. Tanveer Siddiqui, U.S. Tiwary, Natural Language Processing and Information Retrieval”, Oxford University Press, 2008.
3. Steven Bird, Ewan Klein, Edward Loper, Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit (O’Reilly 2009, website 2018) <http://www.nltk.org/book/>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- o **Employability for the Course:** The course in Natural Language Processing equips students with skills in text processing, sentiment analysis, chatbots, language models, and AI-based application development. Students can pursue careers as NLP Engineers, AI Developers, Data Scientists, Chatbot Developers, and Machine Learning Engineers in industries such as healthcare, banking, e-commerce, education, and social media analytics.

Application Development

KU5DSECAP309 : FRONTEND DEVELOPMENT USING REACT

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
V	DSE	300-399	KU5SDECAP309		4(3+1)	75
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: The course Frontend Development Using React introduces students to modern web application development using the React JavaScript library. The course covers the fundamentals of component-based UI design, JSX, props, state management, event handling, routing, and API integration for building interactive and responsive web applications. Students will also learn React hooks, form handling, and basic frontend project deployment techniques. The course emphasizes practical implementation and hands-on development of dynamic single-page applications following modern frontend development practices

Course Prerequisite: Programming concepts

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of ReactJS and its architecture.	U
2	Develop functional components and manage application states effectively.	A
3	Use React Router for navigation and single-page application development.	A,An
4	Integrate APIs and perform data fetching using Axios/Fetch	A,C

5	Build and deploy a complete ReactJS application.	C
---	--	---

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	3	1	2	1	1	3
CO2	3	3	2	2	1	1	3
CO3	3	3	1	2	1	2	2
CO4	3	3	3	2	2	1	2
CO5	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOUR S
1	Introduction to ReactJS		11
	1	Overview of front-end frameworks	
	2	Introduction to ReactJS, JSX, Virtual DOM	
	3	Installation and Project Setup	
	4	Components and Props	
2	State Management and Lifecycle		10
	1	Introduction to State and Lifecycle	
	2	Handling Events and Conditional Rendering	

	3	Handling Lists and Keys	
	4	Forms and Controlled Components	
	5	Lifecycle Methods	
3	React Hooks and Advanced Concepts		12
	1	Understanding Hooks – useState, useEffect	
	2	Custom Hooks	
	3	Context API and useContext Hook	
	4	Error Boundaries and Code Splitting	
4	Routing and Data Handling		12
	1	Introduction to React Router	
	2	Navigation and Parameters	
	3	Fetching Data from APIs using Fetch and Axios	
	4	Managing Async Operations	
5	Teacher Specific Module : REcommended Lab experiments		30
	<i>Directions:</i>		
	1. Install Node.js and create a React app using Create React App or Vite. Display: Student Name , Course ,College Name 2. Create: Header , Footer ,StudentCard Pass student details using props. 3. Create a counter with: Increment , Decrement, Reset. Use useState. 4. Create a Login/Logout interface. Display different messages based on login status. 5. Display a list of products/students dynamically using arrays and map(). Add unique keys. 6. Create a Student Registration form with: Name ,Email ,Course ,Gender .Use controlled components. Validate empty fields. 7. React Hooks Demonstration: Create a program using: useState , useEffect . Display current time updating every second. 8. Navigation using React Router: Create pages: Home , About , Contact, Implement navigation using React Router.		

Essential Readings:

1. Stoyan Stefanov, *React Up and Running*, O'Reilly Media.
2. Alex Banks & Eve Porcello, *Learning React*, O'Reilly.
3. Jonathan Sanderson, *Full-Stack React Projects*, Packt.

4. Dave Ceddia, *Pure React*, Independently Published.
5. Mark Tielens Thomas, *React in Action*, Manning.

Suggested Readings:

1. Kirupa Chinnathambi, *Learning ReactJS: A Hands-On Guide*.
2. Robin Wieruch, *The Road to React*.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

Employability for the Course: React Developer (Frontend) , UI Engineer , Web Application Developer , Full Stack Developer (MERN) , Freelance Web Developer

KU5DSECAP310 : DATABASE MANAGEMENT FOR APPLICATIONS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP310	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	—	35	65	100	1 Hr 30 Minutes

Course Description: The course provides practical knowledge of modern database systems and their use in software applications. The course covers database servers, database configuration, administration, and management using technologies such as MySQL, SQL Server, and NoSQL databases. Students will learn database operations including DDL commands, transactions, backup and recovery, user management, and performance optimization. The course also introduces NoSQL databases and modern data storage techniques used in real-world applications. In addition, students will gain hands-on experience in integrating databases with web and mobile applications for developing secure and efficient data-driven systems.

Course Prerequisite: Database Management System

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand and configure database servers including MySQL, SQL Server, and NoSQL databases for application development environments.	U
2	Perform database administration tasks such as database creation, DDL operations, user management, backup, recovery, and performance optimization.	A
3	Develop and manage relational and NoSQL databases using SQL queries, transactions, indexing, and document-based data storage techniques.	An, C

4	Integrate databases with web and mobile applications to implement secure and efficient data-driven applications using modern development tools and frameworks.	A, C
---	---	------

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	3	2	1	1	2
CO2	3	2	3	3	1	1	2
CO3	3	3	3	2	2	1	2
CO4	3	3	3	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1		DATABASE SERVERS AND SYSTEM CONFIGURATION	11
	1	Introduction to Database Servers, Database Server Services and Management	
	2	Installation and Configuration of MySQL Server Installation and Configuration of Microsoft SQL Server Introduction to PostgreSQL Server Installation and Basic Setup of MongoDB	
	3	Configuring Users, Roles and Privileges Database Authentication Methods	

	4	Importing and Exporting Databases Backup and Restore Operations	
	5	Introduction to Database Administration Tools	
2	ADVANCED DATABASE OPERATIONS AND ADMINISTRATION		12
	1	Managing Tables and Schemas in Different Databases, Advanced DDL Operations in MySQL and SQL Server	
	2	Index Creation and Management, Stored Procedures and Triggers	
	3	Database Replication Basics, Database Migration Techniques	
	4	Handling Large Databases, Query Optimization Techniques	
	5	Command Line and GUI Administration Tools	
3	NOSQL DATABASES AND MODERN DATA MANAGEMENT		10
	1	Introduction to NoSQL Databases Installation and Configuration of MongoDB Working with Collections and Documents	
	2	CRUD Operations in MongoDB JSON-Based Data Storage Querying and Filtering Documents	
	3	Indexing and Aggregation in MongoDB Data Replication and Sharding Concepts Using Firebase Realtime Databases	
	4	Cloud Database Storage Basics Comparing RDBMS and NoSQL in Real Applications	
	5	Data Synchronization Techniques Managing Semi-Structured and Unstructured Data	
4	DATABASE INTEGRATION FOR WEB AND MOBILE APPLICATIONS		12
	1	Connecting MySQL with PHP Applications Database Connectivity using Python Applications Using JDBC for Java Database Connectivity	
	2	CRUD Operations in Web Applications Form Handling and Dynamic Data Storage Session Management with Databases(using Java/Python) Authentication and User Login Systems	

	3	Connecting Databases with Android Applications Using SQLite in Mobile Applications Firebase Integration in Mobile Apps	
	Teacher Specific Module Lab experiments		30
	<i>Recommended Lab Experiments</i>		
5	1. Install and configure the MySQL database server and perform basic database operations after creating a user and a database. Configure Users, Roles, and Privileges 2. Install MongoDB and perform basic NoSQL database operations, by opening MongoDB Compass or Mongo Shell. Create a database and collection. 3. Perform database backup, export, import, and restore operations. a. Export the database using mysqldump. b. Delete the database. c. Restore the database backup. 4. Perform advanced DDL operations for managing tables and schemas in MySQL or SQL Server. 5. Improve database performance using indexes and automate operations using stored procedures and triggers. a. Create indexes on frequently searched columns. b. Create and execute a stored procedure. 6. Export a database from MySQL. Import the database into another server/system. Perform simple migration using dump files. 7. Perform CRUD operations and basic querying in MongoDB. a. Insert documents into collections. b. Update and delete documents. c. Query documents using conditions. d. Create indexes. e. Perform aggregation operations. 8. Create a Firebase project, Enable Realtime Database, Add sample JSON data, & Read and update data from Firebase Console. 9. Connect a PHP application with MySQL database and perform database operations. 10. Connect Python applications with MySQL databases. 11. Perform CRUD operations using JDBC in Java applications. 12. Create login forms with database authentication and session management. a. Create a login form. b. Store user details in the database c. Validate login credentials.		

	d. Create sessions after successful login 13. Connect Android applications with SQLite and Firebase databases. a. Create a simple Android project. b. Implement SQLite database. c. Insert and retrieve records. d. Connect Firebase with Android app e. Store and retrieve cloud data. 14. Create JSON documents with nested data, Import JSON files into MongoDB, Modify documents and synchronize updates.	
--	---	--

Essential Readings:

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan. Publisher McGraw-Hill Education
2. Database Management Systems by V. Vidhya, G. Jeyaram, and K. R. Ishwarya
3. Big Data and NoSQL , Seema Acharya, Subhasini Chellappan, Publisher Wiley India Pvt. Ltd.
4. <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17>
5. PHP and MySQL Web Development, Luke Welling, Laura Thomson, Publisher Addison-Wesley
6. <https://dev.mysql.com/doc/workbench/en/>

Suggested Readings:

1. Fundamentals of Database Systems by Ramez Elmasri and Shamkant B. Navathe, Pearson
2. <https://www.mongodb.com/docs/manual/tutorial>
3. <https://www.w3schools.com/mongodb/index.php>
4. <https://firebase.google.com/docs>
5. <https://developer.android.com/training/data-storage/sqlite>
6. <https://dev.mysql.com/doc/>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU6DSECAP311:BACKEND DEVELOPMENT & FULL STACK INTEGRATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP311	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course provides a comprehensive introduction to full-stack web development using Node.js, Express.js, and React integration, along with fundamental concepts of backend development, REST APIs, and deployment. Students will learn how to build server-side applications using Node.js, create and manage APIs using Express.js, and connect frontend applications with backend services.

Course Prerequisite: Web Technology, Frontend Development using React, Database Management for Applications

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of Node.js	U
2	Develop backend applications using Express.js framework.	A C
3	Build and test RESTful APIs using CRUD operations.	C
4	Connect React frontend applications with Node.js backend APIs.	A
5	Deploy frontend and backend applications	A

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	3	1	2	1	1	3
CO2	3	3	2	2	1	1	3
CO3	3	3	3	2	2	1	2
CO4	3	3	3	2	2	2	2
CO5	3	3	2	3	1	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Node.js Fundamentals		11
	1	Introduction to Node.js and runtime environment	
	2	Single-threaded and non-blocking I/O (concept only)	
	3	Global objects in Node.js	
	4	Core modules overview	
	5	Built-in Modules: File System module (fs), Path module, OS module, HTTP module (basic introduction)	
	6	Modules and exports	
	7	Introduction to npm, package.json (purpose and structure), Installing and managing dependencies	
2	MODULE TITLE : Express.js & REST API Development		12
	1	Introduction to Express.js	
	2	Application Setup	
	3	Basic routing (GET, POST)	

	4	Request and Response : Understanding req and res, Sending response using res.send(), Sending JSON response	
	5	Route Parameters & Query Parameters, Route parameters (/user/:id), Query parameters (?name=value)	
	6	Middleware : Simple built-in middleware (express.json()), Request logging (basic idea)	
	7	REST API Basics	
	8	CRUD Operations (Simple Level) :Create (POST), Read (GET), Update (PUT - concept only), Delete (DELETE - concept only)	
	9	Status Codes (Basic)	
	10	API Testing (Introduction Only) : Introduction to Postman, Testing GET and POST APIs	
3	MODULE TITLE : Frontend & Backend Integration		12
	1	Overview of React frontend and Node.js backend, Client–server architecture, JSON data exchange	
	2	Connecting React with Backend APIs : API integration?, Backend API endpoints overview, Using backend URLs in React	
	3	CORS Configuration : What is CORS? Enabling CORS in Express (cors() middleware), Allowing React frontend access	
	4	API Consumption in React : Using fetch() to call backend APIs, Handling JSON response, Displaying API data in UI	
4	MODULE TITLE Introduction to Deployment & Hosting		10
	1	Introduction to Deployment : Difference between local and live applications	
	2	Overview of hosting platforms	
	3	Frontend Deployment Basics : Deploying frontend using Vercel	
	4	Deploy Node.js backend using Render	
	5	Connect frontend and backend using live URLs	
	6	AWS Basics: Introduction to AWS services	
5	Teacher Specific Module : For practical courses add lab experiments		30
	<i>Directions</i>		
	1. Install and configure Node.js environment.		

	2. Understand module creation and exports in Node.js. 3. Perform file operations using Node.js FS module. 4. Create a simple web server using Node.js. 5. Install and use external packages using npm. 6. Create a simple REST API for user data using Express.js. 7. Create a REST API for managing product data. 8. Implement basic CRUD operations using Express.js. 9. Implement middleware in Express.js. 10. Create simple frontend–backend integration project 11. Case study : To design and develop a full stack web application of a student's choice using frontend, backend, and database integration and deploy.	
--	--	--

Essential Readings:

1. A A Puntambekar, Full Stack Web Development, Technical Publications, First Edition,
2. Philip Ackermann, Full Stack Web Development The Comprehensive Guide, Rheinwerk, Publishing Inc, First Edition, 2023.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- **Employability for the Course:** Full Stack Developer

KU6DSECAP312: MOBILE APPLICATION DEVELOPMENT USING FLUTTER

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP312	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces students to cross-platform mobile application development using Flutter and Dart. The course covers Dart programming fundamentals, Flutter widget-based UI design, state management, navigation, local data storage, and API integration. Students learn to design responsive user interfaces, manage application state, handle user interactions, and develop data-driven mobile applications. Through hands-on laboratory exercises and mini projects, students gain practical experience in building functional Android/iOS applications using modern mobile development practices.

Course Prerequisite: Basic knowledge of computer programming concepts

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the core concepts of Dart required for Flutter development.	U
2	Learn Flutter architecture and build basic UI using widgets.	U, A
3	Handle user interaction, dynamic UI updates, and navigation.	A
4	Integrate APIs, process JSON data, and implement local data persistence using SharedPreferences and SQLite with sqflite.	A
5	Design and develop mobile applications	A

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)***

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	3	1	2	1	1	3
CO2	3	3	1	2	1	1	3
CO3	3	3	1	2	1	2	2
CO4	3	3	3	2	2	2	2
CO5	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Dart Programming Fundamentals		11
	1	Introduction to Dart: Overview of Dart, Installation & setup, Writing first program (Hello World).	
	2	Data Types & Variables: var, final, const, int, double, String, bool.	
	3	Collections: List, Map, Set.	
	4	Control Flow: if-else, loops, Functions, parameters, arrow functions.	
	5	Object-Oriented Programming: Classes, objects, constructors.	
	6	Null Safety & Error Handling: Null safety (?, !, late), try-catch.	
	7	Asynchronous Programming: Future, async/await.	
2	MODULE TITLE : Flutter Fundamentals & UI Basics		12
	1	Introduction to Flutter: Flutter SDK setup, Project structure, Hot reload.	
	2	Widget System: Everything is a widget, Widget tree concept, Stateless vs stateful widgets.	

	3	Layout & Core Widgets: Row, Column, Container, Padding, Expanded, Scaffold, AppBar, Text, Image, Buttons.	
	4	Design Systems: Material Design vs Cupertino, Basic theming.	
	MODULE TITLE : State Management & User Interaction		12
3	1	Stateful Widgets & Lifecycle: setState(), and other lifecycle methods, Basic lifecycle concepts.	
	2	User Input & Forms: TextField and controllers, Input validation.	
	3	Lists & Dynamic UI: ListView.builder, Displaying dynamic data.	
	4	Navigation: Navigator.push() and pop(), Passing data between screens.	
	5	Introduction to State Management: Limitations of setState, Basic concept of Provider.	
	MODULE TITLE Application Development & Data Management		10
4	1	API Integration & JSON Handling: HTTP requests, JSON parsing, Loading and error handling.	
	2	Local Storage: SharedPreferences (key-value storage). Introduction to databases. SQLite using sqflite (Database creation, CRUD operations).	
	3	Code Organization: Folder structure, Reusable widgets, Basic separation of concerns	
	4	Data Passing Between Screens: Navigator arguments.	
	Teacher Specific Module :		30
	<i>Directions</i>		
5	1) Lab set up 2) Student Record Manager (Console Application) Add, update, delete student records. Use List/Map, functions, loops. Implement error handling and async delay. 3) Personal Profile App Create a profile UI with image, text, and buttons. Layout widgets and styling. 4) To-Do List Application Add, delete, and update tasks. Use ListView, TextField, navigation. Manage state using setState. 5) Notes / Task Manager App with Database Add, edit, delete tasks. Store data using SQLite.		

	Display persistent data. ○ Use navigation and structured UI.	
	6) Case study : Study the design and development of a real-world mobile application using Flutter and SQLite.	

Essential Readings:

1. Dart Official Documentation: <https://dart.dev/guides>. (Module 1)
2. Raywenderlich Tutorial Team, Jonathan Sande,, Matt Galloway, Dart Apprentice (First Edition): Beginning Programming with Dart (Module 1)
3. Flutter Official Documentation: <https://docs.flutter.dev>. (Module 2)
4. Vincent Ngo, Stef Patterson, Alejandro Ulate Fallas & Kevin D Moore, Flutter Apprentice Feb 21 2024 , Dart 3.2.6, Flutter 3.16.9 (Module 2)
5. Material Design: <https://material.io>. (Module 2)
6. Flutter state management doc:
<https://docs.flutter.dev/data-and-backend/state-mgmt>. (Module 3)
7. Provider Documentation: <https://pub.dev/packages/provider>. (Module 3)
8. http package: <https://pub.dev/packages/http>. (Module 4)
9. SharedPreferences: https://pub.dev/packages/shared_preferences. (Module 4)
10. sqflite: <https://pub.dev/packages/sqflite>. (Module 4)

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10

a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- o **Employability for the Course:** Mobile app startups, App development

Cyber Security & Cloud Computing

KU5DSECAP313: INTRODUCTION TO CYBER SECURITY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP313	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1Hr 30 Minutes

Course Description: This course introduces the fundamentals of cybersecurity, cyber threats, secure computing, and digital safety. Students learn basic concepts of network, web, mobile, and system security along with cyber laws, ethics, and emerging technologies. The course includes practical activities such as Linux basics, phishing identification, password security analysis, secure browsing, and introductory cybersecurity challenges to develop foundational cybersecurity awareness and skills.

Course Prerequisite: Basic knowledge of computers, internet and digital communication

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand basic cybersecurity concepts, threats and defense mechanisms.	U
2	Identify common cyber attacks including phishing, malware and web attacks.	An
3	Demonstrate basic Linux and network security operations.	A
4	Understand privacy, mobile and social media security risks.	U
5	Explain cyber laws, ethics, digital forensics and emerging cybersecurity trends.	An

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	3	1	2	3
CO2	3	2	2	3	1	2	3
CO3	3	2	3	2	1	1	2
CO4	3	1	2	3	1	3	2
CO5	3	1	1	3	2	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	H O U R S
1	MODULE TITLE: Cybersecurity Fundamentals		11
	1	Introduction to Cybersecurity Definition of Cybersecurity, Importance of Cybersecurity, Evolution of Cyber Threats, Cybersecurity Terminologies, Understanding Cyberspace, World Wide Web (Internet), Critical Information Infrastructure, Internet of Things (IoT) Overview	
	2	CIA Triad and AAA Confidentiality, Integrity, Availability, Authentication, Authorization, Accounting, Multi-factor Authentication, Role-Based Access Control (RBAC), Zero Trust Policy	
	3	Cyber Threats and Malware Types of Cyber Threats, Malware Fundamentals, Viruses, Worms and Trojans, Ransomware and Spyware, Botnets, DoS and DDoS Attacks, Malware Infection Methods, Real-world Cyber Incidents	
	4	Phishing, Vulnerabilities and Risk Management Phishing and Social Engineering, Spear Phishing and Vishing, Vulnerabilities and Exploits, Zero-day Vulnerabilities, Risk, Threat and Attack Surface, Security Domains Overview	
	5	Practical: Identifying phishing emails and Password strength analysis	
2	MODULE TITLE: Operating Systems and Network Security Basics		10

	1	Operating Systems and Linux Basics Introduction to Linux OS, GUI vs CLI, User Accounts and Privileges, Administrative Access, Virtualization Basics, Linux Navigation Commands, File Management Commands	
	2	File Systems and Permissions File System Basics, Linux Directory Structure, File Permissions and Ownership, Read, Write and Execute Permissions, Permission Management Commands, Hidden Files and File Integrity Basics	
	3	HTTP and HTTPS, Ports and Protocols, VPN and Secure Browsing IPv4 Addressing, Public vs Private IP, DNS Fundamentals, Common Ports and Services, HTTP and HTTPS, SSL/TLS Basics, VPN Fundamentals, Browser Security and Secure Browsing Practices	
	4	Practical: Linux terminal basics, Checking IP address and ports, Using ping, traceroute and nslookup	
3	MODULE TITLE: Web, Mobile and Social Media Security		12
	1	Basics of Web Applications and Web Threats HTTP Request and Response, APIs Overview, Sessions and Cookies, OWASP Overview, Injection Attacks, Broken Authentication, Security Misconfigurations	
	2	SQL Injection and Cross Site Scripting SQL Injection Basics, Types of SQL Injection, Impact of SQL Injection, Cross Site Scripting (XSS) Basics, Types of XSS, Input Validation and Prevention Concepts	
	3	Social Media, Privacy and Digital Footprint Social Media Threats, Fake Profiles and Identity Theft, Oversharing Risks, Digital Footprint, Metadata and Tracking, Browser Fingerprinting, Privacy Protection Basics	
	4	Mobile and Communication Security Mobile Security Basics, App Permissions and Risks, Public Wi-Fi Risks, Rogue Access Points, Secure Messaging, End-to-End Encryption, Secure Digital Communication Practices	
	5	Practical: Browser developer tools introduction, Cookie and session demonstration, Privacy settings analysis and Password manager usage	
4	MODULE TITLE: Cyber Law, Ethics, Career Opportunities and New Technologies		12
	1	Cyber Ethics and Cyber Laws Cyber Ethics Fundamentals, Responsible Use of Technology, Ethical Hacking Concepts, Cyber Laws in India, Cybercrime Categories, Information Technology Act 2000, Cyber Offenses and Penalties	

	2	Responsible Disclosure and Digital Rights Responsible Disclosure Process, Bug Bounty Programs, Legal Considerations in Security Research, Digital Certificates, Digital Signatures, Copyrights and Software Licensing	
	3	Cyber Forensics and Blockchain Basics of Digital Forensics, Digital Evidence Handling, Chain of Custody, Disk and Memory Forensics Basics, Introduction to Blockchain, Cryptocurrency and Decentralization	
	4	Careers, Certifications and Emerging Trends Cybersecurity Career Domains, SOC Analyst and Penetration Testing Roles, Cybersecurity Certifications, Capture The Flag (CTF) Basics, AI in Cybersecurity, Cloud Security Overview, Emerging Trends in Cybersecurity	
	5	Practical: Introduction to cybersecurity competitions (CTF)	
5	Teacher Specific Module : Cybersecurity Teaching Methodology and Practical Exposure		30
	Cybersecurity Lab Setup Setting up a basic cybersecurity lab, Linux virtual machines, VirtualBox basics		
	Practical Sessions in Fundamental Security Identifying phishing emails, Password strength analysis, Multi-factor authentication demonstration		
	Practical Sessions in Networking and System Security Linux terminal basics, File permission management, Checking IP addresses and ports, Using ping, traceroute and nslookup		
	Practical Sessions in Web, Privacy and CTF Learning Browser developer tools introduction, Cookie and session demonstration, Privacy settings analysis, Password manager setup and usage, Introduction to CTF platforms, Beginner cybersecurity challenge walkthroughs, Conducting mini cybersecurity competitions and assessments		

Essential Readings:

1. Cryptography and Network Security-Principles and Practice, 8th Edition, by Stallings Agenda Item 65/39 - Annexure - 35 William, published by Pearson, 2020
2. Pfleeger, C.P., Security in Computing 5th Edition, Prentice Hall, Copyright 2010 ISBN 0-13-239077-9. Schneier, Bruce. Applied Cryptography, Second Edition, John Wiley & Sons, 1996.
3. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley

Suggested Readings:

1. William Stallings, Lawrie Brown(2016), Computer Security: Principles and Practices, Second Edition, Pearson Publication.
2. Mike Shema , Anti-Hacker Tool Kit, Mc Graw Hill

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU5DSECAP314: CRYPTOGRAPHY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP314	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2

Course Description: This course introduces the fundamental concepts and techniques of cryptography and network security. It covers classical and modern encryption methods, symmetric and public key cryptosystems, message authentication, digital signatures, hash functions, and key management techniques.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the principles of information security, cryptography, and cryptanalysis.	U
2	Illustrate symmetric, asymmetric key cryptosystems for secure communication.	An
3	Study hash functions, digital signatures, and message authentication mechanisms.	An
4	Interpret public, secret key distribution techniques for secure communication.	An
5	Develop the foundation required for advanced studies in cyber security and secure systems	A C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	3	1	2	3
CO2	3	2	3	3	1	2	2
CO3	3	2	3	3	1	2	2
CO4	3	2	3	3	1	2	2
CO5	3	2	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE Fundamentals of Cryptography and Classical Ciphers		12
	1	Security attacks, Services, Mechanisms. Steganography, Cryptography vs Cryptanalysis.	
	2	Classical encryption techniques – Symmetric cipher model.	
	3	Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers,	
	4	Caesar cipher, Affine cipher, Playfair cipher, Vigenere cipher, Hill cipher	
		Transposition ciphers – Keyless, Keyed, Double transposition.	
		Kirchhoff's principle, Introduction to cryptanalysis techniques	
2	MODULE TITLE (Modern Symmetric Key Cryptosystems)		12
	1	Symmetric key ciphers – Block vs Stream ciphers	
		Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers.	
	2	Data Encryption Standard (DES) – Structure, Key generation, Design criteria,	

	3	Weaknesses, Double DES, Triple DES	
		Security of DES - brute-force attack, differential cryptanalysis, linear cryptanalysis.	
	5	Stream ciphers – Structure, RC4.	
3	MODULE TITLE(Public Key Cryptosystems)		12
	1	Introduction to public key cryptosystems – Principles, Applications, Requirements, Conventional vs Public key cryptosystems.	
	2	RSA cryptosystem – Algorithm, Security, Attacks.	
		Public Key Cryptanalysis.	
4	MODULE TITLE (Message Integrity and Authentication, Key management)		12
	1	Hash functions – Security requirements, Secure Hash Algorithm (SHA-512).	
	2	Digital signatures – Attacks, Forgeries, Requirements, RSA digital signature	
	3	Key management – Distribution of secret keys using symmetric and asymmetric encryption, Distribution of public keys.	
		Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack.	
5	Teacher Specific Module :		12
	<i>Directions</i>		
	Virtual Labs. Use virtual experiments to understand the basic mathematical foundations of cryptography, to gain insightful experience by working with fundamental cryptographic applications and to train in the art of design and analysis of information security protocols. https://www.opencompsci.com/books/security-lab-manual		

Essential Readings:

1. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Pearson Ed.
2. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.

Online Courses

1. NPTEL Foundations of Cryptography, By Dr. Ashish Choudhury, IIIT Bangalore, <https://nptel.ac.in/courses/106106221>.
2. Cryptography, by Prof Don Boneh, Stanford University <https://www.coursera.org/learn/crypto>.

Web resources

1. [Welcome to Virtual Labs](#).
2. <https://www.opencompsci.com/books/security-lab-manual>

Suggested Readings:

1. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
2. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007.
3. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall/CRC publication, 2nd edition, 2014.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	10
Total		100

KU6DSECAP315: ETHICAL HACKING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP315	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: Ethical hacking is a subject that has become very important in present-day context, and can help individuals and organizations to adopt safe practices and usage of their IT infrastructure. The course will cover various attacks and vulnerabilities and ways to secure them. There will be hands-on demonstrations that will be helpful to the participants.

Course Prerequisite: Basic concepts in programming and networking, Knowledge in Cryptography

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand ethical hacking concepts, cyber kill chain, and penetration testing methodologies	R, U
2	Perform basic reconnaissance, scanning, and enumeration techniques	U, A
3	Conduct beginner-level vulnerability assessment and web application security testing	A, An
4	Apply wireless and defensive security practices using standard cybersecurity tools	A, An
5	Prepare basic penetration testing reports and demonstrate responsible security practices	U, A, C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	3	1	2	3
CO2	3	2	3	2	1	1	2
CO3	3	2	3	3	2	2	2
CO4	3	2	3	3	1	2	3
CO5	3	1	2	3	1	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Ethical Hacking Fundamentals and Reconnaissance		11
	1	Introduction to Ethical Hacking and Phases of Hacking	
	2	Types of Hackers and Cyber Kill Chain	
	3	Penetration Testing Methodology	
	4	Footprinting, Reconnaissance and OSINT	
		Practical: Kali Linux Installation, VM Setup, Metasploitable 2 Setup, Basic OSINT Exercises	
2	Network Scanning and Vulnerability Assessment		10
	1	Network Scanning Concepts	
	2	Port Scanning and Service Enumeration	
	3	Vulnerability Assessment	
	4	Introduction to Exploitation Concepts	

		Practical: Host Discovery, Nmap Scanning, Enumeration and Vulnerability Scanning	
3	Web Application Security and Vulnerability Testing		12
	1	Web Application Architecture	
	2	HTTP Request and Response	
	3	OWASP Top 10 Overview	
	4	Input Validation Vulnerabilities	
		Practical: HTTP Interception, SQL Injection Testing, XSS Testing and Directory Enumeration	
4	Wireless Security Assessment and Defensive Security		12
	1	Wireless Security Assessment	
	2	Password Attack Concepts	
	3	Reporting and Documentation	
	4	Introduction to Defensive Security	
		Practical: Password Auditing, Wireshark Traffic Analysis and Penetration Testing Report Preparation	
5	Teacher Specific Module :		30
	<i>Directions</i>		
	1) Kali Linux Installation, VM Setup, Metasploitable 2 Setup, Basic OSINT Exercises 2) Host Discovery, Nmap Scanning, Enumeration and Vulnerability Scanning 3) HTTP Interception, SQL Injection Testing, XSS Testing and Directory Enumeration 4) Password Auditing, Wireshark Traffic Analysis and Penetration Testing Report Preparation		

Essential Readings:

1. Michael T Simpson, Kent Back man, James Corley, “Hands on ethical hacking and network defense”, Cengage Learning, 2nd edition, 2010.
2. Ed Skoudis and Tom Liston, “Counter Hack Reloaded: A step-by-step guide to computer attacks and effective defenses”, Prentice Hall Series in Computer Networking and security, 2nd edition, 2006. 3

3. Rafay Baloch, “Ethical Hacking and Penetration Testing Guide”, CRC Press, 2014.
4. Host Discovery, Nmap Scanning, Enumeration and Vulnerability Scanning

Suggested Readings:

1. Kimberly Graves, “Certified Ethical Hacker: A Study Guide”, Wiley Publishing, Inc., 2010.
2. Stuart Mc Clure, Joel Scambray, “Hacking Exposed Network Security Secrets & Solutions”, McGraw-Hill publishing, edition 7, 2012

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar/Case Study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- **Employability for the Course:** Ethical hacking skills create career opportunities in cybersecurity roles such as penetration tester, security analyst, vulnerability assessor, and cybersecurity consultant. Knowledge of ethical hacking helps students work in securing networks, web applications, cloud systems, and digital infrastructures in IT industries and organizations.

KU6DSECAP316:CLOUD SECURITY AND INTRODUCTION TO DIGITAL FORENSIC

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP316	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	-	30	70	100	2

Course Description: The course provides basic understanding of the principles and challenges of cloud computing security, including risk assessment, legal considerations and the Key strategies and best practices for securing the cloud environment. It provides an overview of the strategies in securing cloud environments using Amazon Web Services (AWS). The course also provides fundamental knowledge in digital forensics principles, evidence analysis techniques and basic skills for applying forensic methods in real-world investigations.

Course Prerequisite: Fundamentals of Cloud Computing

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand security concerns, risks, and legal aspects of cloud computing, focusing on virtualization, provisioning, storage, and networking, to identify and mitigate vulnerabilities.	U
2	Familiarise cloud security architectures and strategies, including encryption methods, sensitive data categorization, and solutions for vendor lock-in, to ensure secure data storage and compliance.	A
3	Apply advanced monitoring and auditing practices using AWS tools like CloudWatch, CloudTrail, and AWS Config to gain actionable insights into cloud operations and enhance compliance.	A
4	Understand the basics of digital forensic, systematic investigation processes, including evidence handling, data acquisition, and advanced forensic analysis techniques.	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	3	3	1	2	3
CO2	3	2	3	3	2	2	3
CO3	3	2	3	3	2	2	3
CO4	3	1	2	3	2	3	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Fundamentals of Cloud Computing and Cloud Security		12
	1	Introduction to Cloud Computing Cloud Service Models (IaaS, PaaS, SaaS) Deployment Models (Public, Private, Hybrid Cloud)	
	2	Virtualization Concepts and Virtual Machines Cloud Security Challenges and Threats	
	3	Authentication and Access Control Password Security and Multi-Factor Authentication Secure Cloud Storage and File Sharing	
	4	Overview of Cloud Security- Security services- Confidentiality, Integrity, Authentication, Non repudiation, Access control	
2	Network and Infrastructure Security in Cloud		12
	1	Firewalls and Intrusion Detection Concepts Network Monitoring and Traffic Analysis	
	2	Introduction to Wireshark and Packet Analysis Cloud Infrastructure Vulnerabilities	
	3	Risk Assessment and Security Best Practices Introduction to Security Auditing	

	4	User authentication methods: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Multi-Factor Authentication (MFA), Single Sign-On (SSO), Privileged access management	
	5	Secure Cloud Architecture : Secure deployment models, Secure APIs and interfaces. Secure workload management, Cloud workload isolation. Secure software development lifecycle in cloud	
3	Cloud Security Design patterns		12
	1	Storage Security: Secure cloud storage architecture, Data redundancy and replication, Storage access controls, Secure backup systems, Storage auditing	
	2	Data Protection Strategies : Encryption techniques, Data masking, Data loss prevention (DLP), Secure deletion methods, Disaster recovery planning	
	3	Introduction to Cloud Bursting: Definition and concept, Hybrid cloud environments, Public and private cloud integration Working of Cloud Bursting: Dynamic workload distribution, Resource scaling mechanisms, Traffic overflow handling Security Challenges in Cloud Bursting: Data security during migration, Secure workload balancing, Network security concerns, Identity management across clouds	
	4	Geo-Tagging in Cloud Environments: Location-aware cloud applications, Cloud-based mapping systems, Mobile and IoT integration	
4	FOUNDATIONS OF DIGITAL FORENSICS		12
	1	Introduction to Cyber Forensics Cyber Crimes and Digital Evidence Need and Importance of Digital Forensics	
	2	Phases/Process of Cyber Forensics Investigation Evidence Collection and Preservation Digital Evidence and Handling Chain of Custody	
	3	Reporting and Documentation in Forensics, Reporting and Communication	
	4	Mobile Forensics Fundamentals Network Forensics Basics Email and Web Forensics File System and Log Analysis	
	5	Legal and Ethical Considerations, Ethical Issues in Digital Investigation, Evidence Admissibility Frameworks	

	6	Introduction to Forensic Tools	
	Teacher Specific Module : Activities		12
	<i>Directions</i>		
5	• Activity 1 : Create a free cloud storage account (Google Drive, OneDrive, Dropbox, etc.) and apply security settings like Enable two-factor authentication, Set strong passwords, Share files with permission settings and Identify public vs private sharing risks • Activity 2: Analyze simple real-life cloud security incidents. Identify possible threats such as weak passwords, phishing, data leakage, Suggest preventive measures, Present the findings • Activity 3: Evaluate password strength using password-checking tools. Create weak and strong passwords, Test complexity and cracking difficulty, Present the password security policies • Activity 4: Prepare a short forensic investigation report based on a given scenario. Document observations, Include evidence summary and present the conclusions		

Essential Readings:

1. J.R. Winkler, Securing the Cloud: Cloud Computer Security Techniques and Tactics, 1st ed., Elsevier, 2011.
2. Cloud Security: Attacks, Techniques, Tools and Challenges by Preeti Mishra (Author), Emmanuel S Pilli (Author), R. C. Joshi (Author), CRC Press
3. Cloud Security for Beginners: Exploring the foundations by Sasa Kovacevic, bpb
4. Cloud Computing: Concepts and Practice - A Comprehensive Guide by Pencil Bitz Paperback ,Altaic Languages Edition, by Atma Prakash Singh (Author), E Jansirani (Author), Dr Nanda Kishor S (Author), Isha Das (Author)
5. B. Nelson et al., Guide to Computer Forensics and Investigations, Sixth Edition, 2020.
6. Practical Cloud Security Handbook: Secure cloud deployments with AWS, Azure, GCP, and IBM Cloud by Shiv Kumar, bpb
7. Nour Moustafa, Digital Forensics in the Era of Artificial Intelligence, Taylor & Francis; 1st edition, 2024
8. J. Sammons, The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, Elsevier, 2014.

9. Security and Data Storage Aspect in Cloud Computing: by Prachi S. Deshpande (Author), Subhash C. Sharma (Author),
10. T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy, 1st ed., O'Reilly, 2009.

Suggested Readings:

1. T. J. Holt et al., Cybercrime and Digital Forensics: An Introduction, Routledge, 2nd Edition, 2017.
2. S. Widup and J. Sammons, Computer Forensics and Digital Investigation with EnCase Forensic, Syngress
3. Cyber Security, Nina Godbole, Sunit Belapure
4. R. Vacca, Cloud Computing Security Foundations and Challenges, CRC Press, 2017.
5. Ronald L. Krutz and Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley, 2010.

Online Resources:

1. AWS Security : <https://docs.aws.amazon.com/security/>
2. AWS Monitoring and Observability : <https://docs.aws.amazon.com/monitoring/>
3. AWS Compliance and Legal Resources : <https://aws.amazon.com/compliance/>

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	5
d)	Activities/case study	5
Total		100

- o **Employability for the Course:** cyber security expert, security engineer, network security engineer

DISCIPLINE SPECIFIC ELECTIVE COURSES

With No SPECIALIZATION

KU5DSECAP317: CRYPTOGRAPHY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP317	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2

Course Description: This course introduces the fundamental concepts and techniques of cryptography and network security. It covers classical and modern encryption methods, symmetric and public key cryptosystems, message authentication, digital signatures, hash functions, and key management techniques.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the principles of information security, cryptography, and cryptanalysis.	U
2	Illustrate symmetric, asymmetric key cryptosystems for secure communication.	An
3	Study hash functions, digital signatures, and message authentication mechanisms.	An
4	Interpret public, secret key distribution techniques for secure communication.	An
5	Develop the foundation required for advanced studies in cyber security and secure systems	A C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
------------	------	------	------	------	------	------	------

CO1	3	2	2	3	1	2	3
CO2	3	2	3	3	1	2	2
CO3	3	2	3	3	1	2	2
CO4	3	2	3	3	1	2	2
CO5	3	2	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE Fundamentals of Cryptography and Classical Ciphers		12
	1	Security attacks, Services, Mechanisms. Steganography, Cryptography vs Cryptanalysis.	
	2	Classical encryption techniques – Symmetric cipher model.	
	3	Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers,	
	4	Caesar cipher, Affine cipher, Playfair cipher, Vigenere cipher, Hill cipher	
		Transposition ciphers – Keyless, Keyed, Double transposition.	
		Kirchhoff's principle, Introduction to cryptanalysis techniques	
2	MODULE TITLE (Modern Symmetric Key Cryptosystems)		12
	1	Symmetric key ciphers – Block vs Stream ciphers	
		Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers.	
	2	Data Encryption Standard (DES) – Structure, Key generation, Design criteria,	
	3	Weaknesses, Double DES, Triple DES	

		Security of DES - brute-force attack, differential cryptanalysis, linear cryptanalysis.	
	5	Stream ciphers – Structure, RC4.	
3	MODULE TITLE(Public Key Cryptosystems)		12
	1	Introduction to public key cryptosystems – Principles, Applications, Requirements, Conventional vs Public key cryptosystems.	
	2	RSA cryptosystem – Algorithm, Security, Attacks.	
		Public Key Cryptanalysis.	
4	MODULE TITLE (Message Integrity and Authentication, Key management)		12
	1	Hash functions – Security requirements, Secure Hash Algorithm (SHA-512).	
	2	Digital signatures – Attacks, Forgeries, Requirements, RSA digital signature	
	3	Key management – Distribution of secret keys using symmetric and asymmetric encryption, Distribution of public keys.	
		Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack.	
5	Teacher Specific Module :		12
	<i>Directions</i>		
	Virtual Labs. Use virtual experiments to understand the basic mathematical foundations of cryptography, to gain insightful experience by working with fundamental cryptographic applications and to train in the art of design and analysis of information security protocols. https://www.opencompsci.com/books/security-lab-manual		

Essential Readings:

3. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Pearson Ed.
4. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.

Online Courses

1. NPTEL Foundations of Cryptography, By Dr. Ashish Choudhury, IIIT Bangalore, <https://nptel.ac.in/courses/106106221>.
2. Cryptography, by Prof Don Boneh, Stanford University <https://www.coursera.org/learn/crypto>.

Web resources

3. [Welcome to Virtual Labs](#).
4. <https://www.opencompsci.com/books/security-lab-manual>

Suggested Readings:

4. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
5. 2. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007.
6. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall/CRC publication, 2nd edition, 2014.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	10
Total		100

KU5DSECAP318 : CLOUD COMPUTING AND BIG DATA

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP318	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 minutes

Course Description: This course introduces the fundamentals of Cloud Computing and Big Data technologies. Students learn cloud models, virtualization, distributed systems, and big data processing tools. Basic hands-on experience is included.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand cloud computing concepts, models, and virtualization	U
2	Apply cloud platforms for storage and simple deployment	A
3	Understand big data concepts and frameworks	An
4	Analyze big data processing techniques and tools	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	1	1	1		2	1
CO 2	2	2	2	2	1	3	1
CO 3	2	2	2	1	3	2	2
CO 4	3	3	2	2	3	2	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE- INTRODUCTION TO CLOUD COMPUTING		11
	1	Basics of Cloud Computing: Definition of cloud computing, Evolution of computing, Need and advantages of cloud computing, Limitations	
	2	Characteristics of Cloud- On-demand self-service, Broad network access, Resource pooling, Rapid elasticity Measured service	
	3	Service Models- IaaS, PaaS, SaaS, Comparison with examples, Real-world applications	
	4	Deployment Models- Public, Private, Hybrid, Community cloud Advantages and disadvantages	
	5	Virtualization- Concept and working, Types of virtualizations, Hypervisors (Type 1 & Type 2), Benefits and limitations	
2	MODULE TITLE- CLOUD PLATFORMS AND SERVICES		10
	1	Cloud Platforms Overview- Amazon Web Services , Microsoft Azure ,Google Cloud , Basic services comparison	
	2	Cloud Storage- Object storage, Block storage, File storage, Backup and recovery	
	3	Cloud Architecture- Frontend and backend ,Components ,Load balancing basics	
	4	Cloud Security- Authentication and authorization, Identity management, Data privacy	
	5	Cloud Operations- Creating virtual machines, Deploying applications, Monitoring and scaling basics	
3	MODULE TITLE- INTRODUCTION TO BIG DATA		12
	1	Big Data Basics- Definition and importance, Sources of big data	
	2	5Vs of Big Data- Volume, Velocity, Variety, Veracity, Value, Examples	

	3	Traditional vs Big Data -Limitations of traditional systems, Need for distributed systems.	
	4	Big Data Technologies -Hadoop, Apache Spark, Distributed computing basics	
	5	Applications of Big Data -Healthcare, Banking, E-commerce, Social media	
4	MODULE TITLE- BIG DATA PROCESSING AND TOOLS		12
	1	Hadoop Ecosystem -Introduction and architecture ,Components	
	2	HDFS - Working of HDFS, NameNode and DataNode	
	3	MapReduce -Concept and phases, Example: Word Count	
	4	NoSQL Databases -Introduction ,Types (Key-value, Document-based) ,MongoDB basics	
	5	Data Analytics & Visualization -Basics of data analytics, Introduction to visualization tools, Use cases	
5	Teacher Specific Module :		30
	<i>Directions</i>		
	1. Create an account in Amazon Web Services or Microsoft Azure and explore the dashboard. 2. Identify any two cloud services and write their purpose. 3. Create a virtual machine (VM), start it, and then stop it. 4. Upload a file to cloud storage and view/download the file. 5. Install Hadoop and verify whether it is working. 6. Create a folder in HDFS, upload a file, and display its contents. 7. Open Apache Spark and execute a simple command. 8. Load a sample dataset and perform a basic operation (like count or display).		

Essential Readings:

1. **Rajkumar Buyya, James Broberg, and Andrzej Goscinski**, *Cloud Computing: Principles and Paradigms*, Wiley India Pvt. Ltd., 2011.
2. **Tom White**, *Hadoop: The Definitive Guide*, 4th Edition, O'Reilly Media, 2015.

3. **Thomas Erl, Wajid Khattak, and Paul Buhler**, *Big Data Fundamentals: Concepts, Drivers & Techniques*, Prentice Hall, 2016.

Suggested Readings:

1. **Anthony T. Velte, Toby J. Velte, and Robert Elsenpeter**, *Cloud Computing: A Practical Approach*, McGraw-Hill Education, 2010.
2. **Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia**, *Learning Spark: Lightning-Fast Big Data Analysis*, O'Reilly Media, 2015.
3. **Seema Acharya and Subhashini Chellappan**, *Big Data Analytics*, Wiley India Pvt. Ltd., 2015.
4. **Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi**, *Mastering Cloud Computing*, McGraw-Hill Education, 2013.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar/Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

KU5DSECAP319 : ARTIFICIAL INTELLIGENCE AND NEURAL NETWORK

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP319	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2

Course Description: The course Artificial Intelligence and Neural Network introduces students to the fundamental concepts, techniques, and applications of Artificial Intelligence (AI) and Neural Networks. The course covers intelligent agents, problem-solving methods, search techniques, knowledge representation, machine learning approaches, probabilistic reasoning, and expert systems.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of Artificial Intelligence	U
2	Apply AI problem-solving and search techniques	A
3	Explain and use knowledge representation methods	U/A
4	Differentiate various machine learning approaches	A
5	Analyze the structure and functioning of Artificial Neural Networks	A

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create ©**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	1	2	2	1	3
CO2	3	3	2	2	2	1	3
CO3	3	2	2	3	2	1	2
CO4	3	3	2	2	3	1	3
CO5	3	3	2	2	3	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE: Introduction to Artificial Intelligence		
	1	Definition and scope of Artificial Intelligence	12
	2	History and applications of AI, Intelligent agents and environments	
	3	Problem solving using AI	
	4	Search techniques: Breadth First Search (BFS), Depth First Search (DFS), Hill Climbing, Best First Search, A* Algorithm	
	5	Knowledge representation: Predicate logic, Semantic networks, Rule-based systems	
2	MODULE TITLE: Machine Learning and Reasoning		
	1	Introduction to Machine Learning	12
	2	Types of learning: Supervised learning, Unsupervised learning, Reinforcement learning	
	3	Probabilistic reasoning, Bayesian networks	
	4	Handling uncertainty in AI	
	5	Expert systems: Architecture, Components, Applications	
3	MODULE TITLE: Artificial Neural Networks		

	1	Biological neuron vs artificial neuron: Structure of neural networks	12
	2	Types of neural networks: Feedforward networks, Feedback/Recurrent networks	
	3	Activation functions: Perceptron model, Multilayer perceptron (MLP), Backpropagation algorithm	
4	MODULE TITLE:Deep Learning and AI Applications		
	1	Introduction to Deep Learning : Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Natural Language Processing (NLP), Computer Vision (concepts only)	12
	2	AI applications in: Healthcare, Banking, Education, Robotics	
	3	Ethical issues in AI	
5	Teacher Specific Module :		12
	<i>Case studies</i>		
	Complete any one case study(Examples given below) based on a selected application, choose one application. The study should include a brief description of the selected application, its objective, real-world use cases, and a simple explanation of how the system works. Students are required to implement a basic working model Employee Salary Prediction Problem Statement : Organizations need systems to estimate employee salary ranges automatically using employee data. Predict the salary of an employee based on experience, education level, and skill rating. Input Parameters : Years of experience, Qualification, Technical skill score, Number of certifications Real-Time Application: Used in HR analytics and recruitment systems. Weather Prediction Problem Statement: Weather conditions can be predicted using historical climate data. Predict whether the weather will be rainy or sunny based on environmental conditions Input Parameters: Temperature, Humidity, Wind speed, Atmospheric pressure		

	Real-Time Application: Used in weather forecasting systems. Customer Churn Prediction Problem Statement : Businesses need to retain customers by predicting possible customer churn. Identify customers who are likely to stop using a service. Input Parameters: Monthly usage, Subscription period, Customer complaints, Payment history Real-Time Application: Used in telecom, banking, and online services. Traffic Sign Recognition Problem Statement : Intelligent transportation systems require automatic traffic sign detection for driver assistance. Recognize traffic signs automatically from images. Input Parameters : Traffic sign images, Shape, Color, Symbol patterns Real-Time Application : Used in self-driving cars and smart traffic systems. Product Recommendation System using Artificial Neural Network Problem Statement : Online shopping platforms need recommendation systems to improve customer experience. Recommend products to users based on previous purchases and interests. Input Parameters : Purchase history, Product ratings, Browsing history, User preferences Real-Time Application : Used in e-commerce websites and streaming platforms.	
--	---	--

Essential Readings:

1. **Stuart Russell and Peter Norvig**, *Artificial Intelligence: A Modern Approach*, Pearson Education / Prentice Hall
2. **Elaine Rich, Kevin Knight and Shivashankar B. Nair**, *Artificial Intelligence*, McGraw Hill Education.

3. **S. Rajasekaran and G. A. Vijayalakshmi Pai** *Neural Networks, Fuzzy Logic and Genetic Algorithms*
4. **Ian Goodfellow, Yoshua Bengio and Aaron Courville**, *Deep Learning*, MIT Press
5. **Simon Haykin**, *Neural Networks and Learning Machines*, Pearson Education.
6. **S. N. Sivanandam, S. Sumathi and S. N. Deepa**, *Introduction to Neural Networks using MATLAB*, Tata McGraw Hill.

Suggested Readings:

1. Artificial Intelligence and Neural Networks – V. S. Janakiraman, Scitech Publications.
2. Recommended Free Datasets: Kaggle, UCI Machine Learning Repository, TensorFlow Datasets

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	3
c)	Viva/Seminar	6
d)	Case Study	6
Total		100

KU5DSECAP320: COMPUTER ORGANIZATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
V	DSE	300-399	KU5DSECAP320	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2

Course Description:

This course introduces the internal structure and functioning of a computer system, focusing on how hardware components interact to execute programs. It covers **data representation, digital logic, CPU organization, memory systems, and input/output mechanisms**. Students will gain foundational knowledge of how instructions are executed at the hardware level, bridging the gap between **high-level programming and machine-level operations**. The course also builds analytical skills required for understanding system performance and architecture.

Course Prerequisite: Knowledge in Digital Systems

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the basic organization and components of a computer system	U
2	Explain data representation and perform basic binary arithmetic	R
3	Describe CPU organization and instruction execution cycle	R
4	Analyze memory hierarchy and input/output organization	An
5	Interpret how hardware design impacts system performance	E

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	2	1	1	1	1	1
CO 2	3	2	1	1	1	1	2
CO 3	3	2	2	2	2	2	2
CO 4	3	3	2	2	2	2	2
CO 5	2	3	3	3	3	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Basic Structure of Computer Systems		12
	1	Functional units: CPU, Memory, Input/Output	
	2	Stored program concept	
	3	Instruction execution cycle	
	4	Register organization	
	5	Bus systems and data transfer	
2	Data Representation & Digital Logic		12
	1	Number systems (binary, octal, hexadecimal)	
	2	Signed number representation (1's & 2's complement)	
	3	Binary arithmetic operations	
	4	Character representation (ASCII, Unicode)	
	5	Basics of logic gates and Boolean algebra	
3	CPU Organization & Instruction Set		12

	1	Instruction formats and types	
	2	Addressing modes	
	3	Arithmetic Logic Unit (ALU)	
	4	Control unit (hardwired & microprogrammed)	
	5	Pipelining basics	
4	Memory & Input/Output Organization		12
	1	Memory hierarchy (cache, main memory, secondary storage)	
	2	Cache memory concepts	
	3	Virtual memory basics	
	4	I/O organization and interfaces	
	5	Interrupts and DMA (Direct Memory Access)	
5	Teacher Specific Module :		12
	Case study : Timeline of processors, Instruction Cycle Execution, Interrupt Handling Mechanism, GPU Architecture and Graphics Processing, Hardware Security Mechanisms, Motherboard organization		
	1. Draw and label the basic block diagram of a computer system 2. In a real computer hardware students must be able to identify RAM, Processor, Motherboard, Hard Disk and SMPS 3. https://logic.ly/demo/ or https://academo.org/demos/logic-gate-simulator/ a. Basic Logic Gate Verification Activity, Verify the truth tables of basic logic gates: AND, OR, NOT, NAND, NOR, XOR. Give different binary inputs (0 and 1) and observe the output. b. Design a simple arithmetic circuit. Simulate a half adder circuit and verify: Sum output and Carry output 4. Draw Memory Hierarchy Pyramid diagram showing Speed, Storage capacity and Cost		

Essential Readings:

1. Computer Organization and Architecture” – William Stallings
2. Computer System Architecture” – Morris Mano
3. Structured Computer Organization” – Andrew S. Tanenbaum

Suggested Readings:

1. Computer Organization” – Carl Hamacher, Zvonko Vranesic & Safwat Zaky

2. Digital Design” – Morris Mano
3. Modern Computer Architecture and Organization” – Jim Ledin
4. NPTEL / SWAYAM video lectures on Computer Organization

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	5
d)	Case study	5
Total		100

KU5DSECAP321 : COMPUTER GRAPHICS

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
V	DSE	300-399	KU5DSECAP321		4(3+1)	75
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces the fundamental concepts of Computer Graphics with emphasis on graphics systems, image representation, geometric transformations, viewing, clipping, 2D and 3D graphics, and animation techniques. The course also familiarizes students with practical graphics programming concepts and applications used in multimedia, gaming, visualization, CAD, and web graphics.

Course Prerequisite: Computer Basics

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamentals of computer graphics systems, display devices, and graphics primitives.	R,U
2	Apply line, circle, and polygon drawing algorithms in graphics applications.	U,A
3	Perform 2D and 3D geometric transformations and viewing operations	A,An
4	Analyse clipping, windowing, and hidden surface techniques used in graphics rendering	An,E
5	Create simple animation and interactive graphics applications using suitable programming tools	A,C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	3	2	2	1	1	1	1
CO 2	3	3	2	2	1	1	1
CO 3	3	3	3	2	2	1	1
CO 4	2	3	3	2	2	2	1
CO 5	2	3	3	3	2	2	2

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE Introduction to Computer Graphics		10
	1	Introduction to Computer Graphics – Applications of Computer Graphics – Components of Graphics Systems	
	2	Video Display Devices: CRT, Raster Scan, Random Scan, LCD, LED, Plasma Displays	
	3	Input and Output Devices – Graphics Software and Standards	
	4	Points, Lines, Circles, Ellipses and Polygon Representation	
2	MODULE Graphics Drawing Algorithms		11
	1	Scan Conversion – Line Drawing Algorithms: DDA and Bresenham's Line Algorithm	
	2	Circle and Ellipse Generation Algorithms	
	3	Polygon Filling Techniques – Flood Fill and Boundary Fill Algorithms	
	4	Character Generation and Attributes of Output	
3	MODULE 2D and 3D Transformations		12
	1	Basic Transformations: Translation, Rotation, Scaling, Reflection and Shearing	

	2	Composite Transformations and Homogeneous Coordinates	
	3	2D Viewing and Window-to-View port Transformation	
	4	3D Transformations and Projections	
4	MODULE Viewing, Clipping and Animation		12
	1	Clipping Operations – Point, Line and Polygon Clipping Algorithms	
	2	Cohen–Sutherland and Liang–Barsky Line Clipping Algorithms	
	3	Hidden Surface Removal Methods and Visible Surface Detection	
	4	Introduction to Animation – Key Frame Animation and Graphics Applications	
5	Teacher Specific Module : For practicals		30
	All exercises must be done using OpenGL Write a program to draw basic 2D shapes Line, Triangle, Rectangle, Circle, Home - Line drawing algorithms - Circle Drawing - Line Clipping 2D transformations - Create a simple animation : Moving car, Bouncing ball, Rotating fan <i>Case study : Create one Mini Graphics Project / Animation Demonstration</i>		

Essential Readings:

1. Donald Hearn and M. Pauline Baker, *Computer Graphics with OpenGL*, Pearson Education.
2. Foley, Van Dam, Feiner and Hughes, *Computer Graphics: Principles and Practice*, Pearson.
3. Steven Harrington, *Computer Graphics: A Programming Approach*, McGraw Hill.
4. Newmann and Sproull, *Principles of Interactive Computer Graphics*, McGraw Hill.
5. Rajesh K. Maurya, *Computer Graphics with Virtual Reality Systems*, Wiley India.
6. Zhigang Xiang and Roy Plastock, *Computer Graphics*, Schaum's Outline Series

Suggested Readings:

1. David F. Rogers, *Procedural Elements for Computer Graphics*, McGraw Hill.

2. F.S. Hill Jr., *Computer Graphics using OpenGL*, Pearson.
3. William M. Newman and Robert F. Sproull, *Principles of Interactive Computer Graphics*, McGraw Hill.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- o **Employability for the Course:** The course equips students with foundational skills required for careers in multimedia development, UI/UX design, game development, animation, CAD applications, visualization systems, and graphics programming.

KU6DSECAP322: SOFT COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP322	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	-	-	30	70	100	2

Course Description: This course introduces the fundamental concepts and techniques of soft computing, including Artificial Neural Networks, Fuzzy Logic, and Genetic Algorithms. It focuses on modeling and solving complex real-world problems using approximate reasoning and adaptive learning methods.

Course Prerequisite: NIL

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the fundamental concepts of soft computing, including artificial neural networks, their learning methods, applications, and hybrid systems.	U
2	Apply the principles of fuzzy logic and perform operations on fuzzy sets and relations for problem-solving.	A
3	Analyze and implement fuzzy systems using membership functions, fuzzification, defuzzification, and fuzzy inference mechanisms.	An
4	Understand the concepts and applications of genetic algorithms.	U

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7
CO 1	2	1		1			2
CO 2	2	2	1	1			2
CO 3	3	2	2	2	1		3
CO 4	4	3	2	2	2	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1		MODULE TITLE- INTRODUCTION TO ARTIFICIAL NEURAL NETWORK	12
	1	Introduction to Soft Computing, "Soft" versus "Hard" computing, Applications of Soft computing techniques	
	2	Artificial Neural Network, Biological Neural Network , Brain vs. Computer - Comparison between Biological Neuron and Artificial Neuron (Brain vs. Computer)	
	3	Basic Models of Artificial Neural Network - Connections- Learning: Supervised Learning, Unsupervised Learning, Reinforcement Learning.	
	4	Building Blocks: Input layer, hidden layers, output layer, weights, bias, activation functions: Step, Linear, Sigmoid, Tanh, and ReLU functions.	
	5	Single Layer Perceptron: Architecture	
	6	Multilayer Feedforward Neural Network, Loss function	
	7	Backpropagation algorithm, Gradient descent	

2	MODULE TITLE- FUZZY LOGIC		12
	1	Introduction to Fuzzy Logic Classical Sets (Crisp Sets)- Operations on Classical Sets: Union, Intersection, Complement, Difference (Subtraction) Properties of Classical Sets Function Mapping of Classical Sets	
	2	Fuzzy Sets- Fuzzy Set Operations: Union, Intersection, Complement Properties of Fuzzy Sets	
	3	Relations : Introduction Cartesian Product of Relation	
	4	Classical Relation- Cardinality, Operations, Properties and Composition	
	5	Fuzzy Relations- Cardinality of Fuzzy Relations, Operations on Fuzzy Relations, Properties of Fuzzy Relations, Fuzzy Composition	

3	MODULE TITLE- MEMBERSHIP FUNCTIONS, DEFUZZIFICATION AND FUZZY REASONING		12
	1	Membership Functions: Introduction, Features Fuzzification Methods of Membership Value Assignments: Intuition, Inference, Rank Ordering	
	2	Defuzzification : Introduction, Lambda-Cuts for Fuzzy Sets (Alpha-Cuts), Lambda-Cuts for Fuzzy Relations	
	3	Fuzzy Rule- Introduction, Truth Values and Tables in Fuzzy Logic, Fuzzy Propositions, Formation of Rules, Decomposition of Rules (Compound Rules), Aggregation of Fuzzy Rules	
	4	Fuzzy Reasoning (Approximate Reasoning): Categorical Reasoning, Qualitative Reasoning	
	5	Fuzzy Inference Systems (FIS)- Construction and Working Principle of FIS, Methods of FIS- Mamdani FIS	

4	MODULE TITLE- GENETIC ALGORITHM		12
	1	Introduction to Genetic Algorithms, Importance of Genetic Algorithms, Biological Background: The Cell, Chromosomes, Genetics, Reproduction, Natural Selection. Generic Algorithm vs. Traditional Algorithms	

	2	Basic Terminologies in Genetic Algorithm: Individuals, Genes, Fitness and Populations Simple GA General Genetic Algorithm	
	3	Operators in Generic Algorithm 7. Encoding: Binary Encoding, Octal Encoding 8. Selection: Roulette Wheel Selection, Random Selection, Rank Selection, Tournament Selection 9. Crossover (Recombination) : Single Point Crossover, Two-Point Crossover, Multipoint Crossover (N-Point Crossover), Uniform Crossover 10. Mutation: Flipping, Interchanging, Reversing	
	4	Stopping Condition for Generic Algorithm Flow: Best Individual, Worst individual, Sum of Fitness, Median Fitness Constraints in Genetic Algorithm Advantages and Limitations of Genetic Algorithm Applications of Generic Algorithm	
	5	Hybrid Systems: Neuro Fuzzy Hybrid Systems, Neuro Genetic Hybrid Systems, Fuzzy Genetic Hybrid Systems.	

5	Teacher Specific Module :	12
	Discuss real-world applications for every concept taught. Explain how each soft computing technique is applied in practical systems. Demonstrate applications using case studies, simulations, or simple projects.	

Essential Readings:

- 6) S.N.Sivanandam, S.N.Deepa, "Principles of Soft Computing", Wiley India Pvt. Ltd., 2nd Edition, 2011.
- 7) S.Rajasekaran, G.A.Vijayalakshmi Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithm, Synthesis and Applications ", PHI Learning Pvt. Ltd., 2017.

Suggested Readings:

- 5) Siman Haykin, "Neural Networks" Prentice Hall of India
- 6) Timothy J. Ross, "Fuzzy Logic with Engineering Applications" Wiley India.
- 7) Kumar Satish, "Neural Networks" Tata Mc Graw Hill.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar/ Case study	10
Total		100

KU6DSECAP323: MOBILE APPLICATION DEVELOPMENT USING FLUTTER

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP323	4(3+1)	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces students to cross-platform mobile application development using Flutter and Dart. The course covers Dart programming fundamentals, Flutter widget-based UI design, state management, navigation, local data storage, and API integration. Students learn to design responsive user interfaces, manage application state, handle user interactions, and develop data-driven mobile applications. Through hands-on laboratory exercises and mini projects, students gain practical experience in building functional Android/iOS applications using modern mobile development practices.

Course Prerequisite: Basic knowledge of computer programming concepts

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the core concepts of Dart required for Flutter development.	U
2	Learn Flutter architecture and build basic UI using widgets.	U, A
3	Handle user interaction, dynamic UI updates, and navigation.	A
4	Integrate APIs, process JSON data, and implement local data persistence using SharedPreferences and SQLite with sqflite.	A
5	Design and develop mobile applications	A

****Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)***

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	3	1	2	1	1	3
CO2	3	3	1	2	1	1	3
CO3	3	3	1	2	1	2	2
CO4	3	3	3	2	2	2	2
CO5	3	3	2	3	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Dart Programming Fundamentals		10
	1	Introduction to Dart: Overview of Dart, Installation & setup, Writing first program (Hello World).	
	2	Data Types & Variables: var, final, const, int, double, String, bool.	
	3	Collections: List, Map, Set.	
	4	Control Flow: if-else, loops, Functions, parameters, arrow functions.	
	5	Object-Oriented Programming: Classes, objects, constructors.	
	6	Null Safety & Error Handling: Null safety (?, !, late), try-catch.	
	7	Asynchronous Programming: Future, async/await.	
2	MODULE TITLE : Flutter Fundamentals & UI Basics		11
	1	Introduction to Flutter: Flutter SDK setup, Project structure, Hot reload.	
	2	Widget System: Everything is a widget, Widget tree concept, Stateless vs stateful widgets.	

	3	Layout & Core Widgets: Row, Column, Container, Padding, Expanded, Scaffold, AppBar, Text, Image, Buttons.	
	4	Design Systems: Material Design vs Cupertino, Basic theming.	
	MODULE TITLE : State Management & User Interaction		12
3	1	Stateful Widgets & Lifecycle: setState(), and other lifecycle methods, Basic lifecycle concepts.	
	2	User Input & Forms: TextField and controllers, Input validation.	
	3	Lists & Dynamic UI: ListView.builder, Displaying dynamic data.	
	4	Navigation: Navigator.push() and pop(), Passing data between screens.	
	5	Introduction to State Management: Limitations of setState, Basic concept of Provider.	
	MODULE TITLE Application Development & Data Management		12
4	1	API Integration & JSON Handling: HTTP requests, JSON parsing, Loading and error handling.	
	2	Local Storage: SharedPreferences (key-value storage). Introduction to databases. SQLite using sqflite (Database creation, CRUD operations).	
	3	Code Organization: Folder structure, Reusable widgets, Basic separation of concerns	
	4	Data Passing Between Screens: Navigator arguments.	
	Teacher Specific Module :		30
	<i>Directions</i>		
5	1) Lab set up(Installation and Configuration of the required software) 2) Student Record Manager (Console Application) Add, update, delete student records. Use List/Map, functions, loops. Implement error handling and async delay. 3) Personal Profile App Create a profile UI with image, text, and buttons. Layout widgets and styling. 4) To-Do List Application Add, delete, and update tasks. Use ListView, TextField, navigation. Manage state using setState. 5) Notes / Task Manager App with Database Add, edit, delete tasks. Store data using SQLite.		

	Display persistent data. ○ Use navigation and structured UI.	
	6) Case study : Study the design and development of a real-world mobile application using Flutter and SQLite.	

Essential Readings:

11. Dart Official Documentation: <https://dart.dev/guides>. (Module 1)
12. Raywenderlich Tutorial Team, Jonathan Sande,, Matt Galloway, Dart Apprentice (First Edition): Beginning Programming with Dart (Module 1)
13. Flutter Official Documentation: <https://docs.flutter.dev>. (Module 2)
14. Vincent Ngo, Stef Patterson, Alejandro Ulate Fallas & Kevin D Moore, Flutter Apprentice Feb 21 2024 , Dart 3.2.6, Flutter 3.16.9 (Module 2)
15. Material Design: <https://material.io>. (Module 2)
16. Flutter state management doc:
<https://docs.flutter.dev/data-and-backend/state-mgmt>. (Module 3)
17. Provider Documentation: <https://pub.dev/packages/provider>. (Module 3)
18. http package: <https://pub.dev/packages/http>. (Module 4)
19. SharedPreferences: https://pub.dev/packages/shared_preferences. (Module 4)
20. sqflite: <https://pub.dev/packages/sqflite>. (Module 4)

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Case study	8
Continuous Evaluation (Practical)		10

a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

- o **Employability for the Course:** Mobile app startups, App development

KU6DSECAP324: INTRODUCTION TO BLOCKCHAIN

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP324	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	-	30	70	100	2 Hours

Course Description: Blockchain technology has generated massive interest among governments, enterprises, and academics, because of its capability of providing a transparent, secured, tamper-proof solution for interconnecting different stakeholders in a trustless setup. This course will cover the basic design principles of Blockchain technology and its applications over different sectors.

Course Prerequisite: Expertise in programming, basic knowledge of computer security, cryptography, networking

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Able to demonstrate the cryptographic primitives in Blockchain technology	U
2	Understand the basic blockchain architecture	U
3	Understand various consensus protocols	U
4	Understand smart contracts, blockchain oracles, and deployment concepts.	U,A
5	Understand the impact and role of Blockchain Technology in financial, information, and other infrastructures.	U,An

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	3	2	1	3
CO2	3	2	3	2	2	1	3
CO3	3	2	2	3	2	1	3
CO4	3	3	2	3	2	1	3
CO5	3	2	2	3	3	3	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE Introduction of Blockchain, Fundamentals of Cryptography		12
	1	Centralised and Decentralised systems (use case: Supply chain management), peer to peer systems	
	2	Blockchain: Introduction, History, Properties of block chain, Decentralisation with Block chain	
	3	Basic crypto primitives: Hash, Secure Hash Algorithms – SHA-256	
	4	Digital signatures – RSA digital signature algorithms.	
	5	Asymmetric cryptography – RSA, Elliptic curve cryptography	
2	MODULE TITLE: BLOCKCHAIN ARCHITECTURE		12
	1	Distributed Systems for Decentralization, Distributed consensus, distributed consensus problem (Concept only)	
	2	Evolution of cryptocurrency	
	3	Open Consensus and Bitcoin	

	4	Blockchain Elements: Blocks structure, block header, mining a block, block generation puzzle, Merkle trees, Hash Pointer, Hash Chain, Chain of Blocks	
	5	Blockchain – The genesis block. Nakamoto Consensus (Proof of Work), Mining – Tasks of miners, mining algorithm	
	MODULE TITLE : CONSENSUS		12
3	1	Consensus for Permissioned Models, Permissionless Models	
	2	Consensus mechanism: Open consensus beyond Proof of Work (PoW), Proof of Stake (PoS),	
	3	Byzantine Fault Tolerance (BFT), Proof of Authority (PoA)	
	4	Ethereum Introduction, Gas in Ethereum	
	MODULE TITLE : SMART CONTRACT, USECASES		12
4	1	Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts.	
	2	Use cases of Blockchain technology – Government, Health care, Finance, Supply chain management.	
	Teacher Specific Module : For practical courses add lab experiments		12
5	Various Use cases for case study Land registry record, Voting Blockchain in Financial Services, Public Sector Use Cases Blockchain for Decentralized Marketplace		

Essential Readings:

1. Imran Bashir, “Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained”, 2nd Edition, Packt Publishing Ltd, March 2018.
2. Daniel Drescher Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress 2017
3. Cryptography and Network Security-Principles and Practice, 8th Edition, by Stallings Agenda Item 65/39 - Annexure - 35 William, published by Pearson, 2020
4. Asharaf S, N. Sivadas, A. S, and J. Franklin, Blockchain Technology: Algorithms and Applications, Wiley India Pvt. Ltd, 2023.

5. Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, “Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger”, Packt Publishing Limited, 2018.

MOOC

1. nptel.ac.in/courses/106105235
2. <https://www.udemy.com/course/build-your-blockchain-az/>

Suggested Readings:

1. Andreas M. Antonopoulos , “Mastering Bitcoin: Unlocking Digital Cryptocurrencies”, O’Reilly Media Inc, 2015
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, “Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction”, Princeton University Press, 2016.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Theory		70
CCA		30
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Case study	10
Total		100

- **Employability for the Course: Blockchain Developer**

KU6DSECAP325 : PARALLEL COMPUTING CONCEPTS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP325	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	-	30	70	100	2 Hours

Course Description: The syllabus covers the basic understanding and advanced concepts of parallel computing. It covers Principles of Parallel Algorithm Design, the Message Passing Paradigm, Process Communications,, Programming Shared Address Space Platforms, multiple threads in parallel computing etc.

Course Prerequisite: Computer Organization and Operating System principles

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Understand the key parallel computational models	U
2	Familiarise and apply parallel and distributed algorithms in problem Solving	U A
3	Learn the communication Models and message passing for parallel algorithm development	U A
4	Develop parallel algorithms using message passing paradigm	C
5	Formulate parallel algorithms for shared memory architectures	C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	2	2	2	1	2
CO2	3	3	2	2	2	1	2

CO3	2	3	3	2	2	1	2
CO4	3	3	3	2	2	1	3
CO5	3	3	3	2	2	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE: Introduction to Parallel Computing		12
	1	What is Parallel Computing, Why Use Parallel Computing, Scope of Parallel Computing	
	2	von Neumann Computer Architecture, Flynn's Classical Taxonomy, General Parallel Terminology	
	3	Parallel Computer Memory Architectures : Shared Memory, Distributed Memory, Hybrid Distributed-Shared Memory	
	4	Parallel Programming Models: Overview , Shared Memory Model, Threads Model, Distributed Memory / Message Passing Model, Data Parallel Model, Hybrid Model, SPMD and MPMP	
2	MODULE: Parallel Programming Platforms		12
	1	Implicit Parallelism: Trends in Microprocessor Architectures	
	2	Dichotomy of Parallel Computing Platforms	
	3	Physical Organization of Parallel Platforms	
	4	Communication Costs in Parallel Machines, Routing Mechanisms for Interconnection Networks	
	5	Impact of Process-Processor Mapping and Mapping Techniques	
3	MODULE: Principles of Parallel Algorithm Design		12
	1	Preliminaries , Decomposition Techniques	
	2	Characteristics of Tasks and Interactions	
	3	Mapping Techniques for Load Balancing	

	4	Methods for Containing Interaction Overheads	
	5	Parallel Algorithm Models	
4	MODULE: Basic Communication Operations		12
	1	One-to-All Broadcast and All-to-One Reduction	
	2	All-to-All Broadcast and Reduction	
	3	All-Reduce and Prefix-Sum Operations, . Scatter and Gather	
	4	All-to-All Personalized Communication, Circular Shift	
	5	Improving the Speed of Some Communication Operations	

5	Teacher Specific Module		12
	Activities: 1. Case study on supercomputers (find out the system configurations and processor performance, prepare a table of comparison for top five supercomputers) 2. Create simple multithreading programs and compare sequential and parallel execution time (Using Java) 3. Learn about multicore processors in modern computers, inspect their computer or smartphone specifications and identify: Number of CPU cores, Processor type, Clock speed and prepare a small comparison table.		

Essential Readings:

1. Introduction to Parallel Computing, Blaise Barney, Lawrence Livermore National Laboratory
2. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, 2nd Ed, Addison-Wesley, 2003
3. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 3rd Ed., Morgan Kaufman, 2016.
4. Steven Brawer, Introduction to Parallel Computing, Academic Press, (1989)

Suggested Readings:

1. Barbara Chapman, Gabriele Jost, Ruud van der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, MIT Press, 2008.

2. William Gropp, Ewing Lusk, Anthony Skjellum Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3rd Ed, MIT Press, 2014.

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
CCA		30
a)	Test Papers	15
b)	Assignment	5
c)	Viva/Seminar	5
d)	Activities	3
Total		100

KU6DSECAP326 : DIGITAL IMAGE PROCESSING

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
VI	DSE	300-399	KU6DSECAP326		4(3+1)	75
Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	-	35	65	100	1 Hr 30 Minutes

Course Description: This course introduces the fundamental concepts and techniques of Digital Image Processing. It covers image representation, pixel relationships, image enhancement methods, spatial and frequency domain processing, image transforms, filtering techniques, segmentation, edge detection, and feature extraction. Students will learn how digital images are analyzed, processed, and improved for various applications such as medical imaging, computer vision, remote sensing, multimedia, and pattern recognition. The course also provides practical exposure to implementing image processing techniques using suitable software tools and programming environments.

Course Prerequisite: Basic knowledge in computers and programming

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Explain basic concepts, components, and applications of digital images	R,U
2	Apply spatial domain techniques for image enhancement	U,A
3	Analyze frequency domain transformations and filtering methods	A,An
4	Evaluate segmentation and feature extraction techniques	An,E
5	Develop simple image processing solutions using software tools	A,C

**Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)*

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	3	2	1	2	1	1	2
CO2	3	3	2	2	2	1	2
CO3	3	3	2	3	2	1	2
CO4	3	2	2	3	3	1	2
CO5	3	3	2	2	2	2	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	Fundamentals of Digital Image Processing:		11
	1	Introduction to Digital Image Processing: Definition, components of image processing system, applications in medical imaging, satellite imaging, multimedia.	
	2	Steps in Image Processing: Image acquisition, preprocessing, segmentation, representation, recognition, interpretation	
	3	Image Sampling and Quantization: Continuous vs digital image, resolution, gray levels, aliasing effects	
	4	Color Image Processing: Basics of color perception, RGB, CMYK, HSV, YCbCr color models, Color transformations, Color enhancement	
	5	Image Representation and Storage: Pixel, image formats, bitmap vs vector images, storage requirements	
2	Image Enhancement and Spatial Domain Processing		12
	1	Basic Pixel Relationships: Neighbors, adjacency (4, 8, m),	

		connectivity, distance measures (Euclidean, Manhattan), Mean, median, variance, Standard deviation, Concept of noise in images, Pixel coordinates (x, y)	
	2	Introduction to Enhancement: Purpose, types of enhancement techniques. Gray Level Transformations: Negative, log transformation, power-law (gamma correction), contrast stretching	
	3	Histogram Processing: Histogram equalization, histogram specification, local enhancement	
	4	Spatial Filtering: convolution and correlation, linear filtering, mask operations, Smoothing filters (mean, median), sharpening filters (Laplacian, gradient)	
	5	Noise and Filtering: Types of noise (Gaussian, Salt & Pepper), noise reduction techniques	
3	Image Transformation and Frequency Domain Filtering		12
	1	Image Transform : 1D & 2D Fourier transform, properties. Walsh transforms, Hadamard Transform, Discrete cosine Transform, Discrete Wavelet Transform.	
	2	Frequency Domain Representation: Frequency components, spectrum analysis	
	3	Filtering Techniques: Low pass filters (ideal, Butterworth), high pass filters(Ideal, Butterworth, Gaussian)	
	4	Homomorphic Filtering: Illumination and reflectance components	
	5	Comparison of Domains: Spatial vs frequency domain methods, advantages and limitations	
4	Image Segmentation and Feature Extraction		10
	1	Segmentation Concepts: Importance, application and challenges of Segmentation.	
	2	Edge Detection: Operators (Sobel, Prewitt, Roberts), Canny edge detection	
	3	Thresholding Techniques: Global, local, Otsu's method.	
	4	Region-Based Segmentation: Region growing, region splitting and merging.	
	5	Representation & Feature Extraction: Boundary representation, descriptors, shape and texture features	
5	Teacher Specific Module 5: For practicals		30

	Directions: Implement the lab exercises using Python	
	Experiment 1: Image Representation Read and display an image Print image dimensions Experiment 2: Grayscale Conversion Convert RGB image to grayscale Compare intensity values Experiment 3: Histogram Analysis Plot histogram of an image Observe intensity distribution Experiment 4: Image Thresholding Convert grayscale image to binary Experiment with different threshold values Experiment 5: Noise Addition and Removal Add salt & pepper noise Apply median filter Experiment 6: Edge Detection Apply Sobel filter Observe edges Experiment 7: Image Transformation Resize image Rotate image Flip image	

Essential Readings:

1. Rafael C. Gonzalez & Richard E. Woods – *Digital Image Processing*
2. S. Jayaraman et al. – *Digital Image Processing*

Suggested Readings:

1. Anil K. Jain – *Fundamentals of Digital Image Processing*
2. William K. Pratt – *Digital Image Processing*
3. Online tutorials (Python OpenCV, MATLAB)

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		65
Theory		50
Practical		15
CCA		35
Continuous Evaluation (Theory)		25
a)	Test Papers	12
b)	Assignment	5
c)	Viva/Seminar	8
Continuous Evaluation (Practical)		10
a)	Lab Skills and Punctuality	3
b)	Observation Book	2
c)	Test	5
Total		100

Employability for the Course: Image Processing Analyst , Computer Vision Intern , Multimedia Developer and AI/ML Entry-Level Engineer

KU6DSECAP327: SOFTWARE TESTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
VI	DSE	300-399	KU6DSECAP327	4(4+0)	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	0	30	70	100	2 Hours

Course Description: This course introduces the fundamentals of Software Testing and Quality Assurance. It covers software testing concepts, testing levels, verification and validation, white box and black box testing techniques, graph and data flow testing approaches, and basic test automation using Selenium. The course also provides practical exposure to automated testing of simple web applications using Selenium. Students will learn to design test cases, analyze software behavior, and execute basic automated tests.

Course Prerequisite: Knowledge in Software Engineering

Course Outcomes:

CO No.	Expected Outcome	Learning Domains
1	Explain the fundamentals of software testing, software quality, and testing processes.	An
2	Differentiate various testing methods, testing levels, and testing approaches used in software development.	U An
3	Apply white box testing techniques including graph coverage and data flow testing methods.	U A
4	Apply black box testing techniques for input domain and functional testing.	A
5	Develop and execute basic automated test scripts using Selenium for simple web applications.	A C

***Remember (R), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C)**

Mapping of Course Outcomes to PSOs

COs / PSOs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	2	2	1	3	1	2	2
CO2	3	3	1	3	2	2	2
CO3	3	3	2	3	2	1	2
CO4	3	3	2	3	2	1	2
CO5	3	3	2	2	1	1	3

COURSE CONTENTS

Contents for Classroom Transaction:

M O D U L E	U N I T	DESCRIPTION	HOURS
1	MODULE TITLE : Introduction to Software Testing		12
	1	What is Software testing? Software Quality, Need of Testing. Testing Process - Level 0 - 4 thinking	
	2	Verification, Validation and Testing	
	3	Faults, Error and Bug	
	4	Test cases, Coverage Criteria.	
	5	Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing	
	6	Testing Methods - Black Box testing, White Box testing, Grey Box testing.	
	7	Manual, Automation Testing	
2	MODULE TITLE Unit Testing - White Box Approaches		12
	1	Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing	
	2	Mutation testing - Mutation and Mutants, Mutation operators, Mutation score.	

	3	Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage	
	4	Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria.	
	5	Graph Coverage for Source Code - Control flow graphs for code	
	6	Data Flow Testing at Design Level : Definition-use relationships, DU pairs, Inter-procedural DU pairs, Coupling DU pairs	
3	MODULE TITLE Unit Testing - Black Box Approaches		12
	1	Domain Testing / Input Space Partitioning - Partitions of a set.	
	2	Input domain modelling - Interface-based approach, Functionality-based approach.	
	3	Identifying values.	
	4	Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example.	
4	MODULE TITLE : Test Automation and introduction to Selenium		12
	1	Introduction to Selenium: Selenium components overview	
	2	Basic Selenium Operations :Launching browser, Opening and closing web pages, Navigation commands	
	3	Selenese Commands and Web Elements : open, click, type, verify, Handling textboxes, Radio buttons, Check boxes, Buttons and links	
	4	Keyboard, mouse actions	
	5	Simple Test Execution : Running Selenium test scripts, Viewing basic test results	
5	Teacher Specific Module : Activities		12
	1. Install and configure Selenium 2. Automate login form testing 3. Handle form elements using Selenium 4. Execute a simple automated test case 5. Case study: Develop and execute a simple automated test script for a simple web application using Selenium.		

Essential Readings:

1. Paul Ammann and Jeff Offutt, Introduction to Software Testing.

2. Software Testing: A Craftsman's Approach, by Paul C. Jorgensen, Third Edition
3. KshirasagarNaik and PriyadarshiTripathy, Software Testing And Quality Assurance: Theory And Practice.
4. Rex Black, "Foundations of Software Testing", Cengage Learning.
5. Dorothy Graham et al., "Experiences of Test Automation: Case Studies of Software Test Automation", Addison-Wesley Professional.
6. Alan Richardson, "Selenium WebDriver: From Foundations to Framework", Leanpub.
7. "Learn Selenium", UnmeshGundeck and Carl Cocchiaro

MOOC

1. [Software Testing Prof. Meenakshi D'Souza - YouTube](#)
2. [Preview: Software Testing | SWAYAM](#)

Suggested Readings:

1. King, James C, "Symbolic Execution and Program Testing", Association for Computing Machinery, July 1976.
2. Software Engineering by Rajib Mall, PHI 2014

Assessment Rubrics:

Evaluation Type		Marks
End Semester Evaluation		70
Continuous Evaluation		30
a)	Test Papers	15
b)	Assignment	5
c)	Activities & Case Study	10
Total		100

APPENDIX

SUMMER INTERNSHIP - WORK REGISTER FORMAT

Maintain a 100 page notebook for the Work Register

First Page

- Student Name
- Register Number
- Year & Semester
- Internship Title
- Organization Name, Address
- Duration (From – To)
- Internship Supervisor in the College
- Internship Mentor in the Organization, Name, designation and Contact

Second Page: Internship Details

- Organization short profile - one paragraph
- Department and work assigned for you
- Internship objectives
- Technologies/tools learnt/used
- Scope of work

Daily Work Log Format

Use the following structure for each day in a separate page:

Date :

Task Assigned :

Work Done - Description:

Tools / Technologies Used/Learnt:

Learning Outcome:

External Guide/Authority Signature:

Example Entries

Date: 06-04-2026

Task Assigned: Develop user input module for database

Work Done - Description: Created a Python script to accept user input (ID, name, age, course) and insert data into MySQL table. Implemented input validation and exception handling.

Tools/Technologies Used/Learnt: Python, MySQL, MySQL Connector

Learning Outcome: Learned dynamic user input handling, validation, and database insertion using Python

External Guide/Authority Signature:

Last Page - After completing the internship

1. Description about the technical skills gained

- Programming languages used/learnt
- DBMS used/learnt
- Tools used/learnt
- Frameworks used/learnt

2. Description about the soft skills developed

- Communication and Teamwork
- Time management
- Meeting the Deadlines
- Timely Submission of the work

3. Brief Report about the work

Acceptance Letter from the Internship Centre

Course Code: _____

1. Name of the Student: _____
2. Register No.: _____
3. Details of the Internship Centre Selected:
 - a. Name of the Centre: _____
 - b. Address of the Centre: _____

 - c. Nature of activities undergoing in the Centre: _____

 - d. Proposed kind of engagement by the candidate at the Centre: _____

 - e. Expected enhancement of skills/knowledge: _____

 - f. Name of the Mentor: _____
 - g. Contact No.: _____

Place: _____

Signature of the Student

Date: _____

COUNTERSIGNED

Name and Signature of the Mentor with Seal: