KANNUR UNIVERSITY
കണ്ണൂർ സർവകലാശാല

(Abstract)

Five Year Integrated Master of Computer Applications Programme at the Information Technology Education Centre (ITEC) , Dr . Janaki Ammal Campus , Palayad- Scheme and Syllabus (Third to Sixth Semesters only) - Approved and Implemented w .e .f 2025 Admission -Orders issued.

ACADEMIC C SECTION

ACAD J/ACAD J3/17099/2025 Dated: 29.08.2026

Read:-1. ACAD D/ ACAD D5/4513/2024 dated 28.03.2025

2. ACAD/ACAD C1/17099/2025 dated 16.09.2025 & 05/01/2026
3. Email dated 30.11.2025 from the Dean , Faculty of Technology
4. Letters of even number dated 13/04/2026 & 07/07/2026
5. Email dated 13/08/2026 from the Coordinator, ITEC Palayad
6. Orders of the Vice Chancellor in the file of even no. dated 22.8.2026

ORDER

1. The proposal to start the Five Year Integrated Programme in Master of Computer Applications at the Information Technology Education Centre (ITEC), Dr. Janaki Ammal Campus, Palayad, with effect from the 2025 admission, was approved vide paper read (1) above.

2. The Scheme and Syllabus of the Five Year Integrated Programme in Master of Computer Applications (First and Second Semester only) were approved and implemented with effect from the 2025 admission, vide paper read (2) above.

3. The Scheme and Syllabus (First to Sixth Semesters only) of the above programme, submitted by the Head, Department of Information Technology were forwarded to the Dean, Faculty of Technology, and the Dean, vide paper read (3) above, submitted the remarks "Verified" with a direction to forward the complete syllabus of the Programme for verification.

4. In tune with the remarks of the Dean, Faculty of Technology, vide paper read (4) above, Head of the Department, Department of Information Technology was requested to submit the complete syllabus (all 10 semesters) of the Programme so as to forward the same to the Dean concerned for verification.

5. As per paper read (5) above, Coordinator, ITEC Palayad once again forwarded the syllabus of the Programme (first six semesters only) for approval as the End Semester Examination of Third Semester had been scheduled on October 2026.

6. The Vice Chancellor, after considering the exigency of the matter 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, approved the Scheme and Syllabus of the Five Year Integrated Programme in Master of Computer Applications (Third to Six Semesters

only) and accorded sanction to implement the same at the Information Technology Education Centre (ITEC), Dr. Janaki Ammal Campus, Palayad, with effect from the 2025 admission, subject to report to the Academic Council.

7. Orders are issued accordingly.

Approved Scheme and Syllabus (First to Six Semesters only) of the Five Year Integrated Master of Computer Applications Programme at the Information Technology Education Centre (ITEC), Dr. Janaki Ammal Campus, implemented with effect from the 2025 admission, are appended to this University Order and uploaded in the University website (www. kannuruniv.ac.in).

Sd/-
Bindu K P G
Joint Registrar (Acad)
For REGISTRAR

To:
1. Head, Dept.of Information Technology ,
2. Controller of Examinations (Through PA to CE)
3. Nodal Officer, FYIMP

Copy To:
1. PA to CE (to circulate among the sections concerned in the Examination Branch)
2. PS to VC, PA to R, PA to CE
3. JR II (Exam)
4. EP IV/EP VI/ EG I/EXC I Sections (Exam)
5. Web manager (to publish in the website)
6. Computer Programmer
7. SF/DF/FC

Forwarded / By Order
SECTION OFFICER

8

SCHOOL OF INFORMATION SCIENCE &TECHNOLOGY

INFORMATION TECHNOLOGY EDUCATION CENTRE

(Extension centre –Janaki Ammal Campus Palayad)

KANNUR UNIVERSITY

DEGREE OF

FIVE YEAR INTEGRATED PROGRAMME IN

MASTER OF COMPUTER APPLICATIONS

B.C.A leading to B.C.A (Honours) & M.C.A

(Choice Based Credit System)

SCHEME& SYLLABUS

(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2024 – 25 ONWARDS)

SCHEME AND SYLLABUS FOR THE DEGREE OF

FIVE YEAR INTEGRATED PROGRAMME IN MASTER OF COMPUTER APPLICATIONS

WITH MULTIPLE PATHWAYS

FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2025 – 26 ONWARDS

1 PROGRAMME SPECIFIC OUTCOMES

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications
PSO2	Enhance the knowledge and skills about System Software and Application Software
PSO3	Attain skills to design Algorithms and Programs
PSO4	Acquire the knowledge in the building, designing and managing IT and IT enabled infrastructure
PSO5	Design, build, and test software systems to meet the given specifications by following the principles of Software Engineering

DEFINITIONS

- Department / School is the department / school established in the Kannur University as per the statute of Kannur University.
- Information Technology Education Center is an extension center of Department /School of Information Technology.
- Academic programs offered through this extension center is fully controlled and monitored by the Department/School of Information Technology.
- All the programs offered through this extension center is approved by the council of Department/School of Information Technology.

- Head of the Department/one of the senior faculty appointed by the Vice Chancellor shall monitor all the activities conducted through this extension center. He/ She shall the responsibility all academic activities in the extension center and also regularly evaluate the status of all faculties and academic curriculums in the center.
- Academic programme is an entire course of study comprising the details such as the programme structure, course details, and evaluation schemes designed to be taught and evaluated in the teaching department / IT education center or jointly under more than one such department / center.
- Course is a segment of a programme limited to one semester subject.
- Programme Structure is a list of courses (Discipline specific Core Courses (DSC), Discipline specific Elective Courses (DSE), Ability Enhancement Courses (AEC) , Value Added Courses (VAC), Skill Enhancement Courses (SEC), and MOOC) that makes up an academic programme, specifying its details such as the syllabus, credits, hours of teaching, evaluation and examination schemes, and the minimum number of credits required for successfulcompletion of the program prepared in conformity with theNEP 2020/AICTE/regulations-2024 of Kannur University.
- Discipline Specific Core (DSC) is a course that is to be compulsorily completed by a student admitted to a particular programme to receive the degree and which cannot be substituted by any other course.
- Discipline Specific Elective (DSE) is an optional course to be selected by a student out of such courses offered in the same department or other departments.
- Wherever in this document “the University” is mentioned, it should be taken as the Kannur University and wherever “the Department” is mentioned, it should be taken as the School of Information Technology/Information Technology Education Centerwhere the respective course is offered.

2 SCOPE

1) This regulation shall apply to Five Year Integrated Programme in Master of Computer Applications with multiple pathways, conductedeither by school of information technology or Information technology Education Centre working as an extension centre,Kannur University.

2) Choice Based Credit Semester System presupposes academic autonomy, cafeteria approach in academic environment, semester system, course credits, alphabetical grading. The course may be either offered by the Department of Information Technology, Mangattuparamba Campus or an extension centre setup by the university in other campuses. In such case the programme offered is strictly under the monitoring of the Department of Information Technology and all course related activities are done by the extension centre, and all these activities are monitored by the senior faculty of the department deputed by the Hon. Vice Chancellor of the University.

3) The Name of degree awarded when a student take exit at 3rd /4th/5th year is specified in the different pathways mentioned as part of this regulation. "*The Name "Five Year Integrated Programme in Master of Computer Application" is only for indicating the five year span of the program according to NEP2020/AICTE/Kannur University regulation"*.

4) Students enrolled in the Five-Year Integrated Programme in Master of Computer Applications (MCA) at the Information Technology Education Centre (ITEC), Janaki Ammal Campus, Thalassery shall complete the first three years of their study at ITEC, Janaki Ammal Campus, Thalassery Campus. On successful completion of the third year, the student has the option to exit the program with a BCA degree or continue to the fourth year for a BCA (Honours) with an exit /and to proceed fifth year to complete the MCA at the Department of Information Technology, Mangattuparamba Campus.

3 ADMISSION

1) Admission to the Five Year Integrated Programme in Master of Computer Applications (Bachelor degree of Computer Application (BCA) leading to BCA (Honours) and MCA with multiple pathways with different specialization) will be done as the regulations prescribed by the university from time to time.

Eligibility for Admission:

Basic qualifications: Candidate must possess

1) Pass with 50% marks/equivalent grade at 10+2 level or equivalent

AND

2) Mathematics/statistics/computerscience/computer application/informatics practices as one of the subjects at 10+2 level.

Admission to the Five-Year Integrated Programme in Master of Computer Applicationsprogrammeshall be made purely based on the entrance examination conducted by the university. Notification in this regard shall be made well in advance. If the number of candidates admitted based on the entrance examination is less than the sanctioned strength, the department can fill the vacancy by advertising the vacancy through press releases. Admission to these seats should be granted based on the marks obtained by the candidates in the qualifying examination. Reservation norms followed by the university should be adhered to in the admission process.

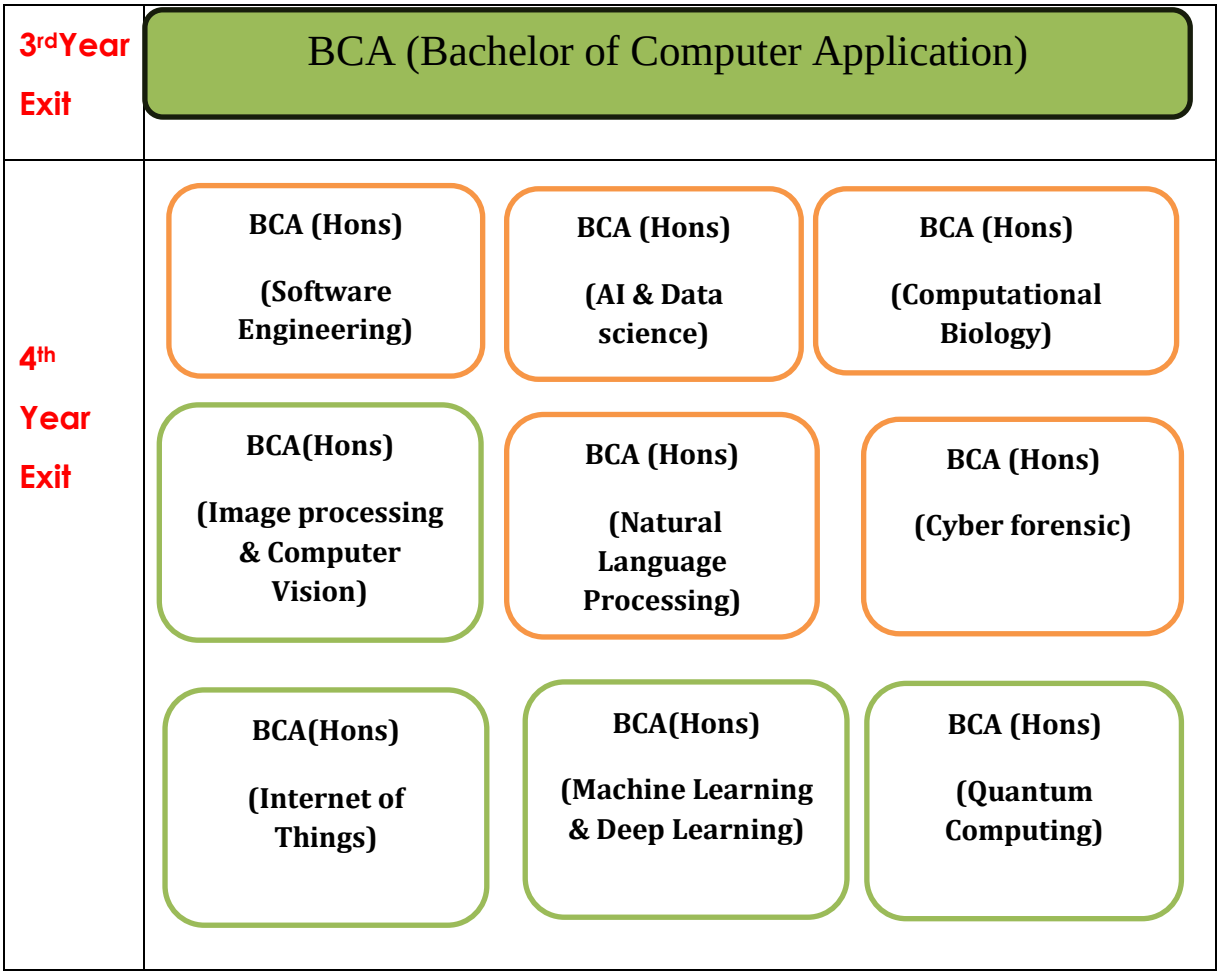
4 REGISTRATION

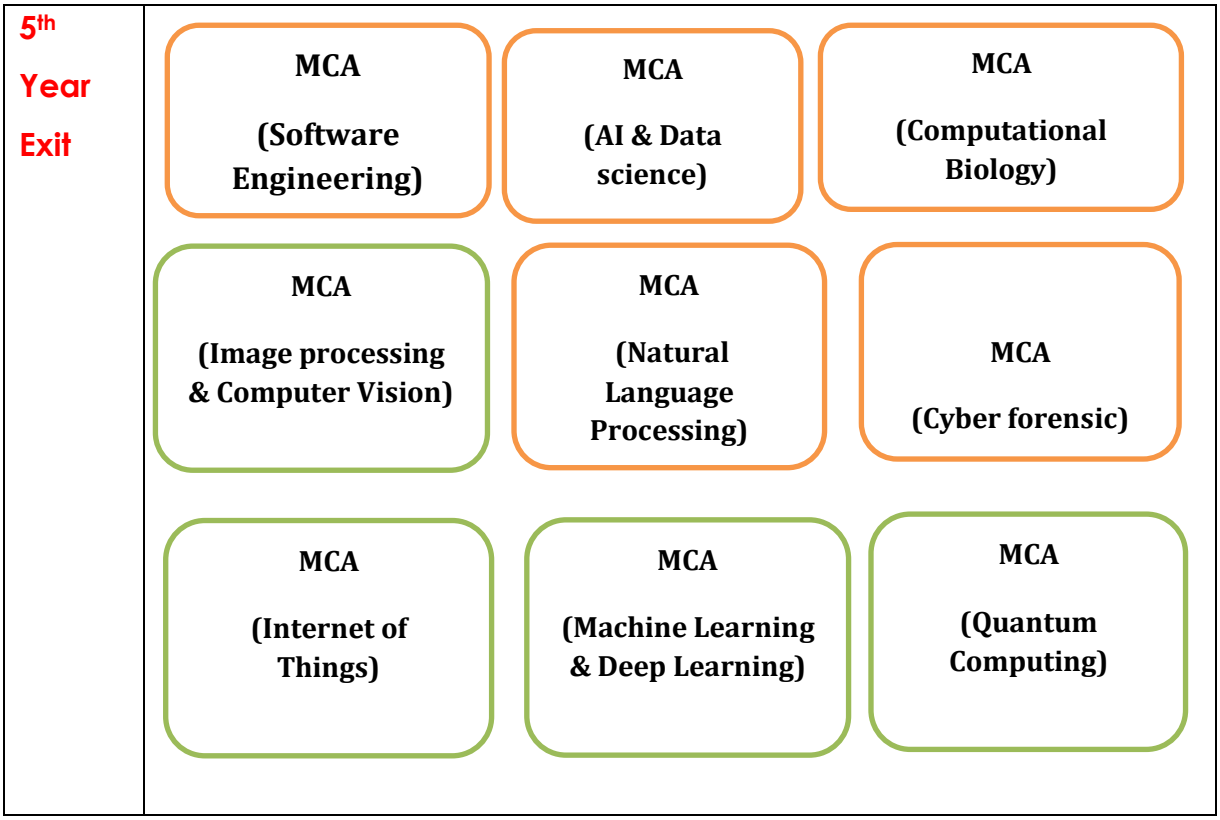
Since students have flexibility to choose various courses and pathways, every student should register for a course for the next semester by the end of the previous semester (except the first semester). Any changes in the selection of the courses are permitted up to the 5 working days of the start date of the new semester. The offering department can decide on the minimum / maximum number of

registrations for a course depending on the instructional facilities and teachers' availability for the listed courses. While choosing the courses students are advised to discuss with semester coordinator/ faculty advisor and also ensure that, the credit requirement for awarding the degree in various pathways as per FYIMP-2024 regulations are satisfied. Student can register for additional courses including MOOCs from second semester onwards, provided the total number of courses registered for that semester is not more than 30.

5 PROGRAMME STRUCTURE

1) Duration of the Five Year Integrated Program in Master of Computer Applicationsprogram shall be 5 years, divided into 10 semesters. Each semester shall have 18 weeks. Students can take exit at 3rd / 4th /5th year as mentioned in FYIMP regulations-2024-25. The Name of the degree awarded to each student will be depending on the pathway courses opted by the students while taking exit at above mentioned years. Different Pathways and corresponding names of degree awarded is mentioned in the below diagram.





2) Minimum and maximum duration for completing the courses will be based on the FYIMP-2024 regulations of the University.

3) Degree may be awarded after securing the minimum required credits for the programmeafter the 3rd / 4th or 5th year exit as Bachelor Degree in Computer Applications (BCA)/ Bachelor Degree in Computer Applications in Honswith respective specialization or Master of Computer Applications with respective specialization chosen by the students according to the pathway mentioned in the regulations based on the FYIMP 2024 regulations/AICTE norms.

6 MOOC

In addition to the courses specified as part of the programme, all students should mandatorily complete 2 MOOC courses of at least 3 credits to complete the requirements of getting BCA (Hons) degree in any of the specialization. The credits earned also will be considered for the computation of GPA and CGPA.

7 EVALUATION

The Continuous and Comprehensive Evaluation (CCE) and the End semester Evaluation (ESE) will be conducted as per the Kannur University FYIMP/AICTE Regulations for all theory and practical courses unless it is specifically mentioned in the respective courses. The ratio of Continuous and Comprehensive Evaluation and End semester Evaluationshall be 50:50.

End Semester Evaluation

End semester Evaluation will be conducted as per FYIMP Regulations-2024.

The courses offered under these programs have either 3 credits or 4 credits courses such as, DSC, DSE,AEC,SEC,VAC,MOOC and Internship. The credit for a given course is typically distributed as follows

Distribution Teaching / Learning hours for 4 Credit Course

1. 4 Credit Lecture session
2. 3 Credit Lecture Sessions and 1 Credit Practical session
3. 2 Credit Lecture Session and 2 Credit Practical session
4. 1 Credit Lecture Session and 3 Credit Practical session
5. 4 Credit Practical Sessions

Similarly, Distribution Teaching / Learning hours for 3 Credit Course

1. 3 Credit Lecture session
2. 2 Credit Lecture Sessionand 1 Credit Practical Session
3. 1 Credit Lecture Sessionand 2 Credit Practical Session
4. 3 Credit Practical Session

End Semester Examination for all these different combinations are as Follows

ESE Scheme for 4 Credit courses		
1	4Credit Theory	Theory Examination of 50 marks. Duration of Examination 2 Hours
2	3 credit Lecture and 1 Credit Practical	ESE(Theory) Maximum marks: 50 marks Duration:2 hours. The obtained Marks will be scaled to 37.5 ESE(Practical) Maximum marks: 50 marks. Duration: 3 hours. Marks obtained will be scale down to 12.5 Total marks obtained for the course will be the sum of scaled ESE marks obtained for theory and practical. i.e. 50 marks
3	2 credit Lecture and 2 Credit Practical	ESE(Theory) Maximum marks: 50 marks Duration:2 hours. The obtained Marks will be scaled to 25 ESE(Practical) Maximum marks: 50 marks. Duration: 3 hours. Marks obtained will be scale down to 25 Total marks obtained for the course will be the sum of scaled ESE marks obtained for theory and practical. i.e. 50 marks
	1 credit Lecture and 3 Credit Practical	ESE(Theory) Maximum marks: 50 marks Duration:2 hours. The obtained Marks will be scaled to 12.5 ESE(Practical) Maximum marks: 50 marks. Duration: 3 hours. Marks obtained will be scale down to 37.5 Total marks obtained for the course will be the sum of scaled ESE

		marks obtained for theory and practical. i.e. 50 marks
	4 Credit Practical	Practical Examination of 50 marks. 3 hours Duration
ESE Scheme for 3 Credit courses (SEC/MDC/VAC)		
	3 Credit Theory	ESE will be conducted for 50 marks
	2 Credit theory and 1 Credit Practical	ESE (Theory) Max marks: 50; Duration: 2 Hours scaled to 30 marks ESE (Practical) Max marks: 50; Duration 2 Hours scaled to 20 marks Total marks obtained for the course will be the sum of scaled ESE marks obtained for theory and practical. i.e. 50 marks
	1 Credit for Theory and 2 Credit for Practical	ESE (Theory) Max marks: 50; Duration: 2 Hours scaled to 20 marks ESE (Practical) Max marks: 50; Duration 2 Hours scaled to 30 marks Total marks obtained for the course will be the sum of scaled ESE marks obtained for theory and practical. i.e. 50 marks
	3 Credit Practical	ESE will be conducted for 50 marks

If any changes in the evaluation patten for any courses it can be mentioned in the assessment rubrics of the corresponding course. Otherwise it will be followed the above rubrics corresponding to their credit.

Continuous and Comprehensive Evaluation (CCE).

CCE for all courses shall follow the FYIMP- Computational Science -Scheme and syllabus -2024 unless it is specifically mentioned in the course syllabus. Where ever there is a practical component associated with a course; Lab test will be conducted as part of CCE, in addition to the other evaluation methods mentioned in FYIMP-Regulations 2024 for CCE. The division of CCE marks for different courses structure shall be as follows.

Course structure	Continuous Assessment
4 Credit course	**Total Marks: 50** Test paper 1: 10 Test Paper 2: 10 Attendance:10 Assignment: 10 Case study/Seminar /Viva: 10
4 Credit course (3 Lecture+ 1 Lab)	**Total Marks: 50** Test 1: 10 Test 2: 10 Attendance: 10

	Lab Test and viva : 10 Record: 05 Assignments/casestudy/ seminar: 05
4 Credit (2 Credit Lecture and + 2 Credit Practical)	**Total Marks: 50** Test 1: 10 Test 2: 10 Attendance:10 Lab Test and viva: 10 Record: 05 Assignments/ case study/ seminar: 05
4 Credit (1 Credit Lecture + 3 Credit Practical)	**Total Marks 50** Test paper 1: 05 Test Paper 2: 05 Attendance: 05 Lab test 1 and viva: 10 Lab Test 2 and viva: 10 Lab record: 5 Seminar/ Assignments / Case Study: 10
4 Credit Practical	Total Marks Lab test 1 and viva:15 Lab test 2 and viva:15 Attendance :10 Record: 10
3 Credit Courses (MDC/SEC/VAC) offered by the IT/Maths/Statistics	
3 credit Lecture	**Total Marks 50** Test 1: 15 Test 2: 15 Attendance:10 Seminar /Assignment/ Case Study: 10
2 Credit Lecture + 1 Credit practical	**Total Marks 50** Test 1: 10 Test 2: 10 Lab test:10

	Attendance:10 Record:05 Seminar/Assignment / Case study:05
1 Credit Lecture and 2 Credit Practical	**Total Marks 50** Test 1: 05 Test 2: 05 Attendance:10 Lab test with Viva: 20 Record: 5 Assignment / Case study/ Seminar: 05
3 Credit Lab	**Total Marks 50** Lab test 1 and viva:15 Lab test 2 and viva:15 Attendance:10 Record: 10

Theend semester practical examination will be conducted by a board of examiners constituted by the Head of the department/Assistant Director of the extension centre where the course is offered. The board of examiners consist of 2 faculty members and one should be the faculty in charge of the respective course. Total minimum requirement for every CCE component for a pass shall follow the minimum requirement mentioned in FYIMP-2024 regulations. If any other mode of assessment is proposed for a specific course; the assessment rubrics should be mentioned in thesyllabus separately associated with that course. Otherwise, evaluation scheme mentioned in the regulations shall be applicable for both CCE and ESE.

Evaluation criteria for the Research, Internship components will follow the criteria mentioned in FYIMP-2024 regulations.

8 COMPLIANCES WITH FYIMP- REGULATIONS -2024

In general, FYIMP – Five Year Integrated Programme in Master of Computer Applications follows the FYIMP-regulations approved by university/AICTE norms. Department council have the sole right to revise the scheme and syllabus based on the feedback and input from various stake holders of the program.

CREDIT DISTRIBUTION AND EXIT PATHWAYS

Semester	Major/Minor DSC	Major DSE	AEC	SEC	MDC	VAC	Internship/Project	MOOC	Total	
1	12		06		03				21	
2	16		03		03				22	
3	16				03	03			22	
4	12	04		03		03			22	
5	16	04		03					23	
6	12	04		03			04		23	
Total BCA	**84**	**12**	**09**	**09**	**09**	**06**	**04**		**133**	**3rd Year Exit**
7	12	08						03	23	
8	04	08					08	03	23	
Total BCA(Hon)	**100**	**28**	**09**	**09**	**09**	**06**	**12**	**06**	**179**	**4th Year Exit**
9	04	12		03						
10							20			
Total MCA	**104**	**40**	**09**	**12**	**09**	**06**	**32**	**06**	**218**	**6th Year Exit**

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(BCA LEADING TO BCA WITH HONS & MCA)

Semester I

No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESE	CCE	T
1.1	100	KU01DSCMCA101	Introduction to Computational Informatics	60	4	4	0	1	50	50	100
1.2	100	KU01DSCMCA102	Principles of Programming	90	4	2	4	2	50	50	100
1.3	100	KU01DSCMCA103	Mathematical foundations for computer application	60	4	4	0	1	50	50	100
1.4	100		MDC-1-	60	3	2	2	1	50	50	100
1.5	100		AEC 1- English 1	45	3	3	0	1	50	50	100
1.6	100		AEC 2-English 2	45	3	3	0	0	50	50	100
				360	21	18	6	6	300	300	600

Semester II

No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESE	CCE	T
2.1	100	KU02DSCMCA104	Object Oriented Programming using C++	90	4	2	4	1	50	50	100
2.2	100	KU02DSCMCA105	Digital Electronics and Computer Organization	60	4	4	0	1	50	50	100
2.3	100	KU02DSCMCA106	Statistical Foundations for Computer Applications	60	4	4	0		50	50	100
2.4	100	KU02DSCMCA107	Database Management Systems	90	4	2	4	1	50	50	100
2.5	100		MDC-2	60	3	2	2		50	50	100
2.6	100		AEC-3 Additional Language	45	3	3	0		50	50	100
				390	22	17	10	03	300	300	600

Semester III

No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
3.1	200	KU03DSCMCA201	Java Programming	90	4	2	4	1	50	50	100
3.2	200	KU03DSCMCA202	Introduction to Data Structure	90	4	2	4	0	50	50	100
3.3	200	KU03DSCMCA203	Web Technology	90	4	2	4	0	50	50	100
3.4	200	KU03DSCMCA204	Data communication and computer Networks	60	4	4	0	0	50	50	100
3.5	200		MDC-3	60	3	2	2	0	50	50	100
3.6	200		VAC-1	45	3	3	0	0	50	50	100
				375	22	15	14	1	300	300	600

Semester IV

No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
4.1	200	KU04DSCMCA205	System Software and Operating System	60	4	4	0	1	50	50	100
4.2	200	KU04DSCMCA206	Software Engineering	60	4	4	0	1	50	50	100
4.3	200	KU04DSCMCA207	Artificial Intelligence	90	4	2	4	1	50	50	100
4.4	200	KU04DSEMCA20.x	DSE-1*	60	4	4	0	1	50	50	100
4.5	200		SEC-1	60	3	2	2	1	50	50	100
4.6	200		VAC-2	45	3	3	0	0	50	50	100
				365	22	19	6	5	300	300	600

	S4 - List of Discipline Specific Electives (DSE-1)											
No	**Level**	**Course Code**	**Course Name**	**Total Hours**	**C**	**Hrs./wk.**			**Assessment Weightage (%)**		
						L	**P**	**Tt**	**ESA**	**CE**	**T**
4E.1	200	KU04DSEMCA208	Python Programming	60	4	4	0	1	50	50	100
4E.2	200	KU04DSEMCA209	Discrete Mathematics	60	4	4	0	1	50	50	100
4E.3	200	KU04DSEMCA210	Mobile Computing	60	4	4	0	1	50	50	100
4E.4	200	KU04DSEMCA211	Operating System Security	60	4	4	0	1	50	50	100
4E.5	200	KU04DSEMCA212	Design Thinking	60	4	4	0	1	50	50	100

Semester V

No	**Level**	**Course Code**	**Course Name**	**Total Hours**	**C**	**Hrs./wk.**			**Assessment Weightage (%)**		
						L	**P**	**Tt**	**ESA**	**CE**	**T**
5.1	300	KU05DSCMCA301	Formal Language & Automate Theory	60	4	4	0	1	50	50	100
5.2	300	KU05DSCMCA302	Analysis and Design of Algorithms	60	4	4	0	1	50	50	100
5.3	300	KU05DSCMCA303	Introduction to IoT	75	4	3	2	1	50	50	100
5.4	300	KU05DSCMCA304	Machine Learning Techniques	75	4	3	2	1	50	50	100
5.5	300	KU05DSEMCA30.x	DSE-2 *	60	4	4	0	0	50	50	100
5.6	300	KU05SECMCA30.x	SEC-2	60	3	2	2	0	50	50	100
				390	23	20	6	4	300	300	600

S5 - List of Discipline Specific Electives (DSE)										
No	**Level**	**Course Code**	**Course Name**	**C**	**Hrs./wk.**			**Assessment Weightage (%)**		
					L	**P**	**Tt**	**ESA**	**CE**	**T**
5E.1	300	KU05DSEMCA305	Computer Graphics	4	3	2	1	50	50	100
5E.2	300	KU05DSEMCA306	Data and Business Analytics	4	3	2	1	50	50	100
5E.3	300	KU05DSEMCA307	Wearable Computing and sensors	4	3	2	1	50	50	100
5E.4	300	KU05DSEMCA308	High Performance Computing	4	4	0	1	50	50	100
5	300	KU05DSEMCA309	Game Development	4	3	2	1	50	50	100

Semester VI

No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
6.1	300	KU06DSCMCA305	Big Data Analytics	60	4	4	0	1	50	50	100
6.2	300	KU06DSCMCA306	Quantum computing	60	4	4	0	0	50	50	100
6.3	300	KU06DSCMCA307	Generative AI	60	4	4	0	0	50	50	100
6.4	300	KU06DSEMCA310	DSE-3- Technology Specific Elective	60	4	4	0	0	0	100	100
6.5	300	KU05SECMCA30.x	SEC-3	60	3	2	2	0	50	50	100
6.6	300	KU06INTMCA301	Internship	60	4	-	9	0	50	50	100
					23	18	11	1	250	350	600

***DSE-3 Technology Specific Elective is meant to foster the students with tools and technologies that they need to know and make use in the design and development of software applications. Seminar Report / Case study report of a specific technology should be submitted by each student for the evaluation. The mode of evaluation of this course shall be based on the presentation, report and viva. It is evaluation is completely under CE component.**

S6 - List of Discipline Specific Electives (DSE-3)

No	Level	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
					L	P	Tt	ESA	CE	T
6E.1	300	KU06DSEMCA314	Information Security	4	4	0	1	50	50	100
6E.2	300	KU06DSEMCA315	Data and Information visualization	4	4	0	1	50	50	100
6E.3	300	KU06DSEMCA316	Virtual and Augmented Reality	4	4	0	1	50	50	100
6E.4	300	KU06DSEMCA317	Game Development	4	4	0	1	50	50	100
6E.5	300	KU06DSEMCA318	Computer Vision	4	4	0	1	50	50	100
6E.6	300	KU06DSEMCA319	Computational Photography	4	4	0	1	50	50	100
6E.7	300	KU06DSEMCA320	High performance computing	4	4	0	1	50	50	100
6E.8	300	KU06DSEMCA321	Green Computing	4	4	0	1	50	50	100

SKILL ENHANCEMENT COURSES (SEC)

	Semester 4 SEC POOL 1										
No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
4SE.1	200	KU04SECMCA201	Python Programming	60	3	2	2	0	50	50	100
4SE.2	200	KU04SECMCA202	Fundamentals of Digital Skilling using Google Workspace for Education	60	3	2	2	0	50	50	100
4SE.3	200	KU04SECMCA203	Web Technologies	60	3	2	2	0	50	50	100
4SE.4	200	KU04SECMCA204	R Programming	60	3	2	2	0	50	50	100
4SE.5	200	KU04SECMCA205	Digital Marketing	60	3	2	2	0	50	50	100

	Semester 5 SEC POOL 2										
No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
5SE.1	300	KU05SECMCA301	The Art of E - Documentation using Latex	60	3	2	2	0	50	50	100
5SE.2	300	KU05SECMCA302	Data Processing with Python	60	3	2	2	0	50	50	100
5SE.3	300	KU05SECMCA303	Data Science Fundamentals	60	3	2	2	0	50	50	100
5SE.4	300	KU05SECMCA305	Swift Programming	60	3	2	2	0	50	50	100
5SE.5	300	KU05SECMCA306	Application Development using React JS	60	3	2	2	0	50	50	100
5SE.6	300	KU05SECMCA307	Application Development using Flutter	60	3	3	3	0	50	50	100

	Semester 6 SEC POOL 3										
No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
6SE.1	300	KU06SECMCA308	Artificial Intelligence in Daily Life	60	3	2	2	0	50	50	100
6SE.2	300	KU06SECMCA309	Fundamentals of Big Data	60	3	2	2	0	50	50	100
6SE.3	300	KU06SECMCA310	Optimization Techniques	60	3	2	2	0	50	50	100
6SE.4	300	KU06SECMCA311	Web Application Security and Ethical Hacking	60	3	2	2	0	50	50	100

6SE.5	300	KU06SECMCA312	Mobile Application Security and Privacy	60	3	2	2	0	50	50	100

MULTI DISCIPLINARY COURSES

Semester 1/Semester 2 MDC POOL 2											
No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
1MD.1	100	KU01MDCMCA101	Introduction to Computational Informatics	60	3	2	2	0	50	50	100
1MD.2	100	KU01MDCMCA102	Principles of Programming	60	3	2	2	0	50	50	100
2MD.1	100	KU02MDCMCA201	Foundations of Data Science	60	3	2	2	0	50	50	100
2MD.2	100	KU02MDCMCA202	Object Oriented Programming using C++	60	3	2	2	0	50	50	100
3MD.1	200	KU03MDCMCA301	Cyber Forensics	60	3	2	2	0	50	50	100
3MD.2	200	KU03MDCMCA302	Scientific Computing	60	3	2	2	0	50	50	100

MDC (Semester 1)

KU01MDCMCA101INTRODUCTION TO COMPUTATIONAL INFORMATICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
1	MDC	100	KU01MDCMCA101	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	1	50	50	100	2

Course Description: This course provides a comprehensive overview of computing, covering historical milestones, hardware components, software systems, and computational thinking principles. Students will explore the evolution of computing systems, from early pioneers to modern processors and quantum units. The curriculum delves into hardware intricacies, software distinctions, and essential concepts in computer science, emphasizing problem-solving skills and algorithmic

thinking. Practical aspects include hands-on experiences with hardware assembling, operating system installation, algorithm and flowchart visualization.

Course Objectives:

- To become aware about the evolution and generation computer system and familiarises various components of computer systems
- To become familiar with various computer hardware components
- To understand different software components and its types and the installation and configuration of operating systems

  selection
- To familiarise different methodologies of problem solving using the computational thinking concept.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Develop a foundational knowledge of computing systems, encompassing their historical development, evolutionary milestones, and the notable contributions of key figures in the field.
CO2	Acquire familiarity with diverse hardware components constituting a computer system.
CO3	Attain the knowledge of computer software and the activities focused on the installation and configuration of diverse hardware components within a computer system including operating systems
CO4	Develop a foundational understanding of computer science as a discipline, examining problems through the lens of computational thinking and inducing the analytical skills to address challenges in the field.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	
CO2	✔	✔	✔	✔	

CO3	✔	✔	✔	✔	
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT1	History and Evolution of Computers, Generations of Computers, Characteristics and Applications of Computers, Basic Computer Organization (Von Neumann Model), Number Systems (Binary, Decimal, Octal, Hexadecimal), Contributions of Computing Pioneers(Charles Babbage, John von Neumann, Alan Turing, John McCarthy). Modern Computing Devices. Single-Core and Multi-Core ProcessorsGPU, TPU (Introduction)
UNIT2	Basic Electronic Components:DiodeTransistorIntegrated Circuit (IC), Motherboard and its Components: CPU and Cooling FanRAM, ROM and Cache Memory,Storage Devices: Hard Disk (HDD) , Solid State Drive (SSD), NVMe SSD, Expansion Slots (PCIe), Input and Output Ports:USB, HDMI, VGA, Ethernet,Input and Output Device
First series examination including theory + Laboratory if any	
UNIT3	Introduction to Software: System Software and Application Software,Operating System: Functions and Types, Proprietary and Open Source Operating Systems,Booting Process, BIOS and UEFI, Bootable Devices, Disk Partitioning (Basic Concepts),File Systems: FAT, NTFS, ext4,Device Drivers and their Functions
UNIT4	Introduction to Problem Solving: Computational Thinking, Problem Decomposition, Pattern Recognition, Abstraction, Logical Thinking, Introduction to Algorithms, Characteristics of a Good Algorithm, Flowcharts, Flowchart Symbols, Introduction to Raptor, Solving Simple Problems using Algorithms and Flowcharts
Second series internal examination including theory + Laboratory if any	

Books/References

1. Gary B. Shelly, Thomas J. Cashman, and Misty E. Vermaat. “Introduction to Computers”, Cengage Learning, 2008.
2. Pradeep K. Sinha and Priti Sinha, Computer Fundamentals: Concepts, Systems & Applications. BPB Publications.
3. Kevin Wilson, Computer Hardware: The Illustrated Guide to Understanding Computer Hardware. Amazon Digital Services LLC – KDP, 2018.
4. John Hanna, OS Installation 101: A Step-by-Step Approach for Newbies.
5. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC, 2014.
6. R.G. Dromey, How to solve it by Computer, PHI, 2008.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7.

KU01MDCMCA202 PRINCIPLES OF PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
1	MDC	100	KU01MDCMCA202	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	1	50	50	100	2

*** ESE duration: 2 hours for theory and 2 hours for Lab**

Course Description: Computer Science is all about developing correct and efficient solutions for our day-to-day problems. The process of developing solutions is not centered on learning a programming language and doing coding straight away. Instead, a blueprint of the proposed solution should be outlined and tested for correctness. Once a proposed blueprint leads to a correct solution, it can be implemented using a suitable programming language. The objective of this course is to impart knowledge to the learner about building the blueprint of a solution. Learners are also exposed to implementing the solutions using the C programming language.

Course Objectives:

- To impart knowledge about various constructs for developing solutions
- To become familiar with using the various constructs to develop solutions
- To compare and contrast various constructs for solution development for selection
- To compare and contrast various constructs for solution development for

iteration

- To implement solutions using C programming language

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Illustrate the foundations of developing solutions using flowcharts and algorithms
CO2	Develop solutions using various selection constructs and implement them in the C programming language
CO3	Develop solutions using various iteration constructs and implement them in the C programming language
CO4	Understand advanced concepts in direct memory handling, file handling and functions.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Introduction to Computers and Programming Languages, Problem Solving Techniques Algorithms and Flowcharts, Features and Applications of C, Structure of a C Program

	Character Set, Tokens, Keywords, Identifiers, Variables and Constants, Data Types, Input and Output Functions (printf(), scanf()), Operators and Expressions
UNIT 2	Conditional Statements, Looping Statements , Nested Loops,break, continue, goto, Functions, Function Declaration, Definition and Call, Parameter Passing.
First series internal examination including theory + Laboratory if any	
UNIT 3	One-Dimensional Arrays, Two-Dimensional Arrays, Character Arrays, String Handling Functions, User-defined String Functions, Introduction to Pointers
UNIT 4	Structures, Unions, Enumerated Data Types, Typedef, File Handling, Text Files and Binary Files File Opening and Closing, Reading and Writing Files, Command Line Arguments (Introduction)
Second series internal examination including theory + Laboratory if any	

Books/References

1. J.B Dixit, Computer Fundamentals and Programming in C, Firewall Media
2. E Balagruswamy, Programming in ANSI C, TMH, 5th Edition
3. Kanetkar, Let Us C, BPB Publications, 8th Edition
4. Ashok N. Kamthane, Programming with ANSI and Turbo C, Pearson Education.
5. Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, 2nd Edition, Pearson.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7.

KU02MDCMCA201Foundations of Data Science

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
2	MDC	100	KU02MDCMCA201	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	1	50	50	100	2

Course Description

This course introduces the fundamental concepts, principles, and techniques of Data Science. Students will learn how data is collected, processed, analyzed, visualized, and interpreted to support decision-making. The course covers data preprocessing, exploratory data analysis, statistical foundations, data visualization, and introductory machine learning concepts using modern tools such as Python and its data science libraries. It provides a strong foundation for advanced courses in Artificial Intelligence, Machine Learning, Big Data Analytics, and Business Intelligence.

.Course Objectives:

After completing this course, students will be able to:

- Understand the fundamentals and applications of Data Science.
- Learn methods of data collection, preprocessing, and cleaning.
- Apply basic statistical techniques for data analysis.
- Create meaningful visualizations for effective communication of data.
- Understand the basics of predictive analytics and machine learning.
- Develop practical skills using Python libraries for data analysis.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the concepts, lifecycle, and applications of Data Science.
CO2	Collect, preprocess, clean, and transform datasets for analysis.
CO3	Perform exploratory data analysis using statistical techniques and data visualization
CO4	Apply basic machine learning algorithms for simple predictive problems.

Mapping of COs to PSOs

CO - PSO Mapping

	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	
CO2	✔	✔	✔	✔	
CO3	✔	✔	✔	✔	
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT1 15 Hours	Introduction to Data Science ,Data Science Life Cycle, Data Science Process, Applications of Data Science ,Types of Data ,Structured, Semi-Structured and Unstructured Data , Sources of Data ,Big Data Overview . Data Science Ecosystem ,Roles of Data Scientist, Data Analyst and Data Engineer ,Introduction to Python for Data Science , Installing Anaconda, Jupyter Notebook ,Basic Python Review
UNIT2 15 Hours	Data Acquisition Methods,Data Formats (CSV, Excel, JSON), Data Cleaning,Handling Missing Values, Removing Duplicate Records ,Data Transformation, Feature Scaling, Data Integration, Introduction to NumPy, Introduction to Pandas, DataFrames and Series. Data Selection and Filtering, Descriptive Statistics, Exploratory Data Analysis (EDA)
First series examination including theory + Laboratory if any	
UNIT3 15 Hours	Importance of Data Visualization, Data Visualization Principles, Matplotlib, Seaborn, Line Chart, Bar Chart, Histogram. Scatter Plot, Pie Chart, Box Plot, Correlation Analysis, Measures of Central Tendency, Mean, Median, Mode, Measures of Dispersion, Variance and Standard Deviation Probability Basics
UNIT4 15 Hours	Introduction to Machine Learning, ypes of Machine Learning, Supervised Learning, Unsupervised Learning, Training and Testing Data,Classification and Regression,k-Nearest Neighbour (KNN),Decision Tree (Concept Only), Clustering using K-Means (Concept Only), Model Evaluation, Accuracy, Precision, Recall and F1-Score, Ethical Issues in Data Science, Privacy and Data Security,Real-world Case Studies, Mini Project using Python
Second series internal examination including theory + Laboratory if any	

Books/References

Joel Grus, Data Science from Scratch: First Principles with Python, 2nd Edition, O'Reilly Media, 2019.

Cathy O'Neil and Rachel Schutt, Doing Data Science, O'Reilly Media, 2013.

Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016.

Wes McKinney, Python for Data Analysis, 3rd Edition, O'Reilly Media, 2022.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, 2nd Edition, Springer, 2009.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor, An Introduction to Statistical Learning with Applications in Python, Springer, 2023.

Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras&TensorFlow, 3rd Edition, O'Reilly Media, 2022.

Nina Zumel and John Mount, Practical Data Science with R, Manning Publications, 2014. (Useful for concepts even if Python is the primary language.)

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7.

KU02MDCMCA202: OBJECT ORIENTED PROGRAMMING USING C++

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
2	MDC	100	KU02MDCMCA202	3	60

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

2	2	1	50	50	100	2

*** ESE Duration: 2 hours for theory and 2 hours for Lab**

Course Description:

This course provides an introduction to the principles and practices of object-oriented programming (OOP) using the C++ programming language. Students will learn fundamental concepts of OOP and how to apply them to develop software solutions using C++.

Course Objectives:

- Understand the principles of Object-Oriented Programming (OOP) using C++
- Master the syntax and semantics of the C++ programming language
- Gain proficiency in defining and utilizing classes and objects in C++
- Learn techniques for code reusability and flexibility through the use of templates and generic programming, along with mastering concepts like interface classes, operator overloading, and friend functions

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the fundamental principles of Object-Oriented Programming (OOP)
CO2	Gain a deep understanding of the concepts of inheritance, polymorphism, virtual functions.
CO3	Master the concept of interface classes and implementation inheritance and acquire proficiency in exception handling
CO4	Master the concept of file handling in C++ and learn the principles of file stream classes in the Standard Template Library (STL).

Mapping of COs to PSOs

CO - PSO Mapping

	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Introduction to Programming Concepts, Overview of C++ and its Features, Writing Your First C++ Program,Structure of C++ program, Tokens, Keywords, identifiers, Variables, Data Types, Operators,manipulators, Basic Input/Output (cin, cout), type cast, type conversion,
UNIT 2	Control Structures: if-else, switch-case, loops (for, while, do-while), Introduction to Functions and Scope of Variables., Functions (Parameters, Return Types), Arrays and Strings,Introduction to Object-Oriented Programming (OOP) Concepts, Classes and Objects: Definitions and Usage, Constructors and Destructors.
First series internal examination including theory + Laboratory if any	
UNIT 3	Inheritance, Types of Inheritance, Access Specifiers, Polymorphism, Function Overriding, Operator Overloading, simple programs to implement inheritance and polymorphism.
UNIT 4	File Handling, File Stream Classes, Opening and Closing Files, File Modes Reading and Writing Files, Exception Handling , try, throw, catch , Simple Programs using File Handling and Exceptions
Second series internal examination including theory + Laboratory if any	

References:

1. 'Programming: Principles and Practice Using C++', Bjarne Stroustrup
2. Programming in C++, M.T. Somashekara, Prentice Hall of India, New Delhi
3. Object Oriented Programming with ANSI & Turbo C++, Ashok N. Kamthane, Pearson Education
4. Let us C++, Yeshwanth Kanethkar, BPB
5.Object Oriented Programming with C++; E. Balagurusamy; 3rd Edn; TMH 2006

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital

Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

KU03MDCMCA301Cyber Forensics

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	MDC	200	**KU03MDCMCA301**	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

Course Objectives:

Understand the fundamentals of cyber security, cyber crime, and cyber forensics.

Learn digital evidence collection, preservation, and forensic investigation techniques.

Understand operating system, network, mobile, and cloud forensics.

Gain knowledge of cyber laws, ethical issues, and forensic reporting.**Course Outcomes:**

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the concepts of cyber security, cyber crime, cyber law, and digital forensics.
CO2	Identify system vulnerabilities and apply evidence acquisition and preservation.
CO3	Perform basic forensic investigations on computers, networks, and mobile devices.

CO4	Analyze digital evidence and prepare legally admissible forensic reports using forensic tools.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	
CO2	✔	✔	✔	✔	
CO3	✔	✔	✔	✔	
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT1	Introduction to Information Security,CIA Triad (Confidentiality, Integrity, Availability) Cyber Security Principles, Types of Cyber Threats and AttacksMalware: Virus, Worm, Trojan, Spyware, Ransomware, RootkitsAuthentication, Authorization and Access ControlPassword Security and Multi-factor Authentication
UNIT2	Network Security BasicsFirewalls, IDS and IPS Cryptography Fundamentals Introduction to Cyber Crime. Introduction to Digital Forensics Cyber Forensic Investigation Process Digital Evidence and Chain of Custody Operating System Security (Windows and Linux)File Systems (FAT, NTFS, EXT)Storage Devices (HDD, SSD, USB, Memory Cards)Disk Structure and PartitionsEvidence AcquisitionLive AcquisitionDead AcquisitionDisk ImagingDeleted File RecoveryFile CarvingMemory (RAM) .
First series examination including theory + Laboratory if any	
UNIT3	Network Security ConceptNetwork ForensicsPacket Capture and Analysis Email ForensicsWeb Browser ForensicsLog AnalysisWireless Network ForensicsMobile Device ForensicsAndroid and iOS ForensicsSIM Card and SD Card AnalysisGPS and Location EvidenceSocial Media ForensicsCloud ForensicsIoT Forensics (Introduction)
UNIT4	Cyber Law FundamentalsInformation Technology Act, 2000 and AmendmentsDigital Signatures and Electronic RecordsIntellectual Property Rights in CyberspacePrivacy and Data ProtectionElectronic Evidence and Legal AdmissibilityCyber EthicsIncident Response and Investigation MethodologyDocumentation and Report WritingExpert Witness ResponsibilitiesStandard Operating Procedures in Digital InvestigationDigital Forensic Tools (Overview)

Second series internal examination including theory + Laboratory if any
Text Books Guide to Computer Forensics and Investigations Digital Evidence and Computer Crime Computer Security: Principles and Practice

KU03MDCMCA302 Scientific Computing

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	MDC	200	**KU03MDCMCA302**	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

Course Description

This course introduces the principles of scientific computing using Python and its scientific libraries. Students will learn computational techniques for numerical analysis, matrix computations, optimization, interpolation, differential equations, and scientific visualization using NumPy, SciPy, Matplotlib, and SymPy. The course emphasizes practical problem-solving and computational thinking through laboratory exercises and mini-projects

Course Objectives:

The students will be able to:

- Understand the concepts and applications of Scientific Computing.
- Develop Python programs using scientific computing libraries.
- Perform efficient numerical computations using NumPy.
- Apply SciPy for solving mathematical and engineering problems.
- Visualize scientific data using Matplotlib.
- Solve real-world computational problems using Python.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the principles and applications of Scientific Computing.
CO2	Develop numerical computing applications using NumPy and Python
CO3	Solve mathematical problems using SciPy and SymPy.
CO4	Create scientific visualizations for data interpretation.Design scientific computing solutions for real-world applications.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	
CO2	✓	✓	✓	✓	
CO3	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓

COURSE CONTENTS

UNIT 1 15 Hours	Introduction Scientific Computing and its Applications, Python for Scientific,Computing,Installing,Anaconda and Jupyter Notebook, Python Review: Variables, DataTypes and Operators, Functions and Modules, File Handling, NumPy Fundamentals, Introduction to NumPy, Creating Arrays, Array Attributes, Indexing and Slicing, Array, Operations, Universal Functions (ufuncs), Broadcasting, Vectorized Computation, Matrix Operation, Linear Algebra Basics Practical: NumPy array creation Matrix operations Vectorized computation
UNIT 2 15 Hours	SciPy Library, Introduction to SciPy, Scientific Constants, Numerical Integration, Optimization, Root Finding, Interpolation, Signal Processing (Introduction), Statistics Module, Symbolic Computing using SymPy, Symbolic Variables, Algebraic Expressions, Equation Solving, Differentiation, Integration, Matrix Algebra, Taylor Series Expansion Practical: Numerical integration Root finding Symbolic differentiation Symbolic integration
First series examination including theory + Laboratory if any	

UNIT 3 15 Hours	Matplotlib Introduction to Data Visualization, Line Plot, Scatter Plot ,Bar ChartHistogram ,Pie Chart,Box Plot, Subplots, Figure Customization, Saving FiguresData Analysis , Loading Scientific DataStatistical Analysis, Correlation, Descriptive StatisticsRandom Number GenerationMonte Carlo Simulation (Introduction) Practical: Visualization of datasets Statistical plots Monte Carlo simulation
UNIT 4 15 Hours	Numerical Computing Solving Linear Systems , Eigenvalues and Eigenvectors, Numerical Differentiation, Numerical Integration, Curve Fitting, Polynomial Approximation ,Ordinary Differential Equations using SciPy, Optimization Problems, Scientific ApplicationsComputational Physics, Computational Biology , Financial Computing , Image Processing (Introduction). Practical: Solving ODEs Curve fitting Optimization Scientific mini-project Practical Components
Second series internal examination including theory + Laboratory if any	

Text Books

Robert Johansson, Numerical Python: Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib, 2nd Edition, Apress, 2019.

Additional References

Hans Petter Langtangen, A Primer on Scientific Programming with Python, Springer.
Steven C. Chapra and Raymond P. Canale, Numerical Methods for Engineers, McGraw-Hill.
Jake VanderPlas, Python Data Science Handbook, O'Reilly.
Wes McKinney, Python for Data Analysis, O'Reilly.
Richard L. Burden and J. Douglas Faires, Numerical Analysis, Cengage Learning.

Semester I

KU01DSCMCA101 INTRODUCTION TO COMPUTATIONAL INFORMATICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
1	DSC	100	KU01DSCMCA101	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description: This course provides a comprehensive overview of computing, covering historical milestones, hardware components, software systems, and computational thinking principles. Students will explore the evolution of computing systems, from early pioneers to modern processors and quantum units. The curriculum delves into hardware intricacies, software distinctions, and essential concepts in computer science, emphasizing problem-solving skills and algorithmic thinking. Practical aspects include hands-on experiences with hardware assembling, operating system installation, algorithm and flowchart visualization.

Course Objectives:

- To become aware about the evolution and generation computer system and familiarises various components of computer systems
- To become familiar with various computer hardware components
- To understand different software components and its types and the installation and configuration of operating systems

  selection
- To familiarise different methodologies of problem solving using the computational thinking concept.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Develop a foundational knowledge of computing systems, encompassing their historical development, evolutionary milestones, and the notable contributions of key figures in the field.
CO2	Acquire familiarity with diverse hardware components constituting a computer system.
CO3	Attain the knowledge of computer software and the activities focused on the installation and configuration of diverse hardware components within a computer system including operating systems
CO4	Develop a foundational understanding of computer science as a discipline, examining problems through the lens of computational thinking and inducing the analytical skills to address challenges in the field.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	
CO2	✔	✔	✔	✔	
CO3	✔	✔	✔	✔	
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT1 15 Hours	**Evolution of Computers:** History, Generations. Overview of Computer System- Von Neumann Model, Number Systems (Binary, Hexa, Octal, Decimal) , Digital Codes (Gray, Excess-3, BCD). Pioneers and Contributors of Computing Systems - First Mechanical computer - Charles Babbage, Stored-Program Architecture - John von Neumann, Turing machine - Alan Turing, First General-Purpose Electronic Digital Computer - John Mauchly and J. Presper Eckert, Artificial Intelligence- John McCarthy (Contributions only). Computing Systems: Past to Present - Single Core, Dual-Core and Multi-Core Processors, GPU, TPU, APU, and QPU.
UNIT2 15 Hours	**Hardware:** Electronic Components – Active Components - Diode, Transistor, Integrated Circuits (Definition, Symbol and Function), Electronic Components - Passive Components – Resistors, Capacitors, Inductors (Definition, Symbol and Function). Motherboard Components – CPU and

	Cooling Fan, RAM, Expansion Slots (PCIe), Input/Output Ports, Chipset. Motherboard Components – BIOS/UEFI Chip, SATA/NVMe Slots, Network Interface, Ports- Ethernet, VGA, HDMI, USB . Computer Components – SMPS, Motherboard, Storage Devises (HDD, SSD, NVMe). Computer Components – RAM (DRAM, SRAM, DDR SDRAM), ROM, Cache.
First series examination including theory + Laboratory if any	
UNIT3 15 Hours	**Software:** System Software, Examples. Operating System – Need of OS, Types – Proprietary and Open Source, Hardware Software Compatibility, POST, Booting. OS Installation – Bootable Media, UEFI/Legacy BIOS, Disk Partitioning, Dual Booting, Boot Manager – BOOTMGR, Grub, File Systems- FAT, NTFS, ext4. Device Drivers – Need of Device Drivers, Driver Interactions (Basic).
UNIT4 15 Hours	**Module 4: Computational Thinking:** Problem Solving - Defining the Problem, Systematic Approach. Computational Thinking – Problem Decomposition, Pattern Identification, Abstraction, Generalization. Logical Thinking – Inductive and Deductive Reasoning, Logical Expressions. Algorithmic Thinking – Intuition vs Precision, Defining algorithms. Algorithm – Need of Algorithm, Qualities of a Good Algorithm, Examples. Flowchart - Flowchart Symbols, Examples. Raptor.
Second series internal examination including theory + Laboratory if any	
Books/References 1. Gary B. Shelly, Thomas J. Cashman, and Misty E. Vermaat. “Introduction to Computers”, Cengage Learning, 2008. 2. Pradeep K. Sinha and Priti Sinha, Computer Fundamentals: Concepts, Systems & Applications. BPB Publications. 3. Kevin Wilson, Computer Hardware: The Illustrated Guide to Understanding Computer Hardware. Amazon Digital Services LLC – KDP, 2018. 4. John Hanna, OS Installation 101: A Step-by-Step Approach for Newbies. 5. David Riley and Kenny Hunt, Computational thinking for modern solver, Chapman & Hall/CRC, 2014. 6. R.G. Dromey, How to solve it by Computer, PHI, 2008.	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7.

KU01DSCMCA102PRINCIPLES OF PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
1	DSC	100	KU01DSCMCA102	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2

*** ESE duration: 2 hours for theory and 3 hours for Lab**

Course Description: Computer Science is all about developing correct and efficient solutions for our day-to-day problems. The process of developing solutions is not centered on learning a programming language and doing coding straight away. Instead, a blueprint of the proposed solution should be outlined and tested for correctness. Once a proposed blueprint leads to a correct solution, it can be implemented using a suitable programming language. The objective of this course is to impart knowledge to the learner about building the blueprint of a solution. Learners are also exposed to implementing the solutions using the C programming language.

Course Objectives:

- To impart knowledge about various constructs for developing solutions
- To become familiar with using the various constructs to develop solutions
- To compare and contrast various constructs for solution development for selection
- To compare and contrast various constructs for solution development for iteration
- To implement solutions using C programming language

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Illustrate the foundations of developing solutions using flowcharts and algorithms
CO2	Develop solutions using various selection constructs and implement them in the C programming language
CO3	Develop solutions using various iteration constructs and implement them in the C programming language
CO4	Understand advanced concepts in direct memory handling, file handling and functions.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 05 Hours Theory + 05 Hours Lab	Problem-Solving: Preparing Solutions using Flowcharts: Conventions - Structure - Symbols. Preparing Solutions using Algorithms - Conventions - Top-Down and bottom-up approach. Program**:** Characteristics - Modular Approach - Style - Documentation and Maintenance - Compilers and Interpreters - Preparing, Running and Debugging Programs - Types of Errors. Fundamentals of C Language: Evolution and Features - Program Structure - Elements - Constructs. Character Set, Tokens, Keywords, Identifier. Data Types, Constants, Symbolic Constants, Variables, Data Input and Output, Statements - Assignment statements. Operators in C: arithmetic, relational, logical, assignment, auto increment, auto decrement, conditional, comma operators. Precedence of operators - expressions – evaluation of expressions, type conversion in expressions – precedence and associativity
UNIT 2 05 Hours	Selection Constructs: Simple if - if else - if else if ladder - switch. Branching statements: break, goto. Case study: Developing solutions (flowcharts and algorithms) for problems using various

Theory + 15 Hours Lab	selection constructs - Comparative Study of various Selection Constructs - Converting a solution using one selection construct with other selection constructs.
First series internal examination including theory + Laboratory if any	
UNIT 3 07 Hours Theory + 15 Hours Lab	Iteration Constructs: Top Tested Vs Bottom Tested - while - for - do while - Nesting of loops - skipping breaking loops. Arrays - 1D and 2D, 3 D - Case study: Developing solutions (flowcharts and algorithms) for problems using various iteration constructs - Comparative Study of various iteration constructs - Converting a solution using one iteration construct with other iteration constructs. Functions and function calling mechanisms.
UNIT 4 13 Hours Theory + 25 Hours Lab	Advanced concepts in C: Concepts of memory allocation for variables- Direct memory accessing - Pointers- pointer arithmetic's- structures- files and file operations- preprocessor directives- preparing customized header files.
Second series internal examination including theory + Laboratory if any	
Books/References	
6. J.B Dixit, Computer Fundamentals and Programming in C, Firewall Media	
7. Anil Bikas Chaudhuri, The Art Of Programming Through Flowcharts Algorithms, Laxmi Publications, New Delhi.	
8. Maureen Spraknle and Jim Hubbard, Problem Solving and Programming Concepts, Pearson	
9. E Balagruswamy, Programming in ANSI C, TMH, 5th Edition	
10. R G Dromey, How to Solve by Computer, Pearson Education, 5th Edition	
11. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, PHI	
12. Kanetkar, Let Us C, BPB Publications, 8th Edition	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7.

KU01DSCMCA103: MATHEMATICAL FOUNDATIONS FOR COMPUTER APPLICATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
1	DSC	100	KU01DSCMCA103	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description: This course provides a fundamental exploration of mathematical concepts essential for computer science. Students will explore into key topics including Linear Algebra, Differential and Integral Calculus. The course aims to equip students with the mathematical tools and reasoning skills necessary for creating and analyzing algorithms, understanding and solving computational problems in various areas of computer science like Data science, Artificial Intelligence.

Course Objectives:

- To impart knowledge propositional logic
- To become familiar with set theory and its applications.
- To acquaint with different matrix operations and arithmetic related to computer related problems selection
- To familiarise different types of linear equations and solving and interpreting the solutions

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Reflect the concept of proportional logic , predicative logic and inference rule for solving computational problems

CO2	Understand the concept and use set theory for solving computational problems using reasoning and approximation.
CO3	Reflect the concept of matrices and determinants as a way to depict and streamline mathematical ideas to perform basic operations.
CO4	Acquire proficiency in solving linear equations using different techniques and understanding the geometric interpretation of solutions.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 10 Hours	**Propositional Logic**: Basic logical operations, conditional statement, bi-conditional statement, converse, inverse and contrapositive statement, well-formed formula, tautology, contradiction, equivalence of formula. laws to determine equivalence, tautological implication. duality law, normal forms- (CNF, DNF, PCNF, PDNF), Predicate calculus- rules of inference, valid arguments, types of quantifiers, properties of quantifier**.**
UNIT 2 10 Hours	**Set Theory**: Set, types of set operations and laws. algebra of sets and duality, inclusion and exclusion principle, Cartesian product- Relations: Definition. properties of relation, types of relation, equivalence class, relation matrix and graph of relation, partition and covering of set, poset, composition of relation**.**
First series internal examination including theory + Laboratory if any	
UNIT 3 20 Hours	**Matrices and Determinants** -Matrices: Definition, Order of a matrix, Types of matrices Operations on matrices: Addition, Subtraction, Multiplication Properties of matrix: Various kind of Matrices, Transpose of a matrix . Elementary Transformations of Matrices and Rank of Matrices Symmetric and Skew Symmetric Matrices . Determinants, Minors, Cofactors, Inverse of

	a matrix.
UNIT 4 20 Hours	**Linear Algebra and Vector Operations**. Linear Independence: Characteristic equations, Eigenvalues, Eigenvector. Solving system of linear equations: Gauss Elimination Method, Gauss Jordan method, Gauss Siedel Methods. Vectors: Definition Magnitude of a vector, Types of Vectors, Vector addition. Dot products and Cross products. Vectors in 2- and 3-space 2.
Second series internal examination including theory + Laboratory if any	

Books/References

1. J.P. Tremblay, R. Manohar- Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill

2.Kenneth H Rosen - Discrete Mathematics and its Applications with Combinatorics ad graph Theory, Seventh Edition, McGraw Hill

3. Skills in Mathematics: Algebra, S.K.Goyal

4. Higher Engineering Mathematics, B S Grewal, Khanna Publishers

5. Higher Engineering Mathematics, Ramana, Tata McGraw Hill

6. Mathematics, P Kandasamy, S. Chand Group

7. Gilbert Strang, "Introduction to Linear Algebra", Wellesley-Cambridge Press, 2023.

8. Kenneth Hoffman, Ray Kunze, " Linear Algebra", Prentice Hall India Learning, 2015.

9. Gilbert Strang, "Calculus", Wellesley-Cambridge Press, 2023.

10. Joseph Edwards, "Differential Calculus for Beginners", Arihant Publications, 2016.

11. Joseph Edwards, " Integral Calculus for Beginners", Arihant Publications, 2016.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

Semester II

KU02DSCMCA104: OBJECT ORIENTED PROGRAMMING USING C++

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
2	DSC	100	KU02DSCMCA104	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2

*** ESE Duration : 2 hours for theory and 3 hours for Lab**

Course Description:

This course provides an introduction to the principles and practices of object-oriented programming (OOP) using the C++ programming language. Students will learn fundamental concepts of OOP and how to apply them to develop software solutions using C++.

Course Objectives:

- Understand the principles of Object-Oriented Programming (OOP) using C++
- Master the syntax and semantics of the C++ programming language
- Gain proficiency in defining and utilizing classes and objects in C++
- Learn techniques for code reusability and flexibility through the use of templates and generic programming, along with mastering concepts like interface classes, operator overloading, and friend functions

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the fundamental principles of Object-Oriented Programming (OOP)
CO2	Gain a deep understanding of the concepts of inheritance, polymorphism, virtual functions.
CO3	Master the concept of interface classes and implementation inheritance and acquire proficiency in exception handling
CO4	Master the concept of file handling in C++ and learn the principles of file stream classes in the Standard Template Library (STL).

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 05 Hours Theory + 10 Hours Lab	Introduction to Programming Concepts, Overview of C++ and its Features, Writing Your First C++ Program,Structure of C++ program, Tokens, Keywords, identifiers, Variables, Data Types, Operators,manipulators, Basic Input/Output (cin, cout), type cast, type conversion, Control Structures: if-else, switch-case, loops (for, while, do-while), Introduction to Functions and Scope of Variables.
UNIT 2 10 Hours Theory + 20 Hours Lab	Detailed Study of Functions (Parameters, Return Types), Function Overloading,Introduction to Arrays and Strings, Basics of Pointers and References- Pointers to objects; Pointers to derived classes. Introduction to Object-Oriented Programming (OOP) Concepts, Classes and Objects: Definitions and Usage, Constructors and Destructors.

First series internal examination including theory + Laboratory if any	
UNIT 3 10 Hours Theory + 20 Hours Lab	Inheritance: Basics and Types, Polymorphism and Virtual Functions, Operator Overloading, Introduction to Templates (Basics), Working with the Standard Template Library (STL): Vectors, Lists, Introduction to Exception Handling.
UNIT 4 05 Hours Theory + 10 Hours Lab	C++ streams; stream classes; unformatted I/O operations; Formatted console I/O operations; Managing output with manipulators. Files – classes for file stream operations; Opening and closing a file; file modes; file pointers and their manipulations; Sequential input and output operation.
Second series internal examination including theory + Laboratory if any	
References: 1. 'Programming: Principles and Practice Using C++', Bjarne Stroustrup 2. Programming in C++, M.T. Somashekara, Prentice Hall of India, New Delhi 3. Object Oriented Programming with ANSI & Turbo C++, Ashok N. Kamthane, Pearson Education 4. Let us C++, Yeshwanth Kanethkar, BPB 5.Object Oriented Programming with C++; E. Balagurusamy; 3rd Edn; TMH 2006	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

KU02DSCMCA105 DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
2	DSC	200	KU02DSCMCA105	4	60

Learning Approach (Hours/ Week)	Marks Distribution	Duration of

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	ESE (Hours)
4	0	1	50	50	100	2

Course Description: Digital Electronics and Computer Organization is an introductory course that delves into the fundamental principles underlying modern digital systems and computer architecture. The course is designed to provide students with a comprehensive understanding of digital circuits, logic design, CPU architecture, and the organization of computer systems. Students will explore the concepts of binary arithmetic, Boolean algebra, Combinational and sequential logic, instruction execution, and CPU operation.

Course Objectives:

- Understand different number systems and Boolean algebra.
- Design of Combinational and sequential logic circuits
- Understand different Computer Instructions
- Understand concepts of register transfer logic and arithmetic operations.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Able to perform the conversion among different number systems.
CO2	Able to design Combinational and sequential logic circuits
CO3	To present the Digital fundamentals, Boolean algebra and its applications in digital systems
CO4	To familiarize with the design of various digital circuits using logic gates

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	Number Systems – Decimal, Binary, Octal, Hexadecimal number, Number Conversion, binary arithmetic, 1‘s and 2‘s complement arithmetic, BCD arithmetic, Boolean Laws: Commutative Laws, Associative Laws and Distributive Laws, Boolean Theorems, Sum of Products and Product of Sums.
UNIT 2 **10 Hours**	De-Morgan’s Laws, Digital Logic Gates:-AND, OR, NOT,XOR,XNOR functions, Universal Logic Gates: NAND and NOR gates, Realization of Boolean expressions using logic gates, Circuit simplification using Boolean Algebra, Logic Diagrams, Combinational Logic Circuits, Simplification of Logic functions, Karnaugh map simplification.
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	Multiplexer, and De-multiplexer: Truth table and logic expression, Implementation using logic gates, Design of Half adder and full adder, Construction of full adder using half-adders, Decoder, Encoder, Digital to analog converter, Analog to digital converter.
UNIT 4 **25 Hours**	Concept of Flip-flops, SR latch, Gated SR latch, Shift Registers, Serial in – Serial out Shift Register (SISO), Serial In – Parallel out shift Register (SIPO), Parallel in – Parallel out Shift Register (PIPO), Parallel in – Serial out Shift Register (PISO), Bidirectional Shift Registers, Accessing I/O Devices, Interrupts – Interrupt Hardware, Direct Memory Access, Buses, Basic concepts of Pipe lining. (25 hours)
Second series internal examination including theory + Laboratory if any	
UNIT X	**Module X:** Instruction formats, Instruction sets, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, memory organization: Memory Hierarchy, Main memory, Auxiliary memory, Associate memory, Cache memory, Complex Instruction Set Computer (CISC) Reduced Instruction Set Computer (RISC),Register Transfer Language, Register Transfer, Bus and Memory Transfers.

Books/References

1. M. Morris Mano, Michael D. Ciletti, “Digital Design”, Pearson, 2013.
2. A. K. Maini, “Digital Electronics: Principles, Devices And Applications, Wiley, 2007.
3. R. Gaonkar, “Microprocessor Architecture, Programming and Applications with the 8085”, Prentice Hall, 2014.
4. "Digital Design and Computer Architecture" by David Harris and Sarah Harris.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU02DSCMCA106: STATISTICAL FOUNDATIONS FOR COMPUTER APPLICATIONS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
2	DSC	100	KU02DSCMCA106	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description: The course on probability and statistics covers fundamental topics including descriptive statistics (measures of central tendency and dispersion), probability theory (events, sample spaces, probability laws, random variables, and distributions), inferential statistics (regression analysis), and applications in various fields such as science, engineering, economics, and social sciences, emphasizing critical thinking, data analysis, and problem-solving skills.

Course Objectives:

- Understand fundamentalstatistical methods for evaluating data samples
- Familiarises the concept of descriptive statistics
- Understand the concept of probability and its applications
- Analysing the data using various statistical methods.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Apply fundamental statistics concepts
CO2	Analyze data using descriptive statistics
CO3	Apply probability and statistics in real-world situations
CO4	Communicate statistical findings effectively

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	Statistical methods: Concepts of statistical population and sample. Collection of Data: Primary and Secondary Data, Methods of Collecting Primary Data. Measures of Central Tendency: Characteristics of a good measure of central tendency, Mean, median, mode, harmonic mean, geometric mean, weighted mean. Measures of Dispersion: Range, quartile deviation, mean deviation, standard deviation and variance. **[05 Hours]**

UNIT 2 **15 Hours**	Random experiment, sample space and events, impossible events, mutually exclusive and exhaustive events, independent and dependent events, equally likely events with examples, Operation of events (Union, Intersection, Complement of Events), Definitions of probabilities, Axioms of probability, De'Morgans Law, Theorems of probability - Addition Theorem, Multiplication, Conditional probability, Baye's Theorem. **[15 Hours]**
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	Basic principles of sample surveys, advantages of sample survey over census, sampling and non-sampling errors , probability sampling, judgment sampling and non-probability sampling, Simple Random Sampling, Systematic Sampling, Stratified Sampling, Cluster Sampling, Multistage sampling **[20 Hours]**
UNIT 4 **20 Hours**	Correlation analysis - Definition and properties of correlation coefficient, Pearson's Coefficient of Correlation , Rank correlation coefficient, Method of least squares and Fitting of Straight Line , Scatter diagram, Regression analysis - linear regression, fitting of regression lines, regression coefficients and their properties, relation between correlation and regression coefficients. **[20 Hours]**
Second series internal examination including theory + Laboratory if any	

Books/References:

1. B L Agrawal (2013): Basic Statistics – New Age International Publishers.
2. Probability and Statistics for Computer Scientists by Michael Baron
3. Gupta, S.P. Statistical Methods. Sultan Chand and Sons: New Delhi
4. Gupta, S. C. and Kapoor, V. K. (2020)Fundamentals of Mathematical Statistics. Sultan Chand and Sons' Publishers, New Delhi.
5. Probability and Statistics for Engineers, Miller I Freund J E, Prentice Hall of India
6. Statistics for Management, Levin R I, Prentice Hall of India

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU02DSCMCA107: INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours

2	DSC	100	KU02DSCMCA107	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: This course provides a comprehensive study of database management systems (DBMS), covering both theoretical concepts and practical implementations. Students will gain an understanding of the fundamental principles underlying the design, implementation, and management of modern database systems.

Course Objectives:

- Understand the fundamental concepts of database systems
- Learn about the relational data model, including relations, keys, and referential integrity
- Develop a solid understanding of relational algebra operators and their application in querying databases
- Learn SQL (Structured Query Language)
- Learn about the normalization process and the desirable properties of decompositions up to BCNF
- Understand transaction management, concurrency control, and error recovery mechanisms in database systems

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the fundamental concepts of database systems
CO2	Develop proficiency in developing queries and subqueries

CO3	Gain a deep understanding of the definitions and properties of 1NF, 2NF, 3NF, and BCNF
CO4	Equips students with the knowledge and skills necessary to design, implement, and manage transaction processing systems effectively

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	Introduction to database systems, Data Abstraction and System structure, View of Data, Data Models, database structure, DBA, Data Base Users. E-R model, Basic concepts; design issues; Conceptual data modeling - entities, entity types, various types of attributes, relationships, relationship types, E/R diagram notation. Mapping Constraints; Relational Data Model - Concept of relations, keys: Primary, Foreign, candidate, referential integrity and foreign keys.
UNIT 2 **07 Hours** **25 Hours**	Relational algebra operators, various types of joins, set operation, division, example queries, tuple relational calculus, domain relational calculus. SQL - Introduction, data definition in SQL, table. Querying in SQL - basic select-from-where block and its semantics, nested queries - correlated and uncorrelated, notion of aggregation, aggregation functions group by and having clauses. DDL, DML, DCL, SQL Functions, Data types in SQL. Developing queries and subqueries
First series internal examination including theory + Laboratory if any	
UNIT 3 **10 Hours** **Theory +** **25 Hours Lab**	Dependencies and Normal forms - Problems encountered with bad schema designs, motivation for normal forms, dependency theory - functional dependencies, Armstrong's axioms for FD's, closure of a set of FDs, minimal covers, definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.
UNIT 4	Integrity constraints, views, Trigger and Sequences, Relational model – Structure of Transaction

08 Hours Theory + 10 Hours Lab	processing and Error recovery - ACID properties. Transactions and Schedules – Characterizing Schedules based on Recoverability, Serializability of schedules. Concurrency Control in databases: Locking Techniques-Timestamp ordering, Multi version concurrency Control – Granularity of data items, error recovery and logging, undo, redo, undo-redo logging and recovery methods.
UNIT X	Overview and History of NoSQL Databases. Definition of the Four Types of NoSQL Databases, NoSQL Key/Value databases, Document Databases, Document oriented Database Features, Graph data model, Column family data model
Second series internal examination including theory + Laboratory if any	
Books/References: 1. H Silbersehatz, Korth and Sudarshan, Database system concepts, 6th edition MGH 2011 Ramakrishnan and Gehrke, Database Management Systems, 3rd Edn, McGraw Hill, 2003 Elmasri and Navathe, Fundamentals of Database systems, 5th Edition, Pearson 2009 C.J.Date-A.Kannan, S.Swamynathan, An introduction to Database System, 8th Edition, Pearson education O'Reilly, Practical PostgreSQL Shroff Publishers (SPD) 2002. Redmond, E. & Wilson, J. (2012). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (1st Ed.). Raleigh, NC: The Pragmatic Programmers, LLC. ISBN-13: 978-1934356920 ISBN-10: 1934356921	

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration, Digital Learning, Team Work

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Refer to section 7

Semester III

KU03DSCMCA201 JAVA PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	300	KU03DSCMCA201	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2

*** ESE duration : 2 hours for theory and 3 hours for Lab**

Course Description: By the end of this course, students will have a solid foundation in Java programming and web development concepts, equipping them with the skills needed to develop robust Java applications and web solutions.

Course Objectives:

- Fundamental Understanding of Java
- Master multithreading concepts and techniques.
- Understand AWT's abstract window toolkit.
- Understand JDBC architecture, JDBC drivers, and steps to connect to a database.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understanding Java Basics and students will gain proficiency in object-oriented programming concepts
CO2	Proficiency in multithreading concepts, expertise in performing input and output operations and developing skills in GUI programming using Java AWT and Swing.
CO3	Understand the architecture of JDBC and various JDBC drivers, and learn how to manage database transactions and handle errors effectively using JDBC
CO4	Develop skills in creating, deploying, and managing Java servlets, gain proficiency in JavaServer Pages (JSP).

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔		✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours Theory + 20 Hours Lab**	Features of Java, Byte code, JDK, JRE, JVM, Data Types -Type Conversion and Casting, Variables, Operators, Control Statements, Looping Statements, Arrays, Strings, Class, Objects- Declaring Objects, Access Specifier, Static, Nested and Inner Classes, this Keyword, Garbage Collection, Methods- method overloading, Constructors, Inheritance-Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interface- Defining an Interface, Implementing Interfaces, Packages - Declaring a package, Importing Packages, Sub-packages, Exception Handling - Types of Exceptions, try-catch, throw, throws, and finally
UNIT 2 **10 Hours Theory + 20**	Thread - Life cycle of a thread, Synchronization, Thread class & Runnable interface, Multithreading. I/O streams, File streams: File Input Stream and File Output Stream, Data Input and Output Streams, Buffered Input and Output Streams Applets- Applet life cycle, working

Hours Lab	with Applets, Working with Graphics: Abstract Window Toolkit (AWT) - Components: Container, Panel, Window, Frame. AWT Controls, Listeners, Layout Managers, Event Handling - Events, Event Sources, Event Classes, Event Listener Interfaces, Adapter Classes. Swing (JFC) - Swing components
First series internal examination including theory + Laboratory if any	
UNIT 3 **03 Hours Theory + 05 Hours Lab**	Database connectivity - JDBC architecture, JDBC Drivers, Steps to connect to Database-Connectivity with MySQL, Driver Manager, Types of JDBC statements: Statement, Prepared statement, Callable statement, ResultSet - Types of ResultSet, blobs and clobs,metadata - Database Metadata, Resultset Metadata, transactions, error handling
UNIT 4 **07 Hours Theory + 15 Hours Lab**	Model-View-Controller (MVC) Architecture, Java Servlets: Introduction to servlet, Servlet life cycle, Developing and Deploying Servlets, Generic and http servlets, GET, POST, HEAD and other requests, Servlet responses, error handling, security, servlet chaining, cookies, session tracking , Java Server Pages: Introduction to JSP, JSP life cycle, JSP expressions and declarations, Directives and Actions: Page directives, JSP actions and implicit objects, JSP Tag Libraries: Standard and Custom Tag Libraries, Expression Language (EL)
UNIT X	Spring Framework: Introduction to Spring: Dependency Injection (DI) and Inversion of Control (IoC),Spring AOP (Aspect-Oriented Programming),Spring MVC :Configuring Spring MVC, Handling web requests, Hibernate: Introduction to Hibernate :Object-Relational Mapping (ORM) Hibernate architecture , Mapping in Hibernate :Mapping Java classes to database tables, HQL (Hibernate Query Language)
Second series internal examination including theory + Laboratory if any	

Books/References:

Herbert Schildt, The complete reference Java2 ,11thed, Released December 2018

Publisher(s): McGraw-Hill ISBN: 9781260440249

JasonHunder& William Crawford, Java Servlet Programming, O'REILLY, 2002

Marty Hall, Larry Brown Core Servlets and Java server pages. Vol 1: Core Technologies. 2nd Edition.

David Flanagan, Java in a Nutshell A desktop quick Reference, 7 Edition, 2018. OReilly& Associates Inc

Rajkumar, Java programming, Pearson, 2013

HarimohanPandey, Java Programming, Pearson, 2012

David Flanagan, Jim Parley, William Crawford & Kris Magnusson , Java Enterprise in a nutshell- A desktop Quick reference -O'REILLY, 2005

Stephen Ausbury and Scott R. Weiner, Developing Java Enterprise Applications, Wiley-2001

Database Programming with JDBC and Java, Reese George, Oreilly

Christian Bauer, Gavin King, and Gary Gregory, “Java Persistence with Hibernate” 2015

Craig Walls "Spring in Action" 2022

Christian Bauer and Gavin King, “Hibernate in Action” 2004

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital

Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU03DSCMCA202 INTRODUCTION TO DATA STRUCTURE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	200	KU03DSCMCA202	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2(T)+3(P)*

*** ESE Duration : 2 hours for theory and 3 hours for Lab**

Course Description: This course helps the learners to understand the various data structures, their organization, and operations. The course is also intended to help the learners to assess the applicability of different data structures and associated algorithms for solving real-world problems which require minimum space and time complexities.

Course Objectives:

- To impart knowledge about various data structures, their representation, and applications.
- Acquire knowledge of various searching and sorting techniques
- To familiarize the design and analysis of linear data structure algorithms related to linked list, stack and queue.
- To familiarize the usage of non-linear data structures and its implementation.

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Familiarizes the need of data structure for organizing/storing data in a computer system effective manner.
CO2	Design and implement array data structure and its operations for solving real world problems.
CO3	To impart knowledge about data structures like linked list, stack, queue and its implementation.
CO4	Familiarization of selected non-linear data structure, its representation and applications in real world problem scenarios.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	**Introduction:** Elementary data organization, Data Structure definition, Data type vs. data structure, Linear and non-linear data structures, Data structure operations, Applications of data structures, algorithms, characteristics, Performance Analysis, Space Complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms.
UNIT 2 **10 Hours Theory + 20 Hours Lab**	Introduction, Linear arrays, Representation of linear array in memory, Traversal, Insertions, Deletion operations, Multidimensional arrays, Parallel arrays, sparse matrix. **Searching & Sorting:** Linear search vs Binary search, Selection Sort, Insertion Sort, Bubble sort, Quick Sort and Merge Sort. **Strings:** Representation, operations and pattern matching. **Recursion:** concept and implementation.
First series internal examination including theory + Laboratory if any	

UNIT 3 **10 Hours Theory + 20 Hours Lab**	Introduction, Array vs. linked list, Representation of linked lists in memory, Traversal, Insertion, Deletion, Searching in a linked list, Header linked list, Circular linked list, Two-way linked list, Garbage collection, Applications of linked lists**. Stack:** Representation of stack using array and linked list, Primitive operation and implementation **Stacks applications:** conversion of infix to postfix expression, evaluation of expression. **Queues:** Representation of Queues as Linked List and array, Primitive Operations, Circular queue, Priority queue, Applications of queue.
UNIT 4 **05 Hours + 10 Hours Lab**	**Non-linear data structure: Trees** - Basic Terminology, representation, Binary Trees, Tree Representations using Array & Linked List, Basic operation on Binary tree, **Traversal of binary trees:-** In order, Preorder & post order, algorithms of tree traversal with and without recursion. Applications of Binary tree, Binary search tree. **Graphs:** Introduction to graphs, Definition, Terminology, Directed, Undirected & Weighted graph, Representation of graphs.
UNIT X	Polynomial representation with arrays and linked list, Parallel array, Implementation of sparse matrix, Implementation of recursion using stack, introduction to AVL tree and its implementation. Graph traversing algorithms.
Second series internal examination including theory + Laboratory if any	
Books/References Seymour Lipschutz, "Data Structures", Tata McGraw- Hill Publishing Company Limited, Schaum's Outlines, New Delhi. YedidyanLangsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, "Data Structures Using C", Pearson Education., New Delhi. Trembley, J.P. And Sorenson P.G., "An Introduction to Data Structures With Applications", Mcgraw- Hill International Student Edition, New York Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Universities Press, Fundamentals of Data Structures in C. Samanta D., Classic Data Structures, Prentice Hall India. Richard F. Gilberg, Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2/e, Cengage Learning.	

TEACHING LEARNING STRATEGIES

- **Lecturing, Demonstration, Digital Learning, Team Work**

MODE OF TRANSACTION

- **Lecture, Seminar, Discussion**

ASSESSMENT RUBRICS

Refer to section 7.

KU03DSCMCA203 WEB TECHNOLOGY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	300	KU03DSCMCA203	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2

ESE Duration: 2 hours for theory and 3 hours for Lab

Course Description: This course provides a comprehensive introduction to web technologies, covering both frontend and backend development concepts. Students will gain practical skills in building dynamic and responsive web applications while learning about the fundamental principles of web design, development, and deployment.

Course Objectives:

- Understand the structure and functionality of the internet and the World Wide Web.
- Differentiate between client-side and server-side technologies and understand their respective roles in web development.
- Gain an introduction to server-side programming languages such as PHP, Python, or Node.js.
- Gain an overview of web hosting options and deployment strategies for deploying web applications.
- Build a simple web project from scratch using frontend and backend technologies learned in previous modules.

Course Outcomes:

SL #	Course Outcomes
CO1	Create basic web pages using HTML and CSS.
CO2	Demonstrate proficiency in JavaScript programming
CO3	Understand the concepts of database management systems (DBMS) and SQL for managing data in web applications.
CO4	Students will be able to build a simple web project

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1			✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours Theory + 10 Hours Lab**	HTML5: New Elements -Structural Elements, New Form/Input Elements, New Attributes Canvas, Video and Audio, Web Storage. Introduction to JavaScript -Syntax Variables and Data Types -Statements -Operators-LiteralsFunctions-Objects-Arrays-Built-in Objects. Host Objects: Browsers and the DOMIntroduction to the Document Object Model DOM History and Levels Intrinsic Event Handling-Modifying Element Style-The Document Tree DOM Event Handling. Scripting with HTML5.JQuery: jQuery Library, jQuery Basics, jQuery Getters and Setters, Altering Document Structure, Handling Events with jQuery.
UNIT 2 **10 Hours Theory + 20 Hours Lab**	PHP: Syntax and variables, Control and functions, string and arrays, creating functions, reading data in web pages, advanced object-oriented programming, Session, Cookies, FTP and HTTP, integrating payment system; Working with database: connecting to MySQL, making MySQL queries, fetching data, MySQL functions, displaying data in tables.

First series internal examination including theory + Laboratory if any	
UNIT 3 **05 Hours** **Theory + 10 Hours Lab**	Introduction to Web 2.0: Difference between Web 1.0 and Web 2.0, MVC Architecture. Scripting XML and JSON: XML Basics, XML request and responses, XML Parsing, XML in a string, XPath, XSTL.JSON Requests and responses, JSON Parsing. Ajax: Using XML and JSON, Syndication: RSS and Atom Feeds.
UNIT 4 **10 Hours** **Theory + 20 Hours Lab**	Content Management System: Introduction, need of CMS, Understanding CMS technologies, Different types of CMS: Portals, Wikis, Blog etc., their features and possible uses. Web services: Introduction, Web service architecture - RPC, SOA, REST, Web service standards – SOAP, WSDL, UDDI. Progressive Web Apps – Introduction, Features, Advantages.
UNIT X	HTML5 elements, Geolocation, DHTML, Introduction to Bootstrap and responsive web design basics, Mash-ups: Introduction, Hybrid application development – Basics, Discover the platforms and frameworks used for hybrid application development.
Second series internal examination including theory + Laboratory if any	
Books/References: Jeffrey C. Jackson, Web Technologies: A Computer Science Perspective, Prentice Hall 2. David Flanagan, JavaScript: The Definitive Guide, 6th Edn. O'Reilly Media.2011 3. Steven Holzner, PHP: The Complete Reference, McGraw Hill Professional, 2008 4. Steve Suehring, Tim Converse, Joyce Park, PHP6 and MY SQL Bible, John Wiley & Sons, 2009 5. Anthony T. HoldenerIII, Ajax: The Definitive Guide, O'Reilly Media, 2008 6. Bob Breedlove, et al, Web Programming Unleashed, Sams Net Publishing, 1stEdn 7. Pedro Teixeira, Instant Node.js Starter, Packt Publishing Ltd., 2013 8. James Snell, Programming Web Services with SOAP, O'Reilly 2002 10. Jacob Lett, Bootstrap Reference Guide, Bootstrap Creative 2018 9. Maximilian Schwarzmüller, Progressive Web Apps (PWA) - The Complete Guide, Packt Publishing 2018 10. Mahesh Panhale, Beginning Hybrid Mobile Application Development, Apress 2016	

TEACHING LEARNING STRATEGIES

- **Lecturing, Demonstration, Digital Learning, Team Work**

MODE OF TRANSACTION

- **Lecture, Seminar, Discussion**

ASSESSMENT RUBRICS

Refer to section 7.

KU03DSCMCA204 DATA COMMUNICATION AND COMPUTER NETWORKS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
3	DSC	300	KU03DSCMCA204	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:

This course exposes the fundamental concepts of main concepts of networking to the learner. TCP/IP network model is taken as the basis for the discussion. The course provides the learner a brisk walk through the fundamental concepts of application layer, transport layer, data link layer, and transport layer.

Course Objectives:

- Provide basic knowledge of the concepts of computer networks
- Give a basic idea about the OSI Model and TCP/IP Network models
- Familiarize with the basic concepts of various layers in TCP/IP network model

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the features of computer networks, protocols, and network design models
CO2	Illustrate the functions and various algorithms / protocols used of application layer and transport layer of TCP/IP model
CO3	Explain the functions and various algorithms / protocols used of network layer of TCP/IP model
CO4	Describe the functions and various algorithms / protocols used of data link layer of TCP/IP model

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Internet: Description - Protocol - Network Edge - Network Core. Performance Metrics: Delay - Loss - Throughput. Layered Architecture - The Internet protocol stack - OSI reference model. Application Layer: Network Application Architectures - Communication of Processes - Services for Transport - Application Layer Protocols. Web and HTTP: HTTP - Persistent and Non Persistent - HTTP Message Format. Socket Programming with UDP and TCP.
UNIT 2	Transport Layer: Services - Relationship between Transport and Data Link Layer - Multiplexing and Demultiplexing. UDP - Segment Structure - Checksum - Principles of Reliable Data Transfer - GBN - SR. TCP: Connection - Segment Structure - Segment and Acknowledgement Numbers - Reliable Data Transfer Flow Control - Congestion Control.
First series internal examination including theory + Laboratory if any	
UNIT 3	Network Layer: Forwarded Routing - Service Models - VC and Datagram Networks - Functioning

	of a Router. IP - IPV4 Addressing - Interface Address - Subnet - Netmask - CIDR - ICMP - IPV6 Addressing - Datagram Format - IP V4 to V6 Transition. Routing Algorithms: LS - DV. Intra AS Routing in Internet: RIP - OSPF
UNIT 4	Data Link Layer: Services - NIC. Error detection and Correction: CheckSum - Parity - CRC. Multiple Access Links and Protocols: Channel Partitioning Protocol - Random Access Protocols: ALOHA - CSMA - CSMA/CD. Link Layer Addressing: MAC Address - ARP. Ethernet: Frame Structure - Technologies. Link Layer Switches Vs Routers.
UNIT X	Cookies - SMTP - RTT Estimation - Inter AS Routing on the Internet: BGP - IEEE 802.3 - Case Study: A Day in the Life of a Web Page Request - Data Centre Networking.
Second series internal examination including theory + Laboratory if any	

Books/References:

James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6th Edition.

Gerry Howser, Computer Networks and the Internet: A Hands-On Approach, Springer, ISBN-13: 9783030344955, 2019.

A. Tanenbaum and D. Wetherall, Computer Networks, 5th edition, Pearson, ISBN-13: 9780132126953, 2013.

TEACHING LEARNING STRATEGIES

- **Lecturing, Demonstration, Digital Learning, Team Work**

MODE OF TRANSACTION

- **Lecture, Seminar, Discussion**

ASSESSMENT RUBRICS

Refer to section 7.

Semester IV

KU04DSCMCA205 SYSTEM SOFTWARE AND OPERATING SYSTEM

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSC	200	KU04DSCMCA205	4	60

Learning Approach (Hours/ Week)	Marks Distribution	Duration of

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	ESE (Hours)
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description: This course is to provide students with basic knowledge of various system software. Particular emphasis will be given to major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management concepts (segmentation, paging, swapping), file systems, I/O systems and mass storage structure. Introducing open software and Linux basic commands.

Course Objectives:

- To know the design and implementation of system software.
- To explain the main components of OS and their working
- To familiarize the operations performed by OS and various scheduling policies
- To teach the different memory management techniques
- To explain file system, mass storage structure and input/output management

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Identify and Learn different system software
CO2	Describe the types, structure, functions and major concepts of Operating Systems
CO3	Understand the concepts of process, process synchronization, CPU scheduling, deadlocks and file system concepts
CO4	Explain LINUX file system concepts and simple commands

Mapping of COs to PSOs

CO - PSO Mapping

	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT1 05 Hours Theory + 05 Hours Lab	System programming – Assemblers, linkers, loaders and compiler (basic ideas). Introduction to compilers: Assemblers: Elements of Assembly Language Programming, Overview of Assembly Process, Design of Two pass Assembler, Macros and Macro Processors, Macro definition, call and expansion, Nested Macro calls, Advanced Macro facilities, Linkers: Linking and Relocation concepts, Design of linkers, Self relocating programs, Linking for overlays. Loaders: introduction to loaders - functions of loaders- Compilers: Introduction to compiler
UNIT 2 10 Hours+ 05 Hours Lab	Introduction: Types of OS - Batch Processing System – Multi programming system - Time Sharing System – Real Time System. Operating System Concepts, System Calls – Operating-System Operations -Process management, Memory management, Storage management, Operating system structures - System components, Operating systems services, System calls, Types of system calls
First series internal examination including theory + Laboratory if any	
UNIT 3 15 Hours + 10 Hours Lab	Processes: Process concept, Process scheduling, Operations on processes, Inter-process communication. Overview of threads. Process Synchronization: Critical-Section Problem, Semaphores. CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms - First come First Served, Shortest Job First, Priority scheduling, Round robin scheduling
UNIT 4 15 Hours Theory+ 10 Hours Lab	Deadlocks: System Model, Necessary conditions, Methods for Handling Deadlocks, Deadlock prevention, Deadlock avoidance - Banker's algorithms, Deadlock detection, Recovery from deadlock. Memory Management: Concept of address spaces, Swapping, Contiguous memory allocation, Segmentation, Paging. Virtual memory, Demand paging, Page replacement algorithms
UNIT X	File System: File concept, Access methods, Tree-structured directories, File system mounting, Protection. File System Implementation: File System structure, implementation. Unix : History of Unix OS, Open Source Software - Issues, Portability, Documentation, Best Practices for Working with Open Source Developers, Varieties of Open Source Licenses, Free Software vs Open Source software. Understanding File system, File Ownership and Permission. Shell - Types, Responsibilities. Basic Commands - cd, mkdir, echo, ls, pwd, rm, who, date, cp, mv, cat, ps. Working with Directories, Standard Input/Output, and I/O Redirection, pipes
Second series internal examination including theory + Laboratory if any	

Books/References:

D.M. Dhamdhere, Systems Programming and Operating Systems, TMH, 2003.

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2013). Operating System Concepts, 9th edition, John Wiley & Sons.

Wood, P., Kochan, S. G. (2016), Shell Programming in Unix, Linux and OS X, 4th edition, Pearson Education.

Andrew S. Tanenbaum, Herbert Bos (2016). Modern Operating Systems, 4th edition, Pearson Education India

William Stallings (2018), Operating systems - Internals and Design Principles, 9th Edition, Pearson Education, PHI.

Raymond, E. S. (2009). The Art of UNIX Programming, 3rd Edition, Pearson Education.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU04DSCMCA206 SOFTWARE ENGINEERING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSC	300	KU04DSCMCA206	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description: This course provides an overview of Software Engineering. It provides the fundamental aspects of the topic and covers the various process models available for creating software projects. It provides insight into the requirement engineering and discusses various techniques for modeling requirements. Fundamentals of software design and component level design in particular is also covered. The course also provides insight into the principles of user interface design, project estimation, scheduling, and testing strategies.

Course Objectives:

- Introduce the basic concepts of software engineering.
- Give basic idea of various software process models.
- Familiarize various processes of software development.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand what software engineering is and its importance, characteristics of software, and challenges in the field.
CO2	Explore various software development life cycle models and the principles of agile methodologies.
CO3	Learn techniques for requirements gathering, analysis, and modeling; understand the fundamentals of software design and architecture.
CO4	Discuss strategies for ensuring software quality, introduction to software testing methodologies, and understanding the importance of software security

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔

CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	Introduction to Software Engineering, Nature of Software, Software Application Domains, Legacy Software (20 hours)
UNIT 2 **15 Hours**	Process Models, Agility and Agile Processes, Scrum, XP, Kanban, DevOps
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	Understanding Requirements, Requirements Modeling, Design Concepts, Architectural Design, Component-Level Design, User Experience Design
UNIT 4 **20 Hours**	Quality Concepts, Software Reviews, Software Quality Assurance, Software Security Engineering, Software Testing at Component and Integration Levels
UNIT X	Quality Concepts, Software Reviews, Software Quality Assurance, Software Security Engineering, Software Testing at Component and Integration Levels
Second series internal examination including theory + Laboratory if any	
Books/References: Pressman RS Maxim BR. Software Engineering: A Practitioner's Approach. 9th ed. McGraw-Hill Higher Education; 2019. Hans van Vliet, Software Engineering: Principles and Practice. Robert C. Martin and Martin Fowler,Agile Software Development: Principles, Patterns, and Practices.	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU04DSCMCA207 ARTIFICIAL INTELLIGENCE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSC	300	KU04DSCMCA207	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description: The objective of the course is to impart theoretical knowledge in the specialized area of Artificial Intelligence. The program focuses on building a comprehensive understanding on the fundamentals of Artificial Intelligence, ability to solve new problems, and a capacity to learn continually and interact with inter- disciplinary groups. This course has the potential to energies scientific and social advancement through technological innovation and entrepreneurship.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	To introduce basic principles that drive complex real-world intelligence applications
CO2	To introduce and discuss the basic concepts of AI Techniques
CO3	To understand how the animal (especially human) intelligence is mimicked by the machines usi algorithms.
CO4	To familiarize the basics of machine learning concept and for solving the real world problems that cannot be solved by conventional algorithms

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Introduction - Overview of AI applications. Introduction to representation and search. The Propositional calculus, Predicate Calculus, Using Inference Rules to produce Predicate Calculus expressions, Application.
UNIT 2	Introduction to structure and Strategies for State Space search, Graph theory, Strategies for state space search, using the State Space to Represent Reasoning with the Predicate calculus (Sate space description of a logical system, AND/OR Graph). Heuristic Search: introduction, Hill-Climbing and Dynamic Programming, The Best-first Search Algorithm, Admissibility, Monotonicity and Informedness.
First series internal examination including theory + Laboratory if any	
UNIT 3	Building Control Algorithm for State space search – Introduction, Production Systems, The blackboard architecture for Problem solving. Knowledge Representation – Issues, History of AI representational schemes, Conceptual Graphs, Alternatives to explicit Representation, Agent based and distributed problem solving. Strong Method Problem Solving –Overview of Expert System Technology, Rule Based Expert system, Model - Based, Case-Based and Hybrid Systems (Introduction to Model based reasoning, Introduction to Case Based Reasoning, Hybrid design), Introduction to Planning. Reasoning in Uncertain Situation(introduction), logic based Adductive Inference. Introduction to PROLOG.
UNIT 4	Machine Learning: Symbol Based – Introduction, Frame –work. The ID3 Decision tree Induction algorithm. Inductive bias and Learnability, Knowledge and Learning, Unsupervised learning, Reinforcement Learning, Machine Learning: Connectionist – Introduction, foundations, Perceptron learning. Machine learning: Social and emergent: Models, The Genetic Algorithm, Artificial Life and Social based Learning.
Second series internal examination including theory + Laboratory if any	

Books/References:

1. S. Russel and p. Norvig, Artificial intelligence – A Modern Approach, 3rdEdn, Pearson

2. D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990

3. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill

George F Luger, Artificial Intelligence – Structures and Strategies for Complex problem solving, 5thEdn, Pearson

George J Klir and Bo yuan, Fuzzy sets and fuzzy logic: Theory and Applications, Prentice Hall India 1995

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

Semester V

KU05DSCMCA301 FORMAL LANGUAGE AND AUTOMATE THEORY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSC	300	KU05DSCMCA301	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

Course Description:

This course introduces fundamental concepts in theoretical computer science, focusing on the study of formal models of computation and their properties. Topics include automata theory, formal languages, computability theory, and complexity theory.

Course Objectives:

- Introduce the theoretical foundations of computation.
- Familiarize formal models of computation such as finite automata, regular expressions, context-free grammars, Turing machines, etc.

- Explore the concepts of decidability and undecidability.
- Familiarize the hierarchy of computational complexity classes and their properties

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the concepts of regular expression, DFA, and NFA.
CO2	Explore the concepts of context-free grammar, push down automata, and how they help in language parsing.
CO3	Explore the concepts of turning machines which will provide insights into the design and analysis of algorithms.
CO4	Understand fundamental concepts of computability such as decidability and undecidability.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Introduction to the Theory of computation and Finite Automata: Three basic concepts: languages, grammar & automata. Some applications. Finite automata: Deterministic Finite Acceptors, Nondeterministic Finite Acceptors, Equivalence of deterministic and nondeterministic finite acceptors, Reduction of the number of states in finite automata – State Equivalence method, table filling algorithm

UNIT 2	Regular Languages and Regular grammars: Regular expressions, connection between regular expressions and regular languages- state elimination method, Arden's lemma, regular grammars. Properties of Regular Languages: closure properties of regular languages, identifying non regular Language
First series internal examination including theory + Laboratory if any	
UNIT 3	Context-free grammars & languages: Context-free grammars, parsing and ambiguity. Simplification of Context free Grammars and Normal forms : methods of transforming grammars, two important normal forms – CNF & GNF Pushdown automata for context-free languages Non deterministic pushdown automata, PDA and context-free languages, deterministic pushdown automata and deterministic context-free languages. Properties of Context-Free Languages: pumping lemmas for context free languages, closure properties for context-free languages.
UNIT 4	Turing machine: Standard Turing machine, combining Turing machines for complicated tasks, Turing's thesis. Other models of Turing machine : Minor variations on the Turing machine theme, Turing machine with complex storage, nondeterministic Turing machine, a universal Turing machine, Linear bounded automata. A Hierarchy of Formal Languages and Automata: Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Grammars and Languages, The Chomsky Hierarchy.
UNIT X	Mathematical preliminaries and notation, Proof techniques -Pumping lemma for linear languages-Limits of Algorithmic computation: Problems that cannot be solved by Turing machines, Undecidable Problems for Recursively enumerable Languages, The Post Correspondence problem-Turing Machine Models and Complexity: Language Families and Complexity Classes, The Complexity Classes P and NP-Finite State Transducer – Mealy Machines and Moore Machines.
Second series internal examination including theory + Laboratory if any	
Books/References:	
1. An introduction to Formal Languages and Automata, Peter Linz, 4thedn, Narosa publishing House 2. John C Martin, Introduction to Languages and the Theory of Automata, McGraw Hill 1997 3. Mishra &Chandrasekharan, Theory of Computer Science: Automata, Languages andComputation, 3rd edn, PHI 4. Hopcroft, Motwani and Ullman, Introduction to automata theory, Languages and Computation, 3rdEdn,., Pearson	

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Refer to section 7.

KU05DSCMCA302 ANALYSIS AND DESIGN OF ALGORITHMS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSC	300	KU05DSCMCA302	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration : 2 hours for theory and 3 hours for Lab**

Course Description: The objective of course is to impart theoretical knowledge in the specialized area of algorithm design and analysis. Study of algorithms is very substantial in classification of problems and their solutions based on complexity. Analysis of algorithms provides a means for choosing an appropriate algorithm for solving a problem at hand. The course provides an insight into all aspects of computational complexity and the use, design, analysis and experimentation of efficient algorithms. The better understanding paves way for successful implementations in various scientific applications. The course will focus on various advanced paradigms and approaches used to design and analyze algorithms.

Course Objectives:

- To introduce basic principles that drive various algorithm design strategies
- Discuss the Complexity Analysis Techniques
- Overview of P, NP Problems
- Discuss about the concept of Design and Analysis of Parallel Algorithms

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Accomplish Knowledge about important computational problems and acquire knowledge to design the algorithm.
CO2	Obtain knowledge to analyze algorithm control structures and solving recurrence.
CO3	Attain information about Complexity Classes
CO4	Accomplish knowledge about Parallel Algorithms

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔		✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔		✔

COURSE CONTENT

UNIT 1 **15 Hours**	Algorithm Design: Introduction, Steps in developing algorithm, Methods of specifying an algorithm, Decisions prior to designing: based on the capabilities of the device, based on the nature of solutions, based on the most suitable data structures. Important Problem Types: Sorting, Searching, String processing, Graph problems, Combinatorial problems, Geometric problems and Numerical problems. Basic Technique for Design of Efficient Algorithm: Brute Force approach (String matching), Divide-and-Conquer approach (Merge sort), Branch-and-Bound technique (Knapsack problem). Greedy approach (Kruskal's algorithm and Prim's Algorithm), Dynamic Programming (Longest Common Subsequence), Backtracking(Sum of subsets problem) (20 Hours)
UNIT 2 **15 Hours**	Algorithm Analysis: Importance of algorithm analysis, Time and Space Complexity. Growth of Functions: Asymptotic notations, Cost estimation based on key operations- Big Oh, Big Omega, Little Oh, Little Omega and Theta notations, Big Oh Ratio Theorem, Big Theta Ratio Theorem, Big Omega Ratio Theorem. Analyzing Algorithm Control Structures, Solving Recurrences: Iteration Method, Substitution Method, The Recursion Tree Method, Master's Theorem, Problem solving using Master's Theorem Case 1, Case 2 and Case 3. Analysis of Strasser's algorithm for matrix multiplication, Analysis of Merge sort
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	Complexity- Complexity Classes: P, NP, NP Hard and NP Complete problems. NP Completeness reductions for Travelling Salesman Problem and Hamiltonian Cycle. P versus NP problem
UNIT 4 **15 Hours**	Design and Analysis of Parallel Algorithms: PRAM models – EREW, ERCW, CREW and CRCW, Relation between various models, Handling read and write conflicts, work efficiency, Brent's theorem. Analyzing Parallel Algorithms: Time Complexity, Cost, Number of Processors, Space Complexity, Speed up, Efficiency, Scalability, Amdahl's Law. Euler Tour Technique,

	Parallel prefix computation, Parallel merging and sorting.
UNIT X	Case study with examples
Second series internal examination including theory + Laboratory if any	

Books/References:

1. Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest, Introduction to Algorithms, 3rd Edition, Prentice Hall of India Private Limited, New Delhi

2. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley

3. Pallaw, V K, Design and Analysis of Algorithms, Asian Books Private Ltd, 2012.

4. Razdan S, Fundamentals of Parallel Computing, Narosa Publishing House, 2014.

5. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013

6. Upadhyay, N, Design and Analysis of Algorithms, SkKataria& Sons, 2008.

7. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley,

8. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India

9. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill

10. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd

11. Oded Goldreich, P,NP and NP- Completeness, Cambridge University Press, 2011.

12. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms,Volume- 1,Addison Wesley, 1997.

13. Sanjeev Arora and Boaz Borak, Computational Complexity- A Modern Approach,Cambridge University Press; 2009.

14. Daniel Hills W and Bruce M Boghosian, Parallel Scientific Computation, Science, Vol 261, Pp. 856-863

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Lab session

MODE OF TRANSACTION

- Lecture and Lab, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Refer to section 7.

KU05DSCMCA303 INTRODUCTION TO IOT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSE	300	KU05DSCMCA303	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course introduces the fundamentals of the Internet of Things (IoT), covering its definition, significance, current technology trends, various IoT protocols, and the relationship between IoT, cloud computing, and fog computing.

Course Objectives:

- To understand the definition and significance of the Internet of Things.
- To gain insights about the current trends associated with IOT technologies.
- To build a conceptual understanding of basic Arduino programming.
- To promote awareness regarding various protocols used in IoT.
- To explore the relationship between IoT, cloud computing and fog computing.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the fundamentals of the Internet of Things and its building blocks along with their characteristics and its domain.
CO2	Learn the architecture, operation, and business benefits of an IoT solution.
CO3	Understand various protocols, and emerging computing paradigms in IoT.
CO4	Understand cloud computing and it importance for building IoT applications

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours Theory** **+05 Hours Lab**	**Introduction to The Internet of Things:** IoT Definition, Elements of an IoT ecosystem, IoT applications, trends and implications, Major components of IOT(Hardware & Software). **IoT Sensing and Actuation:** Introduction, Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics. **Introduction to Arduino Programming**: Integration of Sensors and Actuators with Arduino programming
UNIT 2 **10 Hours Theory+ 05 Hours Lab**	**IoT Architecture and Communication:** Protocol Layered Architecture for IoT, Protocol Architecture of IoT, Infrastructure Protocols: MAC protocols for sensor network, S-MAC, IEEE 802.15.4, Near Field Communication (NFC), RFID, ZigBee, Bluetooth Low Energy (BLE), IPv6 over LowPower Wireless Personal Area Networks (6LoWPAN), Long Term Evolution-Advanced, Z-Wave, Components of ZWave Network, **Protocols for IoT Service Discovery**: DNS service discovery, multicast domain name system.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours Theory+ 10**	**IoT Networking Protocol**: Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), Advanced Message

Hours Lab	Queuing Protocol (AMQP), Data Distribution Service (DDS), Service Discovery Protocols, Routing Protocol for Low Power and Lossy Networks (RPL), sensor networks, unique constraints and challenges, advantages of ad-hoc/sensor network, sensor network architecture, data dissemination and gathering protocol.
UNIT4 10 Hours Theory + 10 Hours Lab	**Cloud Computing**: Introduction, Virtualization, Cloud Models, Service-Level Agreement in Cloud Computing, Cloud Implementation, Sensor-Cloud: Sensors-as-a-Service, **fog computing.**
UNIT X	**The IoT Data Analytics Platforms**: IBM Watson IoT Platform, Splunk Software for IoT Data, Amazon Web Service IoT Platform, Azure IoT Hub, The IoT Data Virtualization Platforms, IoT Data Visualization Platform, Security and Privacy in IoT. **IoT Case Study:** Agriculture, Healthcare, Process Automation & monitoring etc.
Second series internal examination including theory + Laboratory if any	

Books/References:

1. SudipMisra, Anandarup Mukherjee, Arijit Roy, “Introduction to IoT”, Cambridge University Press 2021.
2. Pethuru Raj and Anupama C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press, 2017.
3. Simone Cirani, Gianluigi Ferrari, Marco Picone, Luca Veltri, “Internet of Things: Architectures, Protocols and Standards,” Wiley, 2018.
4. Simone Cirani, Gianluigi Ferrari, Marco Picone, Luca Veltri, “Internet of Things: Architectures, Protocols and Standards,” Wiley, 2018.
5. Fei Hu, “Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations,” CRC Press, 2016.
6. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-Approach)”,1st Edition, VPT, 2014.
7. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015 3. Editors OvidiuVermesan
8. Peter Friess,'Internet of Things – From Research and Innovation to Market Deployment', River Publishers, 2014
9. Fei Hu, “Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations,” CRC Press, 2016.

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning

MODE OF TRANSACTION

- Lecture and Lab, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Refer to section 7.

KU05DSCMCA304 MACHINE LEARNING TECHNIQUES

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSC	300	KU05DSCMCA304	4	90

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	4	1	50	50	100	2(T)+3(P)*

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours** **Theory +**	Introduction to Machine Learning: Concept of the learning task, inductive learning, and the concepts of hypothesis space, introduction to different types of machine learning approaches, examples of machine learning applications, different types of learning: supervised learning, unsupervised learning, reinforcement learning. Feature Extraction- Setting up your machine

05 Hours Lab	learning platform: training, validation and testing, overfitting and under-fitting, different types of error calculation.
UNIT 2 05 Hours + 15 Hours Lab	Regression- Linear regression and logistic regression- Supervised Learning: Introduction, learning a class from example, learning multiple classes, Unsupervised Learning: Introduction, clustering; mixture densities, k-means clustering and hierarchical clustering.
First series internal examination including theory + Laboratory if any	
UNIT 3 10 Hours + 20 Hours Lab	Dimensionality reduction: principal component analysis, linear discriminant analysis, canonical correlation analysis. Introduction to Artificial Neural Network: Understanding brain, perceptron, Multi-Layer perceptron as universal approximator, general architecture of artificial neural network, feed forward and back propagation, different linear and nonlinear activation functions for binary and multi class classification.
UNIT 4 10 Hours + 20 Hours Lab	Introduction to Deep Learning: Fundamentals of deep learning, Deep Feedforward Networks, Regularization for Deep Learning, Optimization for Training Deep Models, Introduction to Convolutional Networks, Sequence Modelling using Recurrent Nets, overview of LSTM, fundamentals of Generative adversarial Networks.
UNIT X	Feature Extraction - Testing the quality of features Reinforcement Learning, Transfer Learning Anomaly Detection and Outlier Analysis, Natural Language Processing (NLP), Graph Neural Networks (GNNs), Adversarial Machine Learning
Second series internal examination including theory + Laboratory if any	
Books/References:	
1. EthemAlpaydin, Introduction to Machine Learning- 3rd Edition, PHI.	
2. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning), MIT Press, 2016.	
3. Tom M. Mitchell, Machine Learning, McGraw-Hill	
4. Kuntal Ganguly, Learning Generative Adversarial Networks, Packt Publishing, 2017.	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

Semester VI

KU06DSCMCA305 BIG DATA ANALYTICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	KU06DSCMCA305	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	C	ESE	Total	
4	0	1	50	50	100	2(T)+3(P)*

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: The primary intention of this course is to make aware about the basic concepts of big data platform and its related application. The course explore the concept of HDFS file system and developing and deploying of Big Data applications using Hadoop and MapReduce. Apart from this, course also explore different developing and deploying various tools for developing applications using HIVE and Pig.

Course Objectives:

- Aims to impart basic skills to the learners for developing mathematical models for big data analysis.
- Acquire the knowledge and architecture of HDFS system for developing and deploying big data applications
- Impart knowledge on fundamental and advanced data structure concepts and techniqus suitable for big data analysis.
- Aim to impart to knowledge in bigdata platforms and developing mathematical/statistical models for big data analysis using machine learning concepts.

Course Outcome:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Familiarize the concept of Big Data platforms and different tools for handling big data.
CO2	Understanding the concept of HDFS and developing MapReduce applications
CO3	Aware about the setting up of Hadoop platform and its configuration for organising Hadoop clusters
CO4	Acquire the knowledge of developing and deploying various tools for developing applications of Big Data platform.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1	Introduction to Big Data Platform – Challenges of Conventional Systems - Intelligent data analysis –Nature of Data - Analytic Processes and Tools - Analysis vs Reporting - Modern Data Analytic Tools -Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error.
UNIT 2	The Hadoop Distributed File System – Components of HadoopAnalyzing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfaces to HDFS Basics-

	Developing a Map Reduce Application, How Map Reduce Works-Anatomy of a Map Reduce Job Run-Failures-Job Scheduling-Shuffle and Sort – Task execution - Map Reduce Types and Formats – specification
First series internal examination including theory + Laboratory if any	
UNIT 3	Cluster Setup and Installation – Hadoop Configuration-Security in Hadoop - Administering Hadoop – HDFS - Monitoring-Maintenance- Hadoop benchmarks- Hadoop in the cloud. Map Reduce Features. Setting up a Hadoop Cluster
UNIT 4	Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive – fundamentals of HBase and Zookeeper - IBM InfoSphereBigInsights and Streams. Visualizations -Visual data analysis techniques, interaction techniques, Systems and applications.
UNIT X	: Introduction To Streams Concepts – Stream Data Model and Architecture - Stream Computing - Sampling Data in a Stream – Filtering Streams – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window - Real time Analytics Platform (RTAP), Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.
Second series internal examination including theory + Laboratory if any	

Books/References:

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
2. Tom White, Hadoop: The Definitive Guide, 3rdEdn, O'reily Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Understanding BigData: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw-Hill Pub, 2012.
4. AnandRajaraman& Jeffrey D Ullman, Mining of Massive Datasets, Cambridge University Press,2012.
5. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & Sons, 2012.
6. Glen J. Myyat, Making Sense of Data, John Wiley & Sons, 2007.
7. Pete Warden, Big Data Glossary, O'Reily, 2011.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSCMCA306 QUANTUM COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSC	300	KU06DSCMCA306	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: Quantum computing is a rapidly advancing field that promises to revolutionize computation and problem-solving. This course provides an introduction to quantum computing tailored specifically for non-physics students. It covers the fundamental principles of quantum mechanics and explores how quantum phenomena can be harnessed to perform powerful computations. No prior knowledge of physics or advanced mathematics is required

Course Objectives:

- Understand the basic principles of quantum mechanics.
- Explore the fundamental concepts of quantum computing.
- Gain knowledge of quantum algorithms and their applications.
- Develop an understanding of quantum gates and quantum circuits.
- Explore the challenges and potential of quantum computing.
- Gain familiarity with quantum programming languages and frameworks.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Develop a foundational understanding of quantum mechanics and its relevance to computing.

CO2	Gain knowledge of quantum algorithms and their potential applications.
CO3	Acquire practical skills in quantum programming and working with quantum circuits.
CO4	Understand the challenges and future directions in quantum computing research.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours** **Theory**	Introduction to Quantum Mechanics -Overview of classical computing and its limitations - Introduction to quantum mechanics -Wave-particle duality-Superposition and measurement-Quantum entanglement (10 Hours)
UNIT 2 **10 Hours** **Theory + 10 Hours Lab**	Quantum Computing Basics -Introduction to qubits and quantum gates -Quantum states and quantum operations - Quantum circuits and circuit model of Computation-Measurement and quantum measurement postulate (15 Hours)
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours** **Theory +** **10 Hours Lab**	Quantum Algorithms-Quantum parallelism and superposition – The Deutsch-Jozsa algorithm – Grover's algorithm for unstructured search -Shor's algorithm for factoring large numbers (20 Hours)
UNIT 4 **10 Hours** **Theory +** **10 Hours Lab**	Quantum Simulators and Hardware- Overview of quantum simulators and their role in quantum computing research Introduction to quantum hardware (e.g., qubits, quantum gates) – Comparison of different quantum computing technologies (e.g., superconducting qubits, trapped ions) – **Quantum Programming Languages and Frameworks**- Introduction to quantum programming languages (e.g., Q#, Qiskit) – Basics of quantum program structure and execution – Quantum gates and operations in programming languages – Hands-on exercises using quantum programming frameworks

UNIT X	Advanced Quantum Computing • Quantum Fourier transform and its applications • Shor's factorization algorithm and quantum cryptography • Quantum simulation and quantum machine learning • Introduction to quantum error correction Quantum Computing Applications • Quantum computing in cryptography and secure communication • Quantum optimization and quantum annealing • Quantum simulation of physical systems • Potential impact of quantum computing on various industries Programming Quantum Computers • Introduction to quantum software development kits (SDKs) • Quantum programming languages: Qiskit, Cirq, and PyQuil • Building and running quantum circuits (10 Hours)
Second series internal examination including theory + Laboratory if any	

Books/References:

1. Quantum Computing for Everyone" Author: Chris Bernhardt Publisher: The MIT Press Year: 2019.
2. Programming Quantum Computers: Essential Algorithms and Code Samples" by Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia (O'Reilly Media).
3. Quantum Computing for Computer Scientists" by Noson S. Yanofsky and Mirco A. Mannucci (Cambridge University Press).
4. Quantum Computing: An Applied Approach" by Jack D. Hidary (Springer)

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration, Digital Learning, Team Work

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Refer to section 7.

KU06DSCMCA307 GENERATIVE AI

Semester	Course Type	Course Level	Course Code	Credits	Total Hours

6	DSC	500	KU06DSCMCA307	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This syllabus serves as a comprehensive guide for understanding and exploring the principles, techniques, and applications of Generative AI. The syllabus is designed to provide the necessary foundation and resources to explore the fascinating realm of generative models. By studying the concepts and techniques outlined in this syllabus, the student will gain the knowledge and skills to harness the power of generative models and contribute to the advancement of AI research and applications.

Course Objectives:

- Discuss about the bases of generative artificial intelligence
- Application programming interfaces for generative AI.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Knowledge about theoretical and conceptual bases of generative AI

CO2	Knowledge about the application space for generative AI
CO3	Knowledge about how generative AI should be able to support creativity
CO4	Ability to use generative AI to create content as texts, images, number, music and video.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2		✔	✔	✔	✔
CO3	✔		✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **05 Hours**	Introduction to Generative AI:Definition and scope of Generative AI. Overview of generative models and their applications, Importance of Generative AI in various domains, Brief discussion on ethical considerations and challenges.(15 hours)
UNIT 2 **15 Hours Theory** **+ 05 Hours Lab**	Language Models and LLM Architectures: Introduction to language models and their role in AI, Traditional approaches to language modeling, Deep learning-based language models and their advantages, Overview of popular LLM architectures: RNNs, LSTMs, and Transformers, Taxonomy of Transformers.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours** **10 Hours Lab**	Generative Pre-trained Transformer: Introduction to Generative Pre-trained Transformer and its significance, Pre-training and fine-tuning processes in GPT, Architecture and working of GPT models, Overview of GPT variants and their use cases.
UNIT 4 **10 Hours Theory** **+ 15 Hours Lab**	Practical Application of Generative Pre-trained Transformer: Introduction to ChatGPT and its purpose, Training data and techniques for ChatGPT, Handling user queries and generating responses,Tips for improving ChatGPT's performance.

UNIT X	Simplifying Development with Language Models: Introduction to LangChain and its objectives, Overview of the LangChain framework and its components Streamlining application development using LangChain Examples of applications built with LangChain. Prompt Engineering: Enhancing Model Outputs: Understanding the concept and significance of prompt engineering, Strategies for designing effective prompts, Techniques for controlling model behavior and output quality. Best practices for prompt engineering in generative AI.

Second series internal examination including theory + Laboratory if any

Books/References:

1. Luger, George F., and William A. Stubblefield. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Pearson, 2004.
2. Cox, Michael T. Theoretical Models of Reasoning under Uncertainty. Kluwer Academic Publishers, 1988.
3. Vaswani Ashish et al, Attention is all you need. Proc. of NeurIPS (2017), pp.5998-6008.
4. A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. arXiv preprint arXiv:2205.09712, 2022.
5. M. Castelli and L. Manzoni, "Generative models in artificial intelligence and their applications," vol. 12, ed: MDPI, 2022, p. 4127.
6. Radford, A., Narasimhan, K., Salimans, T., &Sutskever, I. (2018). Improving language understanding by generative pre-training. Retrieved fromhttps://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf
7. S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. EMNLP, 2015.
8. R. T. Hughes, L. Zhu, and T. Bednarz, "Generative adversarial networks–enabled human–artificial intelligence collaborative applications for creative and design industries: A systematic review of current approaches and trends," Frontiers in artificial intelligence, vol. 4, p. 604234, 2021.
9. G. Harshvardhan, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," Computer Science Review, vol. 38, p. 100285, 2020.
10. S Minaee, T Mikolov, N Nikzad, M Chenaghlu, Large language models: A survey,https://arxiv.org/pdf/2402.06196
11. Khurana, D., Koli, A., Khatter, K. & Singh, S. Natural language processing: state of the art, current trends and challenges. Multimed. Tools Appl. 82, 3713–3744 (2023).
12. G Adomavicius,A Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE transactions on knowledge and data engineering, 2005.
13. BabakAmiri, NikanShahverdi, AmiraliHaddadi, Yalda Ghahremani, "Beyond the Trends: Evolution and Future Directions in Music Recommender Systems Research", *IEEE Access*, vol.12, pp.51500-51522, 2024.
14. Ronakkumar Patel, PriyankThakkar, Vijay Ukani, "CNNRec: Convolutional Neural Network based recommender systems - A survey", Engineering Applications of Artificial Intelligence, vol.133, pp.108062, 2024.
15. MacNeil, S., Tran, A., Mogil, D., Bernstein, S., Ross, E., & Huang, Z. (2022). Generating
16. diverse code explanations using the GPT-3 large language model. Proceedings of the ACM Conference on International Computing Education Research, 2, 37-39.
17. Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT Press.

18. Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. Neurocomputing, 452, 48-62.
19. OpenAI. (2022). OpenAI about page. Retrieved from https://openai.com/about/
20. Pavlik, J. V. (2023). Collaborating with ChatGPT: Considering the implications of generative artificial intelligence for journalism and media education. Journalism and Mass Communication Educator. https://doi.org/10.1177/10776958221149577
21. Radford, A., Narasimhan, K., Salimans, T., &Sutskever, I. (2018). Improving language
22. understanding by generative pre-training. Retrieved from
23. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

	S4 -List of Discipline Specific Electives (DSE-1)											
No	Level	Course Code	Course Name	Total Hours	C	Hrs./wk.			Assessment Weightage (%)		
						L	P	Tt	ESA	CE	T
4E.1	200	KU04DSEMCA208	Python Programming	60	4	4	0	1	50	50	100
4E.2	200	KU04DSEMCA209	Discrete Mathematics	60	4	4	0	1	50	50	100
4E.3	200	KU04DSEMCA210	Mobile Computing	60	4	4	0	1	50	50	100
4E.4	200	KU04DSEMCA211	Operating System Security	60	4	4	0	1	50	50	100
4E.5	200	KU04DSEMCA212	Design Thinking	60	4	4	0	1	50	50	100

KU04DSEMCA208 PYTHON PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSE	200	KU04DSEMCA208	4	75

Learning Approach (Hours/ Week)	Marks Distribution	Duration of ESE

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	(Hours)
3	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course mainly focuses on introducing the fundamental programming concept to the students from other disciplines. This course mainly focuses on the basic concept of programming constructs such as language elements, data types, operators control structures and data structure that will be useful for the programmers to learn the basic programming concept. Structure of this course is well organized to introduce the programing basic concept to advanced concepts such as modules, packages, GUI, and basics of NLP concepts for pattern matching and searching. After completing this course, the students acquire the ability to develop real life applications commonly useful for society in many walks of life.

Course Objectives:

- Aims to impart basic programming skills to the learners in a simplest way.
- Impart knowledge on fundamental and advanced data structure concepts.
- Acquire the knowledge to impart various control structures to implement programming logic.
- Aware about the development of common GUI based applications in simple steps.
- Acquire the ability to analysis of data using NumPy and Pandas

Course Outcome

At the end of the Course, the Student will be able to:

SL #	**Course Outcomes**
CO1	Familiarize the different parts of Python programming and its applications in real world problems
CO2	Understanding the concept of different control structures and functions in Python.
CO3	Aware about the various data types and it operations and supporting methods
CO4	Make aware about the advanced concepts in Python such as reading and writing of data files, NumPy, Pandas and GUIs.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT 1 **10 Hours**	**Introduction**: History of Python Programming, Thrust Areas Of Python, Installing Anaconda Python Distribution, PyCharm IDE and Jupyter Notebook, Creating And Running First Python Project, **Parts of Python Programming Language**-identifiers, keywords, statements and expressions, variables, operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversion, The *typedef()* function and Is operator
UNIT 2 **10 Hours**	**Control Flow Statement**- Decision control flow statement *(if, if …else, if…elif…*, nested *if*), Loop (*while*, *for*),*continue*, *break* statements, Catching Exception Using *try* and *except* Statement **Functions**- Built-In Functions Commonly used Modules, Function definition and calling the function, The *return* statement and *void* function, scope and life time of variables.
First series internal examination including theory + Laboratory if any	
UNIT 3 **13 Hours**	**Lists-** Creating List, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions used on lists, list Methods. **Dictionaries**- Creating Dictionary, Accessing and Modifying *key:value* Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods. **Tuples and Sets-** Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions used on Tuples, Relation between Tuples and Lists, Sets and Set Methods, Frozenset. **Strings-** Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings
UNIT IV **12 Hours**	**Files**- Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary files, Reading and Writing CSV Files, Introduction to NumPy and Pandas. GUIs in Python: Root Window-Fonts and colors- Working with containers and canvas, Frames, Widgets, Button widgets, Arranging widgets in the Frame, Label Widget, Message Widget, Text

	widget, Scrollbar widgets, Check button widget, Radio button widget, Entry Widget, Spin box Widget, List Box Widget, Menu Widget
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

Alberto Fernandez Villan, Mastering OpenCV 4 with Python, Packt Publishing Ltd.

Dr. R Nageswara Rao, Core Python Programming, 2nd edition, Dreamtech Publisher, 2019

Core Suggested Readings

1. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st Edition, O'Reilly Media, 2017. ISBN – 13: 978-1491962299.
2. Wesley J. Chun, Core Python Programming, Second Edition, Publisher: Prentice Hall Pub

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU04DSEMCA209 DISCRETE MATHEMATICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSE	200	KU04DSEMCA209	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	0	50	50	100	2

Course Description: This course is intended to provide the mathematical foundations necessary for a computer science student. Mathematical concepts like elementary discrete mathematics, probability & statistics and linear algebra are included.

Course Objectives:

- Impart knowledge on mathematical logic.
- Give basic idea set theory, relations, functions and graphs and their problem solving.
- Familiarize counting principles and the use of recurrence relations to solve combinatorial and algorithmic problems in computer science
- Impart knowledge on probability and its distributions.
- Familiarize graph theory concepts to solve problems in computer science.

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Describe basic mathematical structures such as sets, functions, and relations, including their properties and operations. (Cognitive level: Understand)
CO2	Analyse and interpret the properties of functions and relations, and apply them to model and solve problems in computer science. (Cognitive level: Apply)
CO3	Apply counting principles and recurrence relations to solve combinatorial and algorithmic problems in computer science. (Cognitive level: Apply)
CO4	Apply graph theory concepts, including Euler and Hamiltonian graphs, to solve computational problems in computer science. (Cognitive level: Apply)

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2		✔	✔	✔	✔
CO3	✔		✔	✔	✔

CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours**	Propositional logic – Propositions, truth tables, converse, contra positive and inverse, compound statements and their truth tables, translating natural language sentences to logical statements, tautology, contradiction, logical equivalence, De Morgan's laws, normal forms. Predicate logic – predicates, universal and existential quantifiers, binding variables, translating natural language sentences to logical statements. **Sets**: representation of sets, set operations, Cartesian product, using set notation with quantifiers, truth sets of quantifiers, computer representation of sets. Functions – one-to-one and onto functions, inverse functions and compositions of functions. .
UNIT 2 **10 Hours**	Relations – properties, functions as relations, relations on a set, combining relations, n-ary relations and their applications, representing relations, closures of relations, Warshall's algorithm, equivalence relations, equivalence classes and partitions.
First series internal examination including theory + Laboratory if any	
UNIT 3 **13 Hours**	Basics of counting, basic counting principles, the inclusion-exclusion principle, the pigeonhole principle, the generalized pigeonhole principle, permutations and combinations, with and without repetitions. Generating permutations and combinations. Recurrence relations, modeling with recurring relations.
UNIT IV **12 Hours**	Graphs – definition, different types of graphs, graph models, basic terminology, representing graphs, isomorphism, connectivity, Euler and Hamilton paths, shortest path problem and Dijkstra's algorithm. Trees - basic terminology, properties (no proofs), spanning trees, depth-first and breadth-first searches.
Second series internal examination including theory + Laboratory if any	

Books:

1. Kenneth H. Rosen, Discrete Mathematics and Applications, TMH 2003
2. J.P. Tremblay and R Manohar Discrete Mathematical Structure with Applications to Computer Science, TMH 2001.
3. John Truss, Discrete Mathematics for Computer Scientists, Pearson Edn 2002
4. Sengadir, Discrete Mathematics, Pearson, 2009.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU04DSEMCA210 MOBILE COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSE	300	KU04DSEMCA210	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

The purpose of this course is to understand the concept of mobile computing paradigm, novel applications, limitations and also to impart knowledge on the typical mobile networking Infrastructure through a popular GSM protocol. It also helps the students to get exposed to Ad-Hoc Networks and gain knowledge about different mobile platforms and application development.

Course Objectives:

- To infer knowledge about mobile communications and its services.
- To Identify several communication access techniques.
- To illustrate technical format, addressing and transmission strategies of packets.
- To determine the functionality of MAC, Network layer and identifying a routing protocol for given Ad Hoc Networks and Perceive knowledge about TCP and failure recovery method.
- To identify and solve database issues using hoarding techniques.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Infer knowledge about mobile communications and its services.
CO2	Identifying several communication access techniques.
CO3	Illustrate technical format, addressing and transmission strategies of packets.
CO4	Determine the functionality of MAC, Network layer and identifying a routing. protocol for given Adhoc Networks and Perceive knowledge about TCP and failure recovery methods.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2		✔	✔	✔	✔
CO3	✔		✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours**	Introduction to Mobile Computing - Architecture of Mobile Computing - Novel Applications – Limitations. GSM - GSM System Architecture - Radio Interface – Protocols - Localization and Calling - Handover - Security - New Data Services. (20 hours)
UNIT 2 **10 Hours**	Medium Access Control Protocol - Wireless MAC Issues - Hidden and exposed terminals - near and far terminals – SDMA – FDMA – TDMA – CDMA - Tunnelling Cellular Mobility - IPv6. (20 hours)

First series internal examination including theory + Laboratory if any	
UNIT 3 13 Hours	Mobile IP – Goals – Assumption - Entities and Terminology - IP Packet Delivery - Agent Advertisement and Discovery – Registration - Tunnelling and Encapsulation – Optimizations - Dynamic Host Configuration Protocol. (20 hours)
UNIT IV 12 Hours	Traditional TCP - Indirect TCP - Snooping TCP - Mobile TCP - Fast Retransmit and Fast Recovery - Transmission /Time-Out Freezing - Selective Retransmission - Transaction Oriented TCP.(20 hours)
Second series internal examination including theory + Laboratory if any	
UNIT X	Hoarding Techniques - Caching Invalidation Mechanisms - Client Server Computing with Adaptation- Power Aware and Context Aware Computing - Transactional Models - Query Processing – Recovery - and Quality of Service Issues.

Core Compulsory Readings

1. Jochen Schiller, "Mobile Communications", Second edition Addison-Wesley, 2008.
2. Reza Behravanfar, "Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML", Cambridge University Press, October 2004.

Core Suggested Readings

1. Adelstein, Frank, Gupta, Sandeep KS, Richard III, Golden, Schwiebert, Loren, "Fundamentals of Mobile and Pervasive Computing", McGraw-Hill Professional, 2005.
2. Hansmann, Merk, Nickolas, Stober, "Principles of Mobile Computing", second edition, Springer, 2003.Martyn Mallick, "Mobile and Wireless Design Essentials", Wiley DreamTech, 2003.
3. Ivan Stojmenovic and Cacute, "Handbook of Wireless Networks and Mobile Computing", Wiley, 2002.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU04DSEMCA211 OPERATING SYSTEM SECURITY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSE	300	KU04DSEMCA211	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2(T)+3(P)*

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

With a focus on Windows and Linux operating systems in particular, this course covers basic operating system ideas, exploring fundamental concepts such as architecture, memory management, file system management, I/O management, network operating systems, and distributed operating systems. It delves into the installation, configuration, and security of both Windows and Linux. Additionally, the course examines cloud computing and virtualization topics including zero-trust security architecture, cloud provisioning, PaaS and SaaS models, private and public cloud environments, and cloud security. The Linux module also encompasses bash scripting.

A comprehensive understanding of operating systems, virtualization, cloud computing, and their corresponding security measures is provided within the course.

Course Objectives:

- Provide a foundational understanding of operating systems, their functions, and their importance in computer systems

- Gain knowledge of the underlying architecture and components of the Windows and Linux operating systems
- Learning Windows and Linux operating system features—system administration and security
- Understand the fundamentals of virtualization and cloud computing

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand how to install and update Windows 11 and Server 2019, Linux and configure GRUB boot loader
CO2	Install, configure, manage, and maintain active directory domain services
CO3	Configure basic Linux network services
CO4	Understand Cloud Computing, Cloud Types and Cloud Service Deployment Models (PaaS, SaaS) and the concept of Cloud Security

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours**	Fundamental operating system concept: Basic Concepts of Operating system – Architecture – Process management- Memory management- File system management- I/O Management – Basics of Network Operating Systems – Basics of Distributed Operating system (10 hours)
UNIT 2 **15 Hours**	Windows Operating system: History of Windows OS and Fundamentals - Installation of Windows Operating System Windows 11 and Server 2019 - Updating windows system and patch management-Windows File system and Disk Management- Active Directory and windows domain-Windows Security and OS Hardening-Virtualization using Hyper-V-Network Based installation method-Network Configuring- Implementation of infrastructure of windows networks- Active Directory Domain Services (ADDS)- DNS, DHCP and IPAM, Network Policy Server (NPS), Local Policies, Group Policies, Flexible Single Master Operation (FSMO)-File Server Resource Manager (FSRM), Windows Server Backup (WSB) (15 hours).
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	Linux Operating system: Linux Fundamentals – (Startup Files, Linux boot process), Basic Commands of Linux, Installation of Linux- Configuring the GRUB boot loader, Disk management and partition, Controlling and managing Services, -Linux Authentication method and User administration-Linux File System and permissions -Methods for installation of Packages. Apt-get and YUM/DNF Package manager. Network configurations- configuration of NFS, FTP and DHCP Servers. Linux Security and OS Hardening. Virtualization using KVM
UNIT IV **25 Hours**	Virtualization and cloud computing: Virtualization concepts - Cloud fundamentals – architecture - Private cloud environment-Public cloud environment - Auto-provisioning-Cloud as PaaS, SaaS-Cloud computing securitization
UNIT X	Power shell Scripting - Windows Administration using power shell, Background Jobs and Remote Administration - Bash Scripting, Introduction to BASH Command Line Interface (CLI) Error Handling Debugging & Redirection of scripts, Automate Task Using Bash Script, Security patches, Logging & Monitoring using script - Ethics and standard of cloud Cloud Data Security - Cloud Application Security - Zero Trust Security Architecture
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Silberschatz, Greg Gagne, and Peter Baer Galvin, Operating System Principles, 7th Edition
2. Richard Peterson, Linux: The complete reference, 6th Edition
3. DacNhuong L, RaghvendraKumar, Virtualization and Cloud Computing, John Wiley and Sons

4. Richard Fox, Linux with operating system concepts
5. Christal Panek, Windows operating system fundamentals

Core Suggested Readings

1. Bruce Payette, Windows PowerShell in Action
2. Harlan Carvey, Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry
3. Christopher Negus, Linux Bible
4. Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley, UNIX and Linux System Administration Handbook
5. Jeremy Moskowitz, Group Policy: Fundamentals, Security, and the Managed Desktop
6. Richard Blum and Christine Bresnahan, Linux Command Line and Shell Scripting Bible.

TEACHING LEARNING STRATEGIES

- Hands-on-oriented teaching, collaborative learning, case studies and project presentations, peer-led discussions.

MODE OF TRANSACTION

- Lecture, seminar, discussion, audio and video presentation, demonstration, practical assignments, and exercises.

KU04DSEMCA212 DESIGN THINKING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	DSE	300	KU04DSEMCA212	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE duration : 2 hours for theory and 3 hours for Lab**

Course Description: The course provides an introduction to Design Thinking, a human-centered

problem-solving methodology used to address real-world challenges and create innovative solutions. Through lectures, discussions, activities, case study discussions, and exploration of various tools and methodologies, students will gain a deeper understanding of design thinking's core principles and its practical application.

Course Objectives:

- Impart knowledge on design thinking, process and mindsets
- Impart knowledge on problem space and solution space
- Familiarize various tools and methodologies used as part of design thinking process.
- Impart knowledge on product design, design sprints.
- Familiarize with real world case studies where design thinking has been applied.

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the basics of design thinking process and mindsets
CO2	Utilize various tools and methodologies for user research, problem definition, and ideation
CO3	Define problems effectively, identify user needs, generate creative solutions, prototype and test ideas to gather user feedback
CO4	Apply design thinking to address real-world challenges

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔

CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **20 Hours**	Design Thinking: foundations, mindsets, process (empathize, define, ideate, prototype, test), benefits and applications Case studies of successful design thinking projects.
UNIT 2 **20 Hours**	Understanding the Problem Space: User research methods - interviews (in-depth, contextual), surveys (open-ended, closed-ended), observations (ethnographic, contextual inquiries). Building user empathy - empathy mapping, customer journey mapping, ecosystem mapping, actor mapping, personas. Identifying and Prioritizing User Needs: 5 Whys analysis, Kano model, user story writing. Problem Definition and Framing: "How-might-we" questions.
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	Exploring the Solution Space: Ideation techniques - brainstorming, brainwriting, worst possible idea, SCAMPER Refining Ideas: crazy 8s, design critiques, affinity diagramming Rapid Prototyping Principles and Techniques: Low-fidelity (paper, cardboard), medium-fidelity (digital tools), high-fidelity (advanced digital tools) User Testing and Feedback Iteration: Usability testing techniques, A/B testing, user feedback analysis
UNIT IV **20 Hours**	Design Thinking in Action: Product Design - User experience (UX) design principles, Information architecture (IA), Design for manufacturability Design Sprints and Other Rapid Innovation Methods: Design sprint framework, Lean Startup methodologies
UNIT X	Case Studies - Identifying the core challenge and user needs, developing a user-centered approach to address the problem, applying relevant design thinking tools and methodologies, iterating on solutions through prototyping and user testing. Case Study 1 Improving Accessibility in E-commerce Platforms (E-commerce Industry) Case Study 2 Enhancing User Engagement in a Learning Management System (Education Technology) Case Study 3 Streamlining the Onboarding Process for a Mobile Fitness App (Health & Wellness Technology) (10 hours)
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation by Tim

Brown (First Edition, 2009)

2. The Design Thinking Toolbox: A Guide for Mastering the Art of Innovative Problem-Solving by Michael Lewrick, Patrick Link, and Leslie Snyder (First Edition, 2018)

3. Sprint: Solving Big Problems and Testing New Ideas in Just Five Days by Jake Knapp (First Edition, 2016)

4. Design Thinking for Dummies by Gavin Ambrose and Paul Dunne (First Edition, 2018)

Core Suggested Readings

1. Creative Confidence: Unleashing the Creative Potential Within Us All by Tom Kelley (First Edition, 2013)

2. This Is Service Design Doing: Applying Service Design Thinking in the Real World by Marc Stickdorn (First Edition, 2018)

3. Design Thinking for Business Growth: How to Design and Scale Business Models and Business Ecosystems by Patrick van der Pijl, Alexander Osterwalder, and Yves Pigneur (First Edition, 2014)

TEACHING LEARNING STRATEGIES

●Lecturing, Demonstration, Digital Learning, Team Work

MODE OF TRANSACTION

●Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Refer to section 7

KU05DSEMCA305 COMPUTER GRAPHICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSE	300	KU05DSEMCA305	4	75

Learning Approach (Hours/ Week)	Marks Distribution	Duration of

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	ESE (Hours)
3	2	1	50	50	100	2

COURSE CONTENTS

UNIT 1 **10 Hours**	Basic Concepts in Computer Graphics. Input devices. Display devices. Line and circle drawing Algorithms.Solid area scan-conversion.Polygonfilling.Projections – Parallel, Perspective.Hidden Line Elimination Algorithms.Image processing – digital image representation – edge detection – Scene segmentation and labeling – region- labeling algorithm – perimeter measurement.Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling.Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates.Basic 3D transformations
UNIT 2 **10 Hours**	Projections – Parallel and perspective projections – vanishing points. Visible surface detection methods– Back face removal- Z-Buffer algorithm, A-buffer algorithm, Depth-sorting method, Scan line algorithm. Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland Hodgeman Polygon clipping algorithm.
First series internal examination including theory + Laboratory if any	
UNIT 3 **13 Hours**	3D Geometric transformations, 3D translation, 3D rotation (coordinate axis rotation, General 3-d rotation, Quaternion methods for 3D rotation), 3D scaling, 3D composite transformations, transformations between 3D coordinate systems.
UNIT IV **12 Hours**	Three-dimensional viewing : Overview of 3D viewing concepts, 3D viewing pipeline, 3D viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformations, orthogonal projections (axonometric and isometric, orthogonal projection coordinates, clipping window and orthogonal projection view volume, Normalization transformation), Oblique parallel projections (Cavalier and cabinet projections, Clipping window and Oblique parallel-projection view volume, Oblique parallel projection transformation matrix, normalization transformation)
UNIT X	Perspective projections (transformation coordinates, perspective-projection equations, vanshing points, view volume, transformation matrix, symmetric and oblique perspective-projection frustum, Normalized perspective-projection transformation coordinates), 3D clipping algorithms (region codes, point and line clipping, polygon clipping)..
Second series internal examination including theory + Laboratory if any	
Text Book : 1. Hearn and Baker, Computer Graphics with OpenGL, 3rdedn, Pearson. Reference Books: 1. Hill Jr. and Kelly, Computer Graphics using OpenGL, 3rdEdn, Pearson 2. Shreiner, Sellers, Kessenich, Licea-Kane, OpenGL programming guide, 8thedn, Pearson. 3. Foley, Van Dam,	

Feiner, Hughes, Computer Graphics- Principles and practice, Second Edition in C, Pearson Education

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU05DSEMCA306 DATA AND BUSINESS ANALYTICS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSC	300	KU05DSEMCA306	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2(T)+3(P)*

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: The course provides an understanding of principles involved in Data and Business Analytics at conceptual level. It describes application areas for studying and implementing Business intelligence. It depicts necessary developing skills to design business analytics and Intelligence projects using data mining and data warehousing concepts.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the basics of Data Business Analytics

CO2	Understand the Business intelligence framework and responsibilities
CO3	Illustrate the data mining concepts for Business Intelligence with suitable examples
CO4	Apply classification and prediction concepts to various applications

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours**	INTRODUCTION TO DATA BUSINESS ANALYTICS & INTELLIGENCE: Business Intelligence Definitions & Concepts, Framework, Data Warehousing concepts and its role in Business intelligence (BI), BI Infrastructure Components – BI Process, BI Technology, BI Roles & Responsibilities, Business, Applications of BI, BI best practices
UNIT 2 **10 Hours**	BASICS OF DATA INTEGRATION: Concepts of data integration, needs and advantages of using data integration, introduction to common data integration approaches, Meta data - types and sources, Introduction to data quality, data profiling concepts and applications, introduction to ETL using Pentaho data Integration (formerly Kettle)
First series internal examination including theory + Laboratory if any	
UNIT 3 **20 Hours**	INTRODUCTION TO MULTI-DIMENSIONAL DATA MODELING: Data Modelling Introduction to data and dimension modelling, multidimensional data model, ER Modelling vs. multi-dimensional modelling, concepts of dimensions, facts, cubes, attribute, hierarchies, star and snowflake schema, Introduction to business metrics and KPIs, Creating cubes using Microsoft Excel
UNIT IV **20 Hours**	DATA MINING FUNCTIONALITIES**:** Association rules mining, Mining Association rules from single level, multilevel transaction databases, Classification and prediction, Decision tree induction, Bayesian classification, k-nearest neighbour classification, Cluster analysis, Types of data in clustering, categorization of clustering methods

UNIT X	HR & SUPPLY CHAIN ANALYTICS: Human Resources – Planning and Recruitment – Training and Development - Supply chain network - Planning Demand, Inventory and Supply – Logistics – Analytics applications in HR & Supply Chain, Customer Behaviour, Analytics applications in Marketing and Sales (5 hours)
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. D R N Prasad, Seema Acharya: Fundamentals of Business Analytics, Wiley India, Second Edition, 2016.
2. David Loshin: Business Intelligence: The Savvy Manager's Guide., Latest Edition By Knowledge Enterprise. 2012

Core Suggested Readings

3. J.Han and M. Kamber: Data Mining: Concepts and Techniques By Morgan Kaufman publishers, Harcourt India pvt. Ltd. Latest Edition
4. Larissa Terpeluk Moss, ShakuAtre: Business Intelligence roadmap by Addison Weseley
5. CindiHowson: Successful Business Intelligence: Secrets to making Killer BI Applications by Tata McGraw Hill
6. Mike Biere: Business intelligence for the enterprise by Addison Weseley, August 2010.
7. Efraim Turban, Ramesh Sharda, Jay Aronson, David King, Decision Support and Business Intelligence Systems, 9th Edition, Pearson Education, 2009.
8. S. Christian Albright, Wayne L. Winston, Business Analytics: Data Analysis & Decision Making, 6th Edition, CENGAGE INDIA , 2017

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU05DSEMCA307 WEARABLE COMPUTING AND SENSORS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSE	300	KU05DSEMCA307	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2(T)+3(P)*

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course provides a comprehensive understanding of wearable computing technology and Body Sensor Networks, as well as the basic principles of different sensors and actuators.

Course Objectives:

- To explore Wearable components and building blocks of Wearable Computing.
- To enumerate the details of Body Sensor Networks (BSN).
- To create a conceptual understanding of the basic principles of sensors, actuators, and their operations.
- To understand various types of sensors, actuators, transducers and its application.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Learn about software, hardware tools, protocols and components required for Wearable Computing.
CO2	Understand the basics of Body Sensor Networks (BSN) and its Programming Framework.

CO3	Classify different Sensors & Actuators based on various physical phenomena and differentiate their performance characteristics.
CO4	Understand the working principle of special-purpose sensors and the need for developing smart sensors.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **10 Hours**	**Introduction to Wearable Components**: History - Internet of Things and Wearables - Wearables' Mass Market Enablers - Human Computer Interface and Human Computer Relationship - A Multi Device World, Role of Wearables, Attributes of Wearables
UNIT 2 **15 Hours**	**Building Blocks for Wearable Computing**: Bluetooth Low Energy (BLE) - Embedded Software Programming - Sensors for Wearables - Android Wear: Notification Settings and Control, Wear Network - Android Wear API: DataItem – DataMapItem – DataMap - Google Fit API: main package - data sub package. **Body Sensor Networks**: Typical m-Health System Architecture - Hardware Architecture of a Sensor Node - Communication Medium - Power Consumption Considerations - Communication Standards - Network Topologies - Commercial Sensor Node Platforms - Bio-physiological Signals and Sensors - BSN Application Domains - Developing BSN Applications - Programming Abstractions - Requirements for BSN Frameworks - BSN Programming Frameworks (15 hours) **Signal Processing In-Node Environment**: Introduction,Background,Motivations and Challenges, The SPINE Framework, Architecture, Programming Perspective-Optional SPINE Modules, High-Level Data Processing
First series internal examination including theory + Laboratory if any	

UNIT 3 **15 Hours**	**Overview of Sensors and Actuators**: Definitions: Sensors & Actuators – Overview of Sensor and Actuator classifications – Performance characteristics of Sensors & Actuators. **Smart Sensors and Applications:** Integrated and Smart sensors, IEEE 1451 standard & Transducer Electronic Datasheets (TEDs), Overview of various smart sensors: Digital temperature sensor (DS1621, TMP36GZ), Humidity sensor (DHT11, DHT22, FC28), IR sensor (FC51), Gas sensor (MQ2,MQ8), Pressure sensors (BMP180), Accelerometers (ADXL335), etc; Structural health monitoring sensors, Introduction to MEMSand Flexible sensors.
UNIT IV **20 Hours**	**Sensors and Transducers**: Thermoelectric sensors, piezoelectric sensors, pyroelectric sensors, photovoltaic sensors, electrochemical sensors, Wearable applications: temperature sensitive fabric, electrochemical sensors. Resistive sensors- Potentiometers, strain gages (piezo-resistive effect), resistive temperature detectors (RTD), thermistors, magnetoresistors, light dependent resistor (LDR), resistive hygrometers, resistive gas sensors. Wearable applications: Strain sensor for monitoring Physiological signals, body movement. **Mechanical Transducers**: Accelerometers: Characteristics and working principle, Types- Capacitive, Piezoresistive, piezoelectric; Gyroscopes: Characteristics and working principle, Rotor Gyroscope; Diaphragm Pressure Sensor –resistive & capacitive type (micro press sensor). Wearable applications: Motion sensors for fall detection, hemiplegic and PD (Parkinson's disease) patients.
UNIT X	Autonomic BSN, Agent-Oriented BSN , Collaborative BSN, Interation of wearable and cloud computing.

Second series internal examination including theory + Laboratory if any

Core Compulsory Readings

1. Sanjay M. Mishra, Wearable Android™: Android wear & Google Fit app development, 2015, 1st edition, John Wiley & Sons, USA
2. Fortino, Giancarlo, Raffaele Gravina, and Stefano Galzarano, Wearable computing: from modelling to implementation of wearable systems based on body sensor networks, 2018, 1st edition, John Wiley & Sons, USA
3. Jacob Fraden, "Handbook of Modern Sensors Physics, Designs, and Applications", 2016, 5th Edition, Springer, Switzerland.
4. Edward Sazonov, Michael R Neuman, "Wearable Sensors: Fundamentals, Implementation and Applications" Elsevier, 2014

Core Suggested Readings

1. Barfield, Woodrow, ed. Fundamentals of wearable computers and augmented reality, 2015, 1st edition, CRC press, USA
2. Nathan Ida, "Sensors, Actuators and their Interfaces - A Multidisciplinary Introduction", 2020, 2nd Edition, IET, United Kingdom.
3. Subhas Chandra Mukhopadhyay, Octavian Adrian Postolache, Krishanthi P. Jayasundera, Akshya K. Swain, "Sensors for Everyday Life Environmental and Food Engineering", 2017, Volume 23, Springer, Switzerland.
4. Volker Ziemann, "A Hands-On Course in Sensors Using the Arduino and Raspberry Pi", 2018, 1st Edition, CRC Press, United States.

5. Jon. S. Wilson, “Sensor Technology Hand Book”, Elsevier Inc., 2005.
6. Subhas C. Mukhopadhyay, “Wearable Electronics Sensors-For Safe and Healthy Living”, Springer International Publishing, 2015.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU05DSEMCA308 HIGH PERFORMANCE COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSE	300	**KU05DSEMCA308**	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: The course on High-Performance Computing (HPC) provides a comprehensive understanding of the principles, architectures, and techniques involved in executing computationally intensive tasks. Students will learn about parallel computing, distributed systems, and the utilization of HPC clusters for scientific simulations and data analysis. The course covers topics such as programming models, optimization strategies, and performance evaluation to maximize the efficiency of HPC applications.

Course Objectives:

- Understand the principles and foundations of high-performance computing and its importance in scientific and computational-driven fields.
- Gain knowledge of parallel computing architectures and techniques used in high-performance computing.
- Develop skills in programming and optimizing algorithms for efficient execution on HPC clusters.
- Learn to utilize advanced tools and frameworks for managing and analyzing large-scale data in high-performance computing environments.
- Apply the acquired knowledge and skills to solve complex computational problems and conduct scientific simulations using high-performance computing resources.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	The learner will be able to design, formulate, solve and implement high performance versions of standard single threaded algorithms
CO2	The learner will know and will be able to demonstrate the architectural features in the GPU and MIC hardware accelerators.
CO3	The learner will be able to design programs to extract maximum performance in a multicore, shared memory execution environment processor.
CO4	The learner will be able to design and deploy large scale parallel programs on tightly coupled parallel systems using the message passing paradigm.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	Levels of parallelism (instruction, transaction, task, thread, memory, function)- Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc)- Architectures: N-wide superscalar architectures, multi-core, multi-threaded.
UNIT 2 **15 Hours**	Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee Microarchitecture and Intel Nehalem microarchitecture- Memory hierarchy and transaction specific memory design- Thread Organization.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	Synchronization- Scheduling- Job Allocation-Job Partitioning- Dependency Analysis- Mapping Parallel Algorithms onto Parallel Architectures- Performance Analysis of Parallel Algorithms.
UNIT IV **15 Hours**	Bandwidth Limitations- Latency Limitations- Latency Hiding/Tolerating Techniques and their limitations- Power-aware Processing Techniques-Power-aware Memory Design- Power-aware Interconnect Design-Software Power Management. Petascale Computing-Optics in Parallel Computing- Quantum Computers- Recent developments in Nanotechnology and its impact on HPC.
UNIT X	Exascale Computing, Heterogeneous Computing, Parallel I/O and Storage Systems, Distributed Shared Memory (DSM), Extreme-Scale Data Analytics, Deep Learning and HPC, High-Performance Computing in Edge and IoT
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. George S. Almasi and AlanGottlieb, Highly Parallel Computing, Benjamin Cumming Publishers.
2. Kai Hwang ,Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGraw Hill 1993

Core Suggested Readings

1. David Culler, Jaswinder Pal Singh, Anoop Gupta, Parallel Computer Architecture: A hardware/Software Approach, Morgan Kaufmann, 1999.
2. K. Hwang& Z. Xu, Scalable Parallel Computing – Technology, Architecture, Programming., McGraw Hill 1998.
3. William James Dally and BrianTowles, Principles and Practices on Interconnection Networks, Morgan Kauffman 2004.
4. Hubert Nguyen , GPU Gems 3, Addison Wesley, 2008, (Chapter 29 to Chapter 41)
5. AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing, , 2nd edition, Pearson, 2003.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU05DSEMCA309 GAME DEVELOPMENT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	DSE	300	KU05DSEMCA309	4	75

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
3	2	1	50	50	100	2(T)+3(P)*

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: Embark on a journey into the dynamic world of game development with our comprehensive course. Designed for both beginners and enthusiasts alike, this course offers a holistic understanding of the game development process. From foundational concepts in design and programming to advanced techniques in graphics, networking, and AI, students will gain hands-on experience using industry-standard tools such as Unity. Through a blend of theory, practical exercises, and project-based learning, participants will emerge equipped with the skills and knowledge necessary to bring their game ideas to life across various platforms. Whether aspiring to join the thriving game industry or simply seeking to unleash their creativity, this course provides a solid foundation for all game development enthusiasts.

Course Objectives:

- Provide a comprehensive understanding of game development fundamentals, tools, and

trends.

- Develop proficiency in game design, prototyping, and development using Unity and other industry-standard tools.
- Explore advanced concepts like scripting, AI, multiplayer networking, and optimization techniques.
- Foster hands-on, project-based learning to apply acquired skills and build a strong portfolio for real-world scenarios.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Gain a solid grasp of game development fundamentals, tools, and trends.
CO2	Become proficient in game design, prototyping, and development using industry-standard tools like Unity.
CO3	Master advanced concepts including scripting, AI, multiplayer networking, and optimization techniques
CO4	Build a strong portfolio, preparing them for real-world scenarios and enhancing employability in the game industry.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	**Introduction to Game Development** Overview of game development: history, trends, and current landscape, Basics of game design principles, Introduction to game engines and their importance, understanding game assets: graphics, sound, and music, Basics of programming languages commonly used in game development (e.g., C#, C++, Java)
UNIT 2 **15 Hours**	**Game Design and Prototyping** Deep dive into game design: mechanics, dynamics, and aesthetics, Prototyping techniques and tools, User experience (UX) and user interface (UI) design for games, Playtesting and iteration, Introduction to version control systems for collaborative development (e.g., Git), Assignment: Create a prototype of the game designed in Module 1 and conduct playtesting sessions for feedback.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	**Game Development with Unity** Introduction to Unity game engine, Understanding Unity's interface and workflow, Scene creation and management, Scripting in C# for game logic, Implementing player controls and interactions, Introduction to 2D and 3D game development in Unity
UNIT IV **15 Hours**	**Advanced Game Development Concepts** Advanced scripting and programming techniques in Unity, Working with physics and animations, Implementing artificial intelligence (AI) for non-player characters (NPCs), Multiplayer concepts and networking, Optimizing games for performance and scalability, Publishing games on various platforms (PC, mobile, console)
UNIT X	Advanced graphics programming: shaders, lighting, and post-processing effects, Procedural content generation (PCG) techniques for creating dynamic and varied game worlds, Introduction to virtual reality (VR) and augmented reality (AR) game development, Advanced networking and multiplayer synchronization techniques, Integrating machine learning for AI behavior and procedural generation, Performance optimization strategies for resource-intensive games
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Book: "Game Design Workshop: A Playcentric Approach to Creating Innovative Games" by Tracy Fullerton
2. Book: "The Art of Game Design: A Book of Lenses" by Jesse Schell

Core Suggested Readings

1. Book: "Unity in Action" by Joe Hocking
2. Book: "Game Programming Patterns" by Robert Nystrom
3. Book: "Real-Time Rendering" by Tomas Akenine-Möller, Eric Haines, Naty Hoffman
4. Book: "Game AI Pro" edited by Steve Rabin

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

SKILL ENHANCEMENT COURSES (SEC)

Semester 4 SEC POOL 1

KU04SECMCA201 PYTHON PROGRAMMING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	SEC	200	KU04SECMCA201	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course mainly focuses on introducing the fundamental programming concept to the students from other disciplines. This course mainly focuses on the basic concept of programming constructs such as language elements, data types, operators control structures and data structure that will be useful for the programmers to learn the basic programming concept. Structure of this course is well organized to introduce the programing basic concept to advanced concepts such as modules, packages, GUI, and basics of NLP concepts for pattern matching and searching. After completing

this course, the students acquire the ability to develop real life applications commonly useful for society in many walks of life.

Course Objectives:

- Aims to impart basic programming skills to the learners in a simplest way.
- Impart knowledge on fundamental and advanced data structure concepts.
- Acquire the knowledge to impart various control structures to implement programming logic.
- Aware about the development of common GUI based applications in simple steps.
- Acquire the ability to analysis of data using NumPy and Pandas

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Familiarize the different parts of Python programming and its applications in real world problems
CO2	Understanding the concept of different control structures and functions in Python.
CO3	Aware about the various data types and it operations and supporting methods
CO4	Make aware about the advanced concepts in Python such as reading and writing of data files, NumPy, Pandas and GUIs.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	**Introduction**: History of Python Programming, Thrust Areas Of Python, Installing Anaconda Python Distribution, PyCharm IDE and Jupyter Notebook, Creating And Running First Python Project, **Parts of Python Programming Language**-identifiers, keywords, statements and expressions, variables, operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversion, The *typedef()* function and Is operator
UNIT 2 **15 Hours**	**Control Flow Statement**- Decision control flow statement *(if, if ...else, if...elif…*, nested *if*), Loop (*while, for*),*continue*, *break* statements, Catching Exception Using *try* and *except* Statement **Functions**- Built-In Functions Commonly used Modules, Function definition and calling the function, The *return* statement and *void* function, scope and life time of variables.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	**Lists-** Creating List, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions used on lists, list Methods. **Dictionaries**- Creating Dictionary, Accessing and Modifying *key:value* Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods. **Tuples and Sets-** Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions used on Tuples, Relation between Tuples and Lists, Sets and Set Methods, Frozenset. **Strings-** Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings
UNIT IV **15 Hours**	**Files**- Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary files, Reading and Writing CSV Files, Introduction to NumPy and Pandas. GUIs in Python: Root Window-Fonts and colors- Working with containers and canvas, Frames, Widgets, Button widgets, Arranging widgets in the Frame, Label Widget, Message Widget, Text widget, Scrollbar widgets, Check button widget, Radio button widget, Entry Widget, Spin box Widget, List Box Widget, Menu Widget
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Gowrishankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372
2. Alberto Fernandez Villan, Mastering OpenCV 4 with Python, Packt Publishing Ltd.
3. Dr. R Nageswara Rao, Core Python Programming, 2nd edition, Dreamtech Publisher, 2019

Core Suggested Readings

1. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st Edition, O'Reilly Media, 2017. ISBN – 13: 978-1491962299.
2. Wesley J. Chun, Core Python Programming, Second Edition, Publisher: Prentice Hall Pub

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print.

ASSESSMENT RUBRICS

Refer to section 7.

KU04SECMCA202 FUNDAMENTALS OF DIGITAL SKILLING USING GOOGLE WORKSPACE FOR EDUCATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
4	SEC	200	KU04SECMCA202	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course is intended to provide a fundamental understanding about the services offered by the Google Workspace for Education. Learners will get an exposure to how to use the services such as Google Drive, Google Docs, Google Sheet and Google Slides for their personal and collaborative learning strategies.

Course Objectives:

- To familiarize with about the Google Workspace for Education
- To understand the fundamentals of Gmail
- To get familiar with Google Drive
- To create documents using Google Doc and Google Sheet
- To create slides using Google Slides

Course Outcome

At the end of the Course, the Student will be able to:

SL	Course Outcomes
CO1	Identify the services offered by Google Workspace for Education and get exposed to the fundamentals of Gmail
CO2	Illustrate the foundations of preparing documents using Google Docs
CO3	Prepare documents using Google Sheets
CO4	Prepare slides using Google Slides

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	Introduction to Google Workplace for Education: Gmail, Google Calendar, Google Drive, Google Docs, Google Sheets, Google Slides. Gmail - Creation of ID - Composing Messages - Folders - Labels - Simple Customizations.
UNIT 2 **15 Hours**	Google Drive: Uses - Accessing - Viewing Drive Contents - Creating Folders - Creating Documents - Making Copies - Uploading Documents - Downloading Folder Contents - Sharing Folders - Erasing Documents - Creating and Managing Workspaces - Searching Contents - Managing Storage.
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	Google Doc: Creation - Saving - Typesetting and Formatting - Images - Tables - Charts - Line - Emoji - Smart Chips - Dropdown - Footnote - Header and Footer - Page Numbers - Indentation - Lists - Checklists - Creating Links - Watermark - Page Orientation - Spelling and Grammar Check - Citations - Modes (View / Review / Edit) - Voice Typing - Print - Download - Sharing.

UNIT IV **15 Hours**	Google Sheet: Creation - Components - Cells and Addressing - Navigating. Entering and Editing Data: Text - Equations - Typesetting and Formatting. Saving Worksheet. Editing Worksheet Data - Moving and copying data - Moving and copying equations - Inserting and deleting ranges, rows, and columns. Formatting: Text - Numbers - Row and Column -- Conditional Format - Borders and Colors. Worksheets: Naming, Copying, Creating, Removing Charts: Types - Pie Chart - Bar Chart. Data - Sorting - Filters - Validation - Removal of Duplicates. - Page Orientation - Spelling and Grammar Check - Print - Download - Sharing.
UNIT X	**Module X (Teacher Specific)**: Google Slide: - Creating Slides - Adding Image, Text Box, Audio, Video, Shape, Chart, Diagram. Typesetting and Formatting - Lists - Checkboxes. Slides: Ordering - Changing Background - Deleting - Transition. Share - Print – Download **Core Compulsory Readings**
Second series internal examination including theory + Laboratory if any	
1. Learn the basics, https://workspace.google.com/intl/en_in/training/ **Core Suggested Readings** 1.Online materials related to Google Workplace for Education. 2 Digital Skilling Course (NPTEL), https://drive.google.com/file/d/1eKZP5LchZqrf3Ba9MjjeMDUh7cjbDOR9/view?usp=sharing	

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Refer to section 7.

KU04SECMCA203 WEB TECHNOLOGY

Semester	Course Type	Course Level	Course Code		Credits	Total Hours
4	SEC	300	KU04SECMCA203		3	60

Learning Approach (Hours/ Week)			Marks Distribution				Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE		Total	
2	2	0	50	50		100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: This course provides a comprehensive introduction to web technologies, covering both frontend and backend development concepts. Students will gain practical skills in building dynamic and responsive web applications while learning about the fundamental principles of web design, development, and deployment.

Course Objectives:

- Understand the structure and functionality of the internet and the World Wide Web.
- Differentiate between client-side and server-side technologies and understand their respective roles in web development.
- Gain an introduction to server-side programming languages such as PHP, Python, or Node.js.
- Gain an overview of web hosting options and deployment strategies for deploying web applications.
- Build a simple web project from scratch using frontend and backend technologies learned in previous modules.

Course Outcomes:

SL #	Course Outcomes
CO1	Create basic web pages using HTML and CSS.
CO2	Demonstrate proficiency in JavaScript programming

CO3	Understand the concepts of database management systems (DBMS) and SQL for managing data in web applications.
CO4	Students will be able to build a simple web project

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT 1 **15 Hours**	Introduction to Web Technologies: Overview of the Internet and the World Wide Web. Client-side vs. server-side technologies. Basic concepts of HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets), Introduction to web browsers and their developer tools, Building and viewing a simple webpage using HTML and CSS.
UNIT 2 **15 Hours**	Advanced HTML5 features: Semantic elements, forms, and multimedia, Advanced CSS3 techniques: Flexbox, Grid layout, transitions, and animations, Introduction to responsive web design principles and techniques. Basics of JavaScript programming: Variables, data types, operators, and control structures. DOM manipulation using JavaScript: Selecting elements, modifying content, and handling events. Introduction to popular frontend frameworks/libraries such as Bootstrap and jQuery
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	Introduction to server-side programming languages: PHP, Python, or Node.js. Basics of database management systems (DBMS) and SQL (Structured Query Language). Building dynamic web applications with server-side scripting. Handling form data and user input in web applications. Introduction to RESTful APIs and AJAX for asynchronous communication
UNIT IV **15 Hours**	Basics of web performance optimization and debugging techniques. Introduction to web security best practices: HTTPS, CORS, XSS, CSRF. Overview of web hosting options and deployment strategies. Building a simple web project from scratch using frontend and backend

	technologies learned in previous modules.
UNIT X	Responsive Web Design Frameworks, JavaScript Frameworks, Advanced Node.js Features, Web Server Frameworks, Advanced DOM Manipulation. Content Managements Systems (CMS) (5 Hours)
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Jeffrey C. Jackson, Web Technologies: A Computer Science Perspective, Prentice Hall
2. David Flanagan, JavaScript: The Definitive Guide, 6th Edn. O'Reilly Media.2011
3. Steven Holzner, PHP: The Complete Reference, McGraw Hill Professional, 2008
4. Steve Suehring, Tim Converse, Joyce Park, PHP6 and MY SQL Bible, John Wiley & Sons, 2009
5. Anthony T. HoldenerIII, Ajax: The Definitive Guide, O'Reilly Media, 2008

Core Suggested Readings

1. Bob Breedlove, et al, Web Programming Unleashed, Sams Net Publishing, 1stEdn
2. Pedro Teixeira, Instant Node.js Starter, Packt Publishing Ltd., 2013
3. James Snell, Programming Web Services with SOAP, O'Reilly 2002 10. Jacob Lett, Bootstrap Reference Guide, Bootstrap Creative 2018
4. Maximilian Schwarzmüller, Progressive Web Apps (PWA) - The Complete Guide, Packt Publishing 2018
5. Mahesh Panhale, Beginning Hybrid Mobile Application Development, Apress 2016

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Lab session

MODE OF TRANSACTION

- Lecture and Lab
- , Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Refer to section 7.

Semester 5 SEC POOL 2

KU05SECMCA301 THE ART OF E - DOCUMENTATION USING LATEX

Semester	Course Type	Course	Course Code	Credits	Total Hours

		Level			
5	SEC	300	KU05SECMCA301	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description:

This course is intended to impart the foundations of document preparation using Latex. Latex is a simple but flexible and powerful document editor that can be used to create documents of varying purposes. It can also be used to create slides for presentations.

Course Objectives:

- To impart knowledge about the structure of Latex documents
- To become familiar with using the various options to type set and format contents in Latex
- To create documents with tables, list and images in Latex
- To add citations and references for a scientific document

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Identify the anatomy of a Latex document and prepare simple documents in Latex
CO2	Illustrate the foundations of typesetting and formatting in Latex
CO3	Prepare documents with lists, tables and images and to create references and citations
CO4	Create slides using beamer

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	WYSIWYG Editors Vs Latex. Latex: History - Advantages - Editors (Windows / Mac / Ubuntu - Online). Anatomy of Latex Document - Common Document Classes - Paper sizes - Commonly Used Packages - Setting margins. Document Creation - Beginning Document - Sections - Adding Text - Inserting blank lines - Ending Document
UNIT 2 **15 Hours**	Typesetting Document - Fonts, symbols, indenting, paragraphs, line spacing, word spacing, titles and subtitles, adding colors to text and entire page, adding bullets and numbered items. Adding header and footer, changing the page orientation, dividing the document into multiple columns
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	Creating Tables: Setting Columns - Merging Rows and Columns - Various Styles and Orientation. Adding Images - Graphics Packages - Rotation - Scaling. Drawing Matrices - Adding Mathematical Symbols and Equations. Adding Bibliography, Cross References and Citations - Adding Table of Contents, Figures and Tables
UNIT IV **15 Hours**	Beamer - Creating Slides - Adding frames - Dividing the slide into multiple columns - Adding different blocks - Adding Tables and Graphics. Case Study: Preparation of Resume, Official Letters, Book, Article, Homework assignment
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

Free online introduction to LaTeX Available at https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)

Learning Latex, D.F.Griffits, D.J.Higham, Siam, Philadelpha, 1997

Latex, https://en.wikibooks.org/wiki/LaTeX

Core Suggested Readings

1.Online materials related to document preparation using Latex and slide preparation using Beamer

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration.

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Refer to section 7.

KU05SECMCA302 DATA PROCESSING WITH PYTHON

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	SEC	300	KU05SECMCA302	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: This course includes an overview of the various tools available for writing codes and running the same in Python, and gets students coding quickly. It also provides hands-on coding exercises using commonly used data structures and writing custom functions after preprocessing the dataset.

Course Objectives:

- Understand how to perform some basic tasks to start exploring and analyzing the imported data set
- Learn how to perform computations on the data to calculate basic descriptive statistical information
- Learn how to manipulate data efficiently.
- Understand the visualization and analytics of data.

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	To understand python fundamentals and data types used in python
CO2	To learn basic statistical measurements required for machine learning
CO3	To familiarize preprocessing of data before processing.
CO4	To experience visualization of 1D and 2D data

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT 1 15 Hours	Python Fundamentals: Basic Concepts, Naming Variables, Operators and operands, Expressions, importing libraries, Redirecting the output, Data types, Lists, Working with files, Working with Directories; Big Data vs Data Science
UNIT 2 15 Hours	Introduction to probability: discrete and continuous variables, probability distribution, Bayes Rule, Variance, Standard Deviation, Types of probability distributions, Algorithms designed using probability
First series internal examination including theory + Laboratory if any	
UNIT 3 15 Hours	Finding datasets, Jupiter notebooks and loading data, pandas vsnumpy, Saving, Dropping Null Values, User Defined Functions, Cleaning Dataset, Graphs and Statistics, Histogram, Working with rows and columns.
UNIT IV 15 Hours	Introduction to basic plots, pandas vs matplotlib, visualizing 1D distributions, visualizing 2D distributions, Higher dimension visualizations.
Second series internal examination including theory + Laboratory if any	
Core Compulsory Readings 1. Wes McKinney, “Python for Data Analysis”, 2nd Edition, O’Reilly	

2. Joel Grus," Data Science from Scratch" First Edition, April 2015
3. McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

TEACHING LEARNING STRATEGIES

Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Print.

ASSESSMENT RUBRICS

Refer to section 7.

KU05SECMCA303 - DATA SCIENCE FUNDAMENTALS

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
5	SEC	300	KU05SECMCA303	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: A Data Science course covers basic and advanced concepts of data analytics, machine learning, statistics, and programming languages like Python or R. It also teaches students how to clean a dataset, and interpret large datasets and identify patterns to create predictive models.

Course Objectives:

- To provide a strong foundation for data science and application area related to it and understand the underlying core concepts and emerging technologies in data science.
- Demonstrate an understanding of statistics and machine learning concepts that are vital for data science
- Explain how data is collected, managed and stored for data science
- To understand how to perform data analysis and visualization tasks.

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	To learn basic ideas about data science fundamentals, machine learning and statistical probabilities.
CO2	To understand different types of data such as structured and unstructured data.
CO3	To familiarize different classification algorithms.
CO4	To experiment different algorithms using python and calculate the accuracy.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENTS

UNIT 1 15 Hours	Introduction to Data Science: Definition, Big data, populations and samples, exploratory data analysis, properties of data, Mathematical and Statistical Skills, Data Analysis Tools, Data Science Applications
UNIT 2 15 Hours	Types of data, Application areas of Data Science, Data Science process, Machine Learning: Introduction, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Introduction to Statistics.
First series internal examination including theory + Laboratory if any	
UNIT 3 15 Hours	Basic Machine Learning Algorithms: Linear Regression, SVM, Naïve Bayes, example program implementation, Data acquisition, Transformation into standardized format, Data Cleaning, Data Reduction, Data Integration, Data Transformation, Normalization.
UNIT IV 15 Hours	Algorithm design and analysis, Data, Database Table, Python, Confusion Matrix, Data Handling and Visualization, Different Chart types, Plotting Functions, Solving problem with Data Science.
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Rachel Schutt&O'neil, "Doing Data Science", Straight Talk from The Frontline O'REILLY, ISBN:978-1-449-35865-5, 1st edition, October 2013.
2. Joel Grus," Data Science from Scratch" First Edition, April 2015

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Refer to section 7.

KU06SECMCA308 ARTIFICIAL INTELLIGENCE IN DAILY LIFE

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	SEC	300	KU06SECMCA308	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: The objective of the course is to communicate the fundamental knowledge in the specialized area of Artificial Intelligence. The program focuses on building a comprehensive understanding on the basics of Artificial Intelligence and interact with inter- disciplinary groups.

Course Objectives:

- To understand the history of artificial intelligence (AI) and its foundations
- To understand the basic principles of AI for solving daily life problems

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Acquire knowledge about Artificial Intelligence
CO2	Obtain knowledge about Search and Control Strategies
CO3	Attain information about the fundamentals of Machine Learning
CO4	Demonstrate awareness and a fundamental understanding of various applications of AI techniques in intelligent agents, expert systems, Natural language Processing- machine learning models.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 15 Hours	Introduction to Artificial Intelligence- History of AI- Advantages and Disadvantages of AI- Applications- AI domains.AI in real life, Expert system - Expert system development- Modern expert systems
UNIT 2 15 Hours	Search and Control Strategies- State- Space representation- Problem Solving - Heuristic Techniques – Hill Climbing – Simulated Annealing
First series internal examination including theory + Laboratory if any	
UNIT 3 15 Hours	Machine Learning- Supervised and Unsupervised Algorithms- Neural Networks- Classification and Predictions model – Applications
UNIT IV 15 Hours	Natural Language Processing - Natural Language Processing Tasks -NLP Applications- Recommender System - Sentimental Analysis

Second series internal examination including theory + Laboratory if any
Core Compulsory Readings
1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach Third Edition Pearson Education 2010 Inc. ISBN: 978-0-13-604259-5.
2. D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990
Core Suggested Readings
1. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill.
2.https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.html

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Lab session

MODE OF TRANSACTION

- Lecture and Lab
- , Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Refer to section 7 of FYIMP- Computational Science - Scheme and Syllabus for the 3 credit courses with 2 Credit Theory + 1 Credit Practical.

KU06SECMCA309 FUNDAMENTALS OF BIG DATA

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	SEC	300	KU06SECMCA309	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Course Description: This course mainly focuses to introduce the fundamentals of big data analytics concept using Python programming language. First part of this course familiarizes basic

concepts of the python programming language and subsequently introduces the big data analytics concepts and machine learning algorithms for developing mathematical/statistical models for analyzing big data using NumPy and Pandas in Python. After completing this course, the students acquire the basic ability to develop real life applications for extracting hidden patterns/trends in a big data environment that is commonly useful for society in many walks of life.

Course Objectives:

- Aims to impart basic programming skills to the learners for developing mathematical models for big data analysis.
- Impart knowledge on fundamental and advanced data structure concepts in Python programming language suitable for big data analysis.
- Aim to impart to knowledge in bigdata platforms and developing mathematical/statistical models for big data analysis using machine learning concepts.
- Acquire the ability to analysis of data using NumPy and Pandas

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Familiarize the different parts of Python programming and its applications in real world problems
CO2	Understanding the concept of different control structures and functions in Python.
CO3	Aware about the different types of data files and an overview of various data analytics methods.
CO4	Acquire the knowledge of different machine learning algorithms and its implementation for building efficient data analytics models for real life.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔

CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 **15 Hours**	History of Python Programming, Thrust Areas Of Python, Installing Anaconda Python Distribution, PyCharm IDE and Jupyter Notebook, Creating And Running First Python Project, **Parts of Python Programming Language**-identifiers, keywords, statements and expressions, variables, operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversion, The *typedef()* function and Is operator
UNIT 2 **15 Hours**	**Control Flow Statement**- Decision control flow statement *(if, if ...else, if...elif*…, nested *if*), Loop (*while, for*),*continue*, *break* statements, Catching Exception Using *try* and *except* Statement **Functions-** Built-In Functions, commonly used Modules, Function definition and calling the function, The *return* statement and *void* function, scope and life time of variables
First series internal examination including theory + Laboratory if any	
UNIT 3 **15 Hours**	**Files: -** Types of Files: -Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary files, Reading and Writing CSV Files, Introduction to NumPy and Pandas. **Data Analytics Overview** - Statistical Computing - Mathematical Computing Using NumPy - Data Processing with Pandas - Data Visualization with Python - Introduction to Model Building for Evaluation
UNIT IV **15 Hours**	Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms - K- Means. Association Rule Mining - Apriori algorithm.
Second series internal examination including theory + Laboratory if any	
Core Compulsory Readings 1. Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013. 2. David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015. 3.Dr. R Nageswara Rao, Core Python Programming, 2nd edition, Dreamtech Publisher, 2019	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06SECMCA310 OPTIMIZATION TECHNIQUES

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	SEC	300	KU06SECMCA310	3	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
2	2	0	50	50	100	2

*** ESE Duration: 2 hours for theory and 3 hours for Lab**

Optimization, also known as mathematical programming, collection of mathematical principles and methods used for solving quantitative problems in various disciplines, including physics, biology, engineering, economics, and business. The subject grew from a insight that quantitative problems in obviously different disciplines have important mathematical elements in common.

Course Objectives:

- To understand the need of optimization
- To study various optimization techniques

Course Outcome

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the importance of optimization technique
CO2	Discuss Optimum design concepts
CO3	Solve the Linear Programming models using graphical and simplex methods

CO4	Evaluate different algorithmic methods for solving constrained and unconstrained optimization problems

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔

COURSE CONTENT

UNIT 1 15 Hours	Optimization: Introduction, Statement of an Optimization problem,formulation of Optimal Problem, Types of Optimization problem.
UNIT 2 15 Hours	Optimum design concepts: Definition of Global and Local optima,Optimality criteria, Convexity and concavity of functions of one and two variables,Lagrangian function, Hessian matrix formulation
First series internal examination including theory + Laboratory if any	
UNIT 3 15 Hours	Linear programming: Standard form of Linear Programming Problem,Canonical form, Elementary operations, Graphical method for two variable optimization problem, Simplex method, Karmarkar's projective scaling method.
UNIT IV 15 Hours	Optimization algorithms for solving unconstrained optimizationproblems – Gradient based method: Cauchy's steepest descent method, Newton'smethod, Conjugate gradient method. Optimization algorithms for solving constrained optimization problems– direct methods – penalty function methods – steepest descent method.
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. G. Hadley, Linear programming, Narosa Publishing House, New Delhi, ISBN 13: 9788185015910.
2. Singiresu S. Rao, Engineering Optimization: Theory and Practice by John Wiley and Sons, (5th edition),ISBN: 978-1-119-55479-3

Core Suggested Readings

1. Shikare MM, Waphare BN, Combinatorial Optimization, Narosa Publication (2004)

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Refer to section 7.

S6 - List of Discipline Specific Electives (DSE-3)

S6 - List of Discipline Specific Electives (DSE-3)											
No	Level	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
					L	P	Tt	ESA	CE	T
6E.1	300	KU06DSEMCA314	Information Security	4	4	0	1	50	50	100
6E.2	300	KU06DSEMCA315	Data and Information visualization	4	4	0	1	50	50	100
6E.3	300	KU06DSEMCA316	Virtual and Augmented Reality	4	4	0	1	50	50	100
6E.4	300	KU06DSEMCA317	Game Development	4	4	0	1	50	50	100
6E.5	300	KU06DSEMCA318	Computer Vision	4	4	0	1	50	50	100
6E.6	300	KU06DSEMCA319	Computational Photography	4	4	0	1	50	50	100
6E.7	300	KU06DSEMCA320	High performance computing	4	4	0	1	50	50	100
6E.8	300	KU06DSEMCA321	Green Computing	4	4	0	1	50	50	100

KU06DSEMCA314 INFORMATION SECURITY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA314**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.This course provides with the basics concepts of cryptography including traditional ciphers, block ciphers, stream ciphers, private and public key cryptosystems. The course also includes key-exchange algorithms, hash functions, authentication systems like MAC and digital signature as well as network security protocols.

Course Objectives:

- Accomplish knowledge on basic concepts of information security
- Study different cryptographic techniques
- Obtain knowledge on Message Authentication and Digital Signatures
- Accomplish knowledge about Hash functions
- Learn various network security protocols

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	**Explain the basic concepts of information security**
CO2	**Illustrate different cryptographic techniques**
CO3	**Describe various concepts of Message Authentication**
CO4	**Explain Hash Functions and Basic network security Essentials**

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓

CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Foundations of Cryptography and security: Ciphers and secret messages, security attacks and services.Classical Encryption techniques - Symmetric cipher model, substitution techniques, transposition techniques, steganography. Basic Concepts in Number Theory and Finite Fields .
UNIT 2	Block cipher principles – The data encryption standard (DES) – strength of DES – Differential and linear cryptanalysis – Block cipher design principles. Advanced encryption standard – AES structure – AES transformation function – key expansion – implementation. Block cipher operations –Multiple encryption – ECB – CBC – CFM – OFM – Counter mode.
First series internal examination including theory + Laboratory if any	
UNIT 3	Public Key cryptography: Prime numbers and testing for primality, factoring large numbers, discrete logarithms. Principles of public-key crypto systems – RSA algorithm. Diffi-Helman Key exchange, Elgammal Cryptographic systems - Hash functions – examples – application – requirements and security – Hash function based on Cipher block chaining – Secure Hash algorithm.
UNIT IV	Message authentication requirements - Message authentication functions – requirements of message authentication codes - MAC security – HMAC – DAA – CCM – GCM. Digital signatures, Digital signature standard. Transport-Level Security ,Wireless Network Security, Electronic Mail Security, IP Security
UNIT X	Introduction on Security Essentials Malicious Software ,Types of Malicious Software; Intruders , Intrusion Detection;Firewalls ,The Need for Firewalls, Firewall Characteristics , Types of Firewalls; Legal and Ethical Issues, Cybercrime and Computer Crime ;Intellectual Property
Second series internal examination including theory + Laboratory if any	
Text book William Stallings, Cryptography and Network Security, Pearson 2004 References 1. Foorouzan and Mukhopadhyay, Cryptography and Network security, 2ndedn	

2. BuceSchneier., Applied cryptography – protocols and algorithms, SpringerVerlag 2003

3. William Stallings, Network Security Essentials, , 4thedn, Pearson

4. Pfleeger and Pfleeger, Security in Computing, 4thEdn, Pearson

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA315 DATA AND INFORMATION VISUALIZATION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA315**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.

This course focuses on building creative and technical skills to transform data into visual reports for the purpose of a better understanding. Students will learn to organize and visualize data with an emphasis on applying design principles to produce clear, elegant graphs from the data.

Course Objectives:

- Employ best practices in data visualization to develop charts, maps, tables, and other visual representations of data
- Implement visualizations in Python

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Identify appropriate visualization methods for a given data type
CO2	Describe information visualization methods such a cladogram, cartogram, heatmap and dendrogram
CO3	Illustrate various methods for information visualization
CO4	Explain various aspects of scientific visualization

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Data and types of data, Data variability, uncertainty and context. Basics of Data Visualization: Definition of Data Visualization, Need for Visualization, how to visualize data, General types of Data Visualization, pros and Cons of Data Visualization. Visualization Components: Visual cues, Coordinate systems,

	Scales, and Context. Diagrams used for data visualization: Bar chart, Histogram, Scatter plot, Scatter plot, Network, Streamgraph, Tree map, Gantt chart, Stripe graphic, Animated spiral graphic. Visualization basedon types of data: Visualizing Categorical data, Visualizing Time series data, Visualizing Spatial data
UNIT 2	Information Visualization: Definition, Objectives of Information Visualization. Visual representation of large-scale collection of non - numerical information. Design Principles of Information Visualization: Principle of Simplicity, Principle of Proximity, Principle of Similarity, Principle of Closure, Principle of Connectedness, Principle of Good Continuation, Principle of Common fate, Principle of Familiarity, Principle of Symmetry
First series internal examination including theory + Laboratory if any	
UNIT 3	Methods for Information Visualization: Cartogram, Cladogram (phylogeny), Concept Mapping, Dendrogram (classification). Graph drawing, Heat map, Hyperbolic Tree, Tree mapping Multidimensional scaling. Information visualization reference model
UNIT IV	Scientific visualization: Introduction, Methods for visualizing twodimensional and three-dimensional data sets, volume visualization. Data Visualization using in Python matplotlib Module, pyplot, plot(), scatter, bar charts, Formatting, figure(), subplot(), text(), xlabel(), ylabel(), title(), Plotting Mathematical Functions
UNIT X	**Case study with real world problems, techniques for creating interactive visualizations using tools like D3.js, Plotly, or Bokeh. Geospatial Visualization, Network Visualization, Text and Document Visualization, Temporal Visualization, Biomedical Visualization**
Second series internal examination including theory + Laboratory if any	
1. Nathan Yau, Data Points. Wiley Big Data Series 2. Healy, Kieran, Data Visualization: A Practical Introduction. Princeton University Press 3. Ben Bederson and Ben Shneiderman. The Craft of Information Visualization: Readings and Reflections. Morgan Kaufmann, 2003	

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA316 VIRTUAL AND AUGMENTED REALITY

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA314**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.This course offers a comprehensive exploration of the domains of virtual reality (VR) and augmented reality (AR). Comprising five detailed modules, the curriculum delves into the fundamental concepts, theoretical underpinnings, and integration of various sensory inputs. Moreover, it provides an exploration of foundational concepts and tools related to AR.

Course Objectives:

- **To give a comprehensive knowledge on the historical and new perspectives of VR and AR.**
- **To provide an in-depth comprehension of the foundational aspects and mathematical frameworks of VR and AR.**
- **To familiarize with the latest progressions and technological advancements within these fields.**

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Understand the fundamentals of VR, its historical context, hardware and software, and various application domains of VR and AR
CO2	Gain knowledge of the challenges inherent in VR, such as latency, hardware limitations, navigation issues, and health concerns.
CO3	Familiarize the geometric principles, the fundamentals of light and optics, and visual perception for designing and developing VR..
CO4	Develop an understanding of the audio perception in VR, and visual rendering which specifies what the visual display should show through an interface.
CO5	Know in detail about the fundamentals and applications of AR systems and technologies, along with motion, tracking, and interaction design techniques within VR environments.
CO6	Develop an understanding of the evaluation fundamentals to build better VR systems and experiences, advanced technologies for future VR and AR, and tools and platforms for AR.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Introduction to Virtual Reality: Introduction – Definition of VR, History of VR; VR Hardware and Software - sensors, displays (CAVE and HMDs), virtual world generator, game engines; Human physiology and perception; Presence and immersion in VR; VR Applications. Challenges in VR - latency, hardware challenges, navigation,

	motion sickness, other health issues. (20 hours)
UNIT 2	Geometry of Virtual Worlds: Geometric modelling; Transforming models; Matrix algebra and 2D rotations; 3D rotations and yaw, pitch, and roll; Axis-angle representations; Quaternions; Homogeneous transforms; Eye transforms; Canonical view transform; Viewport transform. Optics and Visual Perception in VR: Optics - Light propagation, refraction, lenses, optical aberrations, human eye, cameras and displays; Physiology of Vision - the human eye, photoreceptors and densities, eye movements, implications for VR; Visual perception - depth perception, motion perception, colour perception, frame rates and displays.(20 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	Audio in VR: Physics of sound and ear physiology; auditory perception; auditory localization; Rendering. Visual Rendering: Visual rendering; Ray tracing and shading; Rasterization; Pixel shading; VR rendering problems; anti-aliasing; Distortion shading; postrendering image warp. (20 hours)
UNIT IV	Motion in VR, Tracking, and Interaction: Motion - motion in real world, vestibular system, physics of motion in virtual world; Tracking - orientation tracking, tilt drift correction, yaw drift correction, tracking with a camera, perspective n-point problem, filtering, lighthouse approach; Interaction - overview of interfaces, locomotion, manipulation, social interaction. Augmented Reality: Introduction to AR; The Basics of AR Functionality; AR applications; Working of AR and Concepts of AR; AR hardware – sensors, processors, displays, and AR systems; AR software; AR content – visual content, C129 audio content, content for other senses (touch, taste, and smell); Interaction in AR – real world interaction, manipulation, navigation, communication, interaction in projected AR environments. (20 hours)
UNIT X	Evaluating VR Systems and Experiences: Perceptual training; Recommendations for developers and best practices; VR sickness; Experimental methods that involve human subjects. (10 hours) Advanced Technologies for VR: Touch and Haptics; Taste and smell; Robotic interfaces; Telepresence; Brain-machine interfaces. AR, Tools, Platforms, and Fundamentals: Mobile augmented reality; AR development tools and platforms – Unity3D, AR software development kits, ARToolkit, ARCore; Setting Up Unity 3D for AR; AR Fundamentals in Unity 3D.
Second series internal examination including theory + Laboratory if any	
Core Compulsory Readings 1. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2023. 2. Foundations of Sensation and Perception (2nd Edition), Mather, G, Psychology Press, 2009.	

3. Fundamentals of Computer Graphics (3rd Edition), Shirley, P., Ashikhmin,

M., &Marschner, S. A K Peters/CRC Press, 2009.

4. The VR book: Human-centered design for virtual reality, Jerald, J,

Morgan & Claypool, 2015.

5. Understanding augmented reality: Concepts and applications, Craig, A.

B, Morgan Kaufmann Publishers, 2013.

Core Suggested Readings

1. 3D User Interfaces: theory and practice, LaViolaJr, J. J., Kruijff, E.,

McMahan, R. P., Bowman, D., &Poupyrev, I. P, AddisonWesley, 2005.

2. Handbook of virtual environments: Design, implementation, and

applications, Hale, K. S., &Stanney, K. M. (Eds.), CRC Press, 2015

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA317 GAME DEVELOPMENT

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA317**	4	60

Learning Approach (Hours/ Week)	Marks Distribution	Duration of

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	ESE (Hours)
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.Embark on a journey into the dynamic world of game development with our comprehensive course. Designed for both beginners and enthusiasts alike, this course offers a holistic understanding of the game development process. From foundational concepts in design and programming to advanced techniques in graphics, networking, and AI, students will gain hands-on experience using industry-standard tools such as Unity. Through a blend of theory, practical exercises, and project-based learning, participants will emerge equipped with the skills and knowledge necessary to bring their game ideas to life across various platforms. Whether aspiring to join the thriving game industry or simply seeking to unleash their creativity, this course provides a solid foundation for all game development enthusiasts.

Course Objectives:

● Provide a comprehensive understanding of game development fundamentals, tools, and trends. C132 ● Develop proficiency in game design, prototyping, and development using Unity and other industry-standard tools. ● Explore advanced concepts like scripting, AI, multiplayer networking, and optimization techniques. ● Foster hands-on, project-based learning to apply acquired skills and build a strong portfolio for real-world scenarios.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Gain a solid grasp of game development fundamentals, tools, and trends.
CO2	Become proficient in game design, prototyping, and development using industry-standard tools like Unity
CO3	Master advanced concepts including scripting, AI, multiplayer networking, and optimization techniques
CO4	Build a strong portfolio, preparing them for real-world scenarios and enhancing employability in the game industry

Mapping of COs to PSOs

CO - PSO Mapping

	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Introduction to Game Development Overview of game development: history, trends, and current landscape, Basics of game design principles, Introduction to game engines and their importance, understanding game assets: graphics, sound, and music, Basics of programming languages commonly used in game development (e.g., C#, C++, Java) (15 hours)
UNIT 2	Game Design and Prototyping Deep dive into game design: mechanics, dynamics, and aesthetics, Prototyping techniques and tools, User experience (UX) and user interface (UI) design for games, Playtesting and iteration, Introduction to version control systems for collaborative development (e.g., Git), Assignment: Create a prototype of the game designed in Module 1 and conduct playtesting sessions for feedback. (15 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	Introduction to Unity game engine, Understanding Unity's interface and workflow, Scene creation and management, Scripting in C# for game logic, Implementing player controls and interactions, Introduction to 2D and 3D game development in Unity, (15 hours)
UNIT IV	Advanced Game Development Concepts Advanced scripting and programming techniques in Unity, Working with physics and animations, Implementing artificial intelligence (AI) for non-player characters (NPCs), Multiplayer game development concepts and networking, Optimizing games for performance and scalability, Publishing games on various platforms (PC, mobile, console) (15 hours)
UNIT X	Advanced graphics programming: shaders, lighting, and postprocessing effects, Procedural content generation (PCG) techniques for creating dynamic and varied game worlds, Introduction to virtual reality (VR) and augmented reality (AR) game development, Advanced networking and multiplayer synchronization techniques, Integrating machine learning for AI behavior and procedural generation, Performance optimization strategies for resource-intensive games(15 hours)

Second series internal examination including theory + Laboratory if any
Core Compulsory Readings 1. Book: "Game Design Workshop: A Playcentric Approach to Creating Innovative Games" by Tracy Fullerton 2. Book: "The Art of Game Design: A Book of Lenses" by Jesse Schell Core Suggested Readings 1. Book: "Unity in Action" by Joe Hocking 2. Book: "Game Programming Patterns" by Robert Nystrom 3. Book: "Real-Time Rendering" by Tomas Akenine-Möller, Eric Haines, Naty Hoffman 4. Book: "Game AI Pro" edited by Steve Rabin

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA318 COMPUTER VISION

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA318**	4	60

Learning Approach (Hours/ Week)	Marks Distribution	Duration of

Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	ESE (Hours)
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.The course focuses on building a broad understanding on the fundamentals of computer vision. The objective of the course is to communicate theoretical knowledge about machine vision. This is a field of artificial intelligence that enables computers and systems to derive meaningful information from visual inputs and has the prospective to energies C176 scientific advancement through technological revolution

Course Objectives:

● To understand the basic concepts of computer vision and techniques involved ● Introduce applications of machine vision ● Illustrate knowledge about motion analysis

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the basic fundamentals of computer vision
CO2	Describe various aspects of object recognition
CO3	Illustrate various aspects of 3D vision
CO4	Analyze motion application techniques

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓

CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Introduction: Motivation, Difficulty, Image analysis tasks, Image representations, Image digitization, Image properties, Colour images, Cameras. Data Structures: Levels of image data representation - Traditional image data structures - Hierarchical data structures. (15 hours)
UNIT 2	Object Recognition: Knowledge representation, Statistical pattern recognition, Neural nets, Syntactic pattern recognition, Recognition as graph matching, Optimization techniques in recognition, Fuzzy systems (15 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	3D vision: 3D vision: Tasks - Basics of projective geometry - Scene construction from multiple views, Uses: Shape from X - Full 3D objects - 3D model based vision (15 hours)
UNIT IV	Motion Analysis: Differential motion analysis methods, Optical flow, Analysis based on interest points, Detection of specific motion patterns, , Motion models to aid tracking.(15 hours)
UNIT X	Texture: Statistical texture description, Syntactic texture description methods, Hybrid texture description methods, Texture recognition method applications ,2D view based 3D representation,Video Tracking (15 hours)
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. Szeliski, Richard, Computer Vision: Algorithms and Applications- 2ndEdn,

Springer Verlag, 2022.

2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis

and Machine Vision, Cengage Learning, New Delhi, 2014.

3. Wesley E. Synder and Hairong Qi, Machine Vision, Cambridge University

Press, USA, 2010.

Core Suggested Readings

1. Rafael C Gonzalez, Richard E Woods, Steven L Eddins, Digital Image

Processing, Pearson Education, New Delhi, 2009.

2. Shapiro L and Stockman G, Computer Vision, Prentice-Hall, 2001

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA319 Computational Photography

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA319**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.

The course "Computational Photography" explores the fusion of photography and computer science, focusing on innovative techniques to enhance, manipulate, and analyze digital images. Beginning with foundational concepts, it covers image enhancement, segmentation, deep learning, and applications in diverse fields.

Additionally, the course addresses ethical considerations. Through theoretical instruction and handson projects, students gain skills for careers in visual effects, virtual reality, and computer vision.

Course Objectives:

- Provide a comprehensive understanding of computational

photography techniques for enhancing digital images.

● Introduce advanced concepts such as image segmentation and deep

learning for image processing.

● Explore applications of computational photography in various fields,

including medicine and remote sensing.

● Address ethical considerations and social implications related to

computational photography practices.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Proficiency in applying computational photography techniques to enhance and manipulate digital images effectively.
CO2	Mastery of advanced concepts such as image segmentation and deep learning for image processing.
CO3	Ability to utilize computational photography methods in diverse applications such as medical imaging and remote sensing.
CO4	Awareness of ethical considerations and social implications associated with computational photography practices.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓

CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Fundamentals of Computational Photography: Introduction to computational photography: Definition, history, and significance. Basic concepts in digital imaging: Image formation, sensors, and image processing. Image enhancement techniques: Noise reduction, sharpening, and contrast adjustment. Computational photography algorithms: HDR imaging, panorama stitching, and image stacking. (15 hours)
UNIT 2	Image segmentation and object detection: Algorithms for identifying and isolating objects in images. C144 Image fusion and blending: Techniques for combining multiple images to enhance visual quality. Content-aware image manipulation: Tools and algorithms for intelligent image editing. Deep learning for image processing: Introduction to convolutional neural networks and their applications in image processing. (15 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	Computational Photography Applications: Computational imaging for mobile devices: Overview of smartphone camera technologies and their computational photography features. Computational photography in computer vision: Applications in augmented reality, object recognition, and scene understanding. Computational photography in medical imaging: Techniques for image enhancement, segmentation, and analysis in medical imaging applications. Computational photography in remote sensing: Applications in satellite imaging, environmental monitoring, and agriculture. (15 hours)
UNIT IV	Ethical and Social Implications of Computational Photography: Privacy and surveillance issues: Ethical considerations related to image capture, processing, and sharing. Bias and fairness in computational photography algorithms: Challenges and solutions for ensuring fairness and inclusivity. Creative manipulation and authenticity: Discussion on the impact of computational techniques on the authenticity of visual content. Regulatory frameworks and guidelines: Overview of existing regulations and guidelines governing computational photography practices.(15 hours
UNIT X	Computational Photography for Visual Effects and Virtual Reality: Introduction to visual effects (VFX) and virtual reality (VR) technologies. Techniques for image-based rendering and scene reconstruction in VFX and VR applications. Advanced image manipulation algorithms for creating realistic visual effects. Deep learning approaches to image synthesis and generation for VFX and VR. Computational photography in immersive storytelling and interactive C145 experiences. Case studies and applications of computational photography in the film industry and VR content creation. (15 hours)

Second series internal examination including theory + Laboratory if any
Core Compulsory Readings 1. Szeliski, R. (2010). Computer vision: Algorithms and applications. Springer Science & Business Media. 2. Forsyth, D. A., & Ponce, J. (2012). Computer vision: A modern approach. Prentice Hall. 3. Lu, J., & Shao, L. (2014). Perceptual computing: Aiding people in making subjective judgments. Springer. 4. Jacobs, C. E. (2018). Ethics in Computational Photography (1st ed.). MIT Press. Core Suggested Readings 1. Grossberg, M. D. (2017). Data Acquisition Techniques Using PCs (2nd ed.). Academic Press. 2. Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2018). Artificial Intelligence and Games (1st ed.). Springer. 3. Smith, W. A. (2019). Principles of 3D Image Analysis and Synthesis (1st ed.). Springer. 4. Elgammal, A., &Elhoseiny, M. (2020). The AI Renaissance: Essays on Artificial Intelligence and Its Implications (1st ed.). Springer

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7

KU06DSEMCA320 HIGH PERFORMANCE COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA320**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.

The course on High-Performance Computing (HPC) provides a comprehensive understanding of the principles, architectures, and techniques involved in executing computationally intensive tasks. Students will learn about parallel computing, distributed systems, and the utilization of HPC clusters for scientific simulations and data analysis. The course covers topics such as programming models, optimization strategies, and performance evaluation to maximize the efficiency of HPC applications.

Course Objectives:

● Understand the principles and foundations of high-performance computing and its importance in scientific and computational-driven fields.

● Gain knowledge of parallel computing architectures and techniques used in high-performance computing.

● Develop skills in programming and optimizing algorithms for efficient execution on HPC clusters.

● Learn to utilize advanced tools and frameworks for managing and analyzing large-scale data in high-performance computing environments.

● Apply the acquired knowledge and skills to solve complex

computational problems and conduct scientific simulations using highperformance computing resources.

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	The learner will be able to design, formulate, solve and implement high performance versions of standard single threaded algorithms
CO2	The learner will know and will be able to demonstrate the architectural features in the GPU and MIC hardware accelerators.
CO3	The learner will be able to design programs to extract maximum performance in a multicore, shared memory execution environment processor.
CO4	The learner will be able to design and deploy large scale parallel programs on tightly coupled parallel systems using the message passing paradigm.

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Levels of parallelism (instruction, transaction, task, thread, memory, function)- Models (SIMD, MIMD, SIMT, SPMD, Dataflow Models, Demand-driven Computation etc)- Architectures: N-wide superscalar architectures, multi-core, multi-threaded. (15 hours)
UNIT 2	Processor Architecture, Interconnect, Communication, Memory Organization, and Programming Models in high performance computing architectures: (Examples: IBM CELL BE, Nvidia Tesla GPU, Intel Larrabee C173 Microarchitecture and Intel Nehalem microarchitecture- Memory hierarchy and transaction specific memory design- Thread Organization. (15 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	Synchronization- Scheduling- Job Allocation-Job PartitioningDependency Analysis- Mapping Parallel Algorithms onto Parallel ArchitecturesPerformance Analysis of Parallel Algorithms. (15 hours)
UNIT IV	Bandwidth Limitations- Latency Limitations- Latency Hiding/Tolerating Techniques and their limitations- Power-aware Processing Techniques-Power-aware Memory Design- Power-aware Interconnect DesignSoftware Power Management. Petascale Computing-Optics in Parallel Computing- Quantum Computers- Recent developments in Nanotechnology and its impact on HPC. (15 hours)
UNIT X	Exascale Computing, Heterogeneous Computing, Parallel I/O and Storage Systems, Distributed Shared Memory (DSM), Extreme-Scale Data Analytics, Deep Learning and HPC, High-Performance Computing in Edge and IoT, (15 hours)
Second series internal examination including theory + Laboratory if any	

Core Compulsory Readings

1. George S. Almasi and AlanGottlieb, Highly Parallel Computing,

Benjamin Cumming Publishers.

2. Kai Hwang ,Advanced Computer Architecture: Parallelism, Scalability,

Programmability, McGraw Hill 1993

Core Suggested Readings

1. David Culler, Jaswinder Pal Singh, Anoop Gupta, Parallel Computer

Architecture: A hardware/Software Approach, Morgan Kaufmann,

1999.

C174

2. K. Hwang& Z. Xu, Scalable Parallel Computing – Technology,

Architecture, Programming., McGraw Hill 1998.

3. William James Dally and BrianTowles, Principles and Practices on

Interconnection Networks, Morgan Kauffman 2004.

4. Hubert Nguyen , GPU Gems 3, Addison Wesley, 2008, (Chapter 29 to

Chapter 41)

5. AnanthGrama, Anshul Gupta, George Karypis, and Vipin Kumar,

Introduction to Parallel Computing, , 2nd edition, Pearson, 2003.

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

KU06DSEMCA321 GREEN COMPUTING

Semester	Course Type	Course Level	Course Code	Credits	Total Hours
6	DSE	300	**KU06DSEMCA321**	4	60

Learning Approach (Hours/ Week)			Marks Distribution			Duration of ESE (Hours)
Lecture	Practical/ Internship	Tutorial	CE	ESE	Total	
4	0	1	50	50	100	2

*** ESE Duration: 2 hours for theory**

Course Description:.

The course focuses on building a broad understanding on the fundamentals of computer vision. The objective of the course is to communicate theoretical knowledge about machine vision. This is a field of artificial intelligence that enables computers and systems to derive meaningful information from visual inputs and has the prospective to energies C176 scientific advancement through technological revolution

Course Objectives:

● To understand the basic concepts of computer vision and techniques involved ● Introduce applications of machine vision ● Illustrate knowledge about motion analysis

Course Outcomes:

At the end of the Course, the Student will be able to:

SL #	Course Outcomes
CO1	Explain the basic fundamentals of computer vision
CO2	Describe various aspects of object recognition
CO3	Illustrate various aspects of 3D vision
CO4	Analyze motion application techniques

Mapping of COs to PSOs

CO - PSO Mapping					
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**
CO1	✓	✓	✓	✓	✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓

COURSE CONTENT

UNIT 1	Introduction: Motivation, Difficulty, Image analysis tasks, Image representations, Image digitization, Image properties, Colour images, Cameras. Data Structures: Levels of image data representation - Traditional image data structures - Hierarchical data structures. (15 hours)
UNIT 2	Object Recognition: Knowledge representation, Statistical pattern recognition, Neural nets, Syntactic pattern recognition, Recognition as graph matching, Optimization techniques in recognition, Fuzzy systems (15 hours)
First series internal examination including theory + Laboratory if any	
UNIT 3	3D vision: 3D vision: Tasks - Basics of projective geometry - Scene construction from multiple views, Uses: Shape from X - Full 3D objects - 3D model based vision (15 hours)
UNIT IV	Motion Analysis: Differential motion analysis methods, Optical flow, Analysis based on interest points, Detection of specific motion patterns, , Motion models to aid tracking.(15 hours)
UNIT X	Texture: Statistical texture description, Syntactic texture description methods, Hybrid texture description methods, Texture recognition method applications ,2D view based 3D representation,Video Tracking (15 hours)
Second series internal examination including theory + Laboratory if any	

1. Szeliski, Richard, Computer Vision: Algorithms and Applications- 2ndEdn,

Springer Verlag, 2022.

2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis

and Machine Vision, Cengage Learning, New Delhi, 2014.

3. Wesley E. Synder and Hairong Qi, Machine Vision, Cambridge University

Press, USA, 2010.

Core Suggested Readings

1. Rafael C Gonzalez, Richard E Woods, Steven L Eddins, Digital Image

Processing, Pearson Education, New Delhi, 2009.

2. Shapiro L and Stockman G, Computer Vision, Prentice-Hall, 2001

TEACHING LEARNING STRATEGIES

- Lecturing, case study/mini projects, Team Learning, presenting seminars on selected topics, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Video tutorial

ASSESSMENT RUBRICS

Refer to section 7