KANNUR UNIVERSITY

MASTER OF COMPUTER APPLICATIONS
(Choice Based Credit Semester System)
Regulations, Curricula, Syllabus and Scheme of Evaluation
(With effect from 2020 admission onwards)

REGULATIONS

1. **Duration of the MCA programme** shall be 2 years, divided into 4 semesters. Each semester shall have 18 weeks. The maximum period of completion is eight semesters (4 years).
2. **Eligibility for Admission:**

2.1 A pass in BCA/ Bachelor Degree in Computer Science/Engineering or equivalent degree recognized by the Kannur University

OR

A pass in B.Sc./ B.Com/B.A with Mathematics at 10 + 2 Level or at Graduation Level or any equivalent degree recognized by the Kannur University with additional bridge course as per the norms of the Kannur University.

2.2 Obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying examination.

3 Programme Structure: Programme Structure means a list of courses (Core, Elective and Open Elective) that makes up an Academic Programme, specifying the syllabus, Credits, hours of teaching, evaluation and examination schemes and the minimum number of credits required for the successful completion of the programme prepared as per the University Rules.

3.1 Choice Based Credit System: The CBCS provides an opportunity for the students to select both elective and open elective courses from the prescribed pool of courses.

3.2 Credits: The total minimum credits required to complete MCA programme is 80.

3.3 Core Course: Core Course means a course that a student admitted to a particular programme must successfully complete to receive the degree and which cannot be substituted by any other course.

3.4 Elective Course: Elective Course means an optional course to be selected by a student from the pool of courses.

3.5 Open Elective: Open Elective Course means an optional course to be selected by a student out of such courses offered in any other Department/Centre. Students of other Department will opt these courses subject to fulfilling of eligibility criteria as laid down by the Department offering the course.

3.6 Theory and Practical courses: The evaluation scheme for each Theory/ Practical course shall contain two parts; (a) Continuous Assessment (CA) and (b) End Semester Evaluation (ESE); 40% marks shall be given to CA and the remaining 60 % to ESE.

3.7 Attendance: The minimum attendance required for each semester shall be 75% of the total number of

classes conducted for that semester. Those who secure the minimum attendance in a semester alone will be allowed to register for the End Semester Examination. Condonation of shortage of attendance may be granted as per Kannur University PG regulations.

3.8. Continuous Assessment (CA)

Theory: The components of theory evaluation are as follows:

	COMPONENTS	% OF MARKS
I	Test paper	40%
Ii	Assignment	20%
Iii	Case Study/Seminar/Viva	40%

i. **Test Papers:** There shall be a minimum of two test papers to be conducted for each course. If more than two test papers are conducted, then two best scores shall be taken for the award of Internal Assessment (IA) marks. The dates of test papers shall be announced well in advance and the marks should be displayed in the notice board within one week of the test paper. Online tests such as objective type and open text book test paper (online or offline mode) also may be opted for conducting the test papers.

ii. **Assignments:** Two or more assignments (including practical assignments) shall be given for each course. The mode of assessment of the assignments shall be decided by the faculty concerned with due approval from the department council and shall be declared at the beginning of the semester. (It is suggested that to the extent possible, give individual assignments and also conduct short viva based on the assignment submitted). Assignment shall include practical assignments also. All the assignment works can be conducted either offline/ online mode. If a faculty wishes to conduct the internal assessment mode online, she/ he can be utilize any online learning platform (like ***MOOCs*** etc.) for the evaluation.

iii. MOOC course

(a) Each student shall successfully complete one MOOC course in the second semester courses (*Communicative English, Financial Accounting, Android programming, Windows programming, Environmental studies or any new topic/programming language/technology introduced recently etc*.) with a minimum period of eight (08) weeks offered by reputed organizations like NPTE: or Coursera.

(b) The Department should arrange facilities to the student for the online course and should provide one faculty in the department as in-charge of the program for giving guidance for choosing an appropriate online course. The concerned faculty-in charge will be responsible for the internal evaluation of the above program.

(c) Each student shall opt at least one MOOC course(s) at the beginning of each semester, in consultation with the faculty in-charge. The list of MOOC course opted by the students shall be placed before the Department council. The council may approve or reject the proposals on valid grounds. Once the list of MOOC course is finalized, the same may be forwarded to the University. The department shall ensure that the list is finalized and forwarded to the University at the beginning of each semester.

(d) The MOOC course completion certificate shall be submitted to the faculty in charge, who will verify it and forward the same along with the internal assessment marks to the Head of Department. After due verification, Head of Department shall forward the same to the University for awarding credit for the course.

iv. Practical: The Components of CA for practical courses except Case study as follows:

	COMPONENTS	% OF MARKS
i	Lab Test	40%
ii	Completion of the list of Lab assignments prescribed by the faculty	20%
iii	Periodical assessment of assignment in the Lab & viva	40%

v. Case Study:

	COMPONENTS	% OF MARKS
i	Periodical viva / short quizzes /short programming assignments to evaluate the basic knowledge/understanding of the topic/Tool.	30%
ii	Coding – Logic, Selection of appropriate constructs / features of the Tool, Style etc.	30%
iii	Execution of the case study - output	20%
iv	Viva based on case study	20%

vi. Mini Project:

	COMPONENTS	% OF MARKS
i	Understanding of the problem / concepts	20%
ii	Adhering to methodology	15%
iii	Quality of presentation and demonstration	15%
iv	Quantum of work / effort	25%
v	Organization and content of Project report	5%
vi	Viva based on Project	20%

Note: All the records in respect of Continuous Assessment (CA) must be kept in the department and must be made available for verification by the University. The results of the CA shall be displayed on the notice board within 5 working days from the last day of a semester. It should be get signed by the candidates. The marks awarded for various components of the CA shall not be rounded off, if it has a decimal part. The total marks of the CA shall be rounded off to the nearest whole number

3.9 End Semester Evaluation (ESE):

There shall be single valuation system of answer books by the Kannur University. However, there is a provision for revaluation, as per the University regulation.

a. Pattern of questions: Questions shall be set to assess knowledge acquired, standard application of knowledge, application of knowledge in new situations, critical evaluation of knowledge and the ability to synthesize knowledge.

b. Question paper for end semester theory examination shall consist of:

The ESE shall be made based on examinations for each course conducted by Controller of Examinations as per the common norms under the CCSS. The question paper for ESE for Theory Examinations shall contain three sections. The Question paper should contain minimum 3 questions from each unit and shouldn't contain more

than 5 questions from the same unit.. The distribution of the no of questions and marks are given in the following table.

Part	Marks	Number of questions to be answered	Number of questions in the question paper	Type of questions (Level - Bloom's Taxonomy)
A	15	5	6	1 Remembering 2 Understanding
B	15	3	5	6. Creating
C	30	3	5	3. Applying 4. Analysing 5. Evaluating
Total	60	11	16	

c. Practical Examination:

Practical Examination shall be conducted and evaluated by two examiners one internal and one external. Details of evaluation of ESE practical courses are given along with respective syllabus and the scheme of evaluation will be finalized by the Board of Examiners, from time to time.

3.10 Project: A project work has to be undertaken by all students. The project can be software development following all or some of the software development lifecycle or an R & D project. The hours allotted for project work may be clustered into a single slot so that students can do their work at a centre or location for a continuous period of time. The Major project work should be carried out in the Department /Institution or in an Industry / R & D organization of national repute. Project work shall be carried out under the supervision of a Teacher. If the project is carried out in an Industry / R & D organization outside the campus, then a co-guide shall be selected from the concerned organization. If the project work is of interdisciplinary nature, a co-guide shall be taken from the other department concerned. Every student should do the Project individually and no grouping is allowed. The candidates are required to get the synopsis and the guide approved by the department before the commencement of the project. A co-guide should be a postgraduate in CS/ Application/ IT or allied subject or a person of eminence in the area in which student has chosen the project. At the end of the semester the candidate shall submit the Project report (two bound copies and one soft copy) duly approved by the guide and co-guide for End Semester Evaluation. The project report shall be prepared according to the guidelines appended along with this regulations/ Guidelines approved by the University.

Evaluation of Project:

i. A Departmental committee duly constituted by the Head of the Department will review the project periodically.

ii. Continuous Assessment of project work: There shall be three internal presentations before the committee (Minimum two members, including the guide). The assessment is based on presentation, interim report and viva voce. The total mark for CA shall be divided among the **three** presentations in the ratio **20%:30%:50%.** Each internal presentation shall be evaluated based on the following components:

	COMPONENTS	% OF MARKS
i	Understanding of the problem / concepts	20%
ii	Adhering to methodology	15%
iii	Quality of presentation and demonstration	15%
iv	Quantum of work / effort	25%
v	Organization and content of Project report	5%
vi	Viva based on Project	20%

iii. End Semester Assessment of Project: A board of two examiners appointed by the University shall conduct ESE evaluation. The evaluation shall be based on the report, presentation of the work, demonstration of the work and a detailed viva voce based on the work carried out. A candidate will not be permitted to attend the Project evaluation without duly certified project reports. Also, a project will be evaluated only if the candidate attends the ESE presentation and Viva voce on the scheduled date and time. A board shall evaluate a maximum of 10 candidates in a day. The End Semester evaluation shall consist of the following components:

	COMPONENTS	% OF MARKS
I	Understanding of the problem/requirements/ concepts related to the project	15
Ii	Adhering to methodology (Software engineering phases or research methodology) and the candidates understanding of the components of methodology	15
Iii	Quality of Modelling of the problem and solution/ database design / form design / reports / testing (For research projects - relevance /novelty of the work(s)/ use of data/ proposal of new models /analysis of algorithms/ comparison and analysis of results /findings)	20
Iv	Quality of presentation / demonstration	15
V	Quantum of work / effort - assessed through the content of report, presentation and viva	25
Vi	Organization and content of report	10

iv. A student shall pass in the Project course if she/he secures a separate minimum of 40 % for the external and 40% for ESE and CA put together.

v. If a candidate fails in the evaluation of Project, he/she has to repeat the project course along with the next batch and undergo both CA and ESE. Unlike theory/practical courses, the CA mark will not be retained.

vi. There shall be no improvement chance for the marks obtained in the Project course.

3.11 Guideline for preparing project Report (Both Mini and Major Project)

i. Arrangement of contents:

The sequence in which the project report material should be arranged and bound should be as follows:

1. Cover Page & Title Page
2. Plagiarism Report

3. Bonafide Certificate
4. Abstract
5. Table of Contents
6. List of Tables
7. List of Figures
8. List of Symbols, Abbreviations and Nomenclature
9. Chapters

The chapters may be broadly divided into 3 parts (i) Introductory chapter, (ii) Chapters developing the main theme of the project work (iii) implementation details (if any) and Conclusion. The main text will be divided in to several chapters and each chapter may be further divided into several divisions and sub-divisions. Each chapter should be given an appropriate title.

Tables and figures in a chapter should be placed in the immediate vicinity of the reference where they are cited.

Footnotes should be used sparingly. They should be typed single space and placed directly underneath in the very same page, which refers to the material they annotate.

10. Appendices
11. References

The tables and figures shall be introduced at appropriate places.

ii. **Page Dimension and Binding Specifications**:

The dimension of the project report should be in **A4** size. The project report should be bound using flexible cover of the thick white art paper. The cover should be printed in black letters and the text for printing should be identical.

iii. All the project report submitted by the students should be plagiarism checked using **Turnitin software** and the plagiarism report generated by the software should be verified and signed by the Head of the Department or person-in charge of the Project Coordinator.

3.12 Research Seminar:

The research seminar helps the student to nurture scientific temper. During the preparation for research seminar, students will get an opportunity to work with leading researchers and experts which helps the students to refine their skills, searching scholarly information, text processing and presentation. This helps them to integrate their understanding; includes defining the hypothesis, providing a rationale for it and selecting a line of argument. Students then write term papers to showcase and review the results obtained during research seminars.

Sl. No	Components	Marks.Max (50)
1	Report	16
2	Presentation and defense	24
3	Topic & content organization of presentation	10

3.13 Viva Voce: A general Viva Voce covering all courses in the Programme shall be conducted in the fourth semester. The Viva voce shall be conducted by two external examiners. The Viva voce shall **not** be clubbed

with the project evaluation. The details of the mode of conduct and evaluation of Viva Voce shall be decided by the BOE.

4. Grading system

4.1 An alphabetical grading system shall be adopted for the assessment of student's performance in a course. The grade is based on a ten-point scale. The following table gives the range of marks, grade points and the alphabetical grade.

Range of Marks (%)	Grade points	Alphabetical grade
90-100	9	A+
80-89	8	A
70-79	7	B+
60-69	6	B
50-59	5	C
<50	0	F

4.2. A minimum of grade point5 (Grade C) is needed for the successful completion of a course.

4.3. Performance of a student at the end of each semester is indicated by the Grade Point Average (GPA) and is calculated by taking the weighted average of grade points of the courses successfully completed. Following formula is used for the calculation. The average will be rounded off to two decimal places.

GPA = Sum of (grade points in a course multiplied by its credit) /Sum of credits of courses

4.4. The overall performance of a student is indicated by the Cumulative Grade Point Average (CGPA) and is calculated using the same formula given above.

4.5. Empirical formula for calculating the percentage of marks will be CGPA × 10 + 5.

4.6. Based on CGPA overall letter grade of the student shall be in the following way.

CGPA	Overall letter grade
8.5 and above	A
7.5 and above but less than 8.5	A
6.5 and above but less than 7.5	B
5.5 and above but less than 6.5	B
4.5 and above but less than 5.5	C

4.7. Conversion of Grades into classification

Classification	Overall Letter Grade
First class with Distinction	A+ and A
First class	B+ and B
Second class	C

4.8. Supplementary Examinations for Failed Candidates:

i) Candidates who have failed (F grade) in the semester examinations (except project work) can appear for the failed papers for the particular semester along with regular students. However, the Continuous Evaluation (CE) marks shall remain the same. Two such supplementary chances will be given for each semester within two years.

ii) In the event of failure in Project Work the candidate shall re-register for project work, re do the project work and resubmit the project report a fresh for evaluation. The Continuous Evaluation marks shall be freshly allotted in this case.

4.9. Appearance for Continuous Evaluation and End Semester Evaluation are compulsory and no grade shall be awarded to a candidate if he/she is absent for CE/ESE or both.

4.10. A student who fails to complete the programme / semester can repeat the full programme / semester once, if the department council permits so.

4.11. There shall be no provision for improvement of CE or ESE.

4.12. No student shall be allowed to take more than eight consecutive semesters for completing MCA programme from the date of enrolment.

5. Grade Card: The Controller of Examination shall issue the Grade cards of all semesters and consolidated grade statement and certificates on completion of the programme, based on the authenticated documents submitted by the Head of the Department.

6. Grievance Redressal Mechanism

6.1 Committees will be constituted at the Department and University levels to look into the written complaints regarding continuous Evaluation (CE). Department Level Committee (DLC) will consist of the Department Council and a student nominee of the Department Student's Union from the concerned faculty.

6.2. University Level Committee (ULC) will consist of the Pro-Vice-Chancellor (Chairman and Convener), the Convener of the Curriculum Committee (vice-chairman), the Head of the Department concerned and a nominee of the Students' Union. Department Level Committee will be presided over by the HOD and the University Level Committee by the Pro-Vice Chancellor. Department Level Committee will have initial jurisdiction over weeks of publication of results of CE and disposed of within two weeks of receipt of complaint. Appeals to university Level Committee should be made within one month of the decision taken by the Department level committee and disposed within two months of the receipt of the complaint.

6.3. Complaints unsolved by the University Level Grievance committee shall be placed before the Vice Chancellor.

PROGRAMME SPECIFIC OUTCOME OF MCA PROGRAMME

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 to setup Computer Networks.
PSO5	Design, build up, put into practice and test software systems to meet the given specifications by following the principles of Software Engineering.
PSO6	Furnish the candidate to do the real time jobs linked with Information Technology and Computer Application.

COURSE STRUCTURE

(With effect from 2020 admission onwards)

Semester	Core	Elective	Open Elective	Practical	Project	Total
I	**15**	**0**	**0**	**6**	**0**	**21**
II	**14**	**0**	**0**	**8**	**0**	**22**
III	**9**	**3**	**4**	**2**	**2**	**20**
IV	**2**	**9**	**0**	**0**	**6**	**17**
Total	**40**	**12**	**4**	**16**	**8**	**80**

SEMESTER I

Subject Code	Subject	Instructional Hours/Week			Marks			C
		L	P	T	ESA	CA	Tot	
MCAIT01C01	Digital Fundamentals and Computer Organization	3	0	0	60	40	100	3
MCAIT01C02	System software and Operating systems	3	0	0	60	40	100	3
MCAIT01C03	Python Programming	4	0	0	60	40	100	2
MCAIT01C04	Computer Network and Linux Administration	3	0	0	60	40	100	3
MCAIT01C05	Discrete Structures and Optimization	3	0	0	60	40	100	3
MCAIT01P01	Lab- I: Python Programming	0	5	1	60	40	100	3
MCAIT01P02	Lab- II: Network and Linux Administration	0	4	1	60	40	100	3
MCAIT01C06	Research Seminar	0	0	3	0	50	50	1
Total		16	9	5	420	330	**750**	**21**

SEMESTER II

Subject Code	Subject	Instructional Hours/Week			Marks			C
		L	P	T	ESA	CA	Tot	
MCAIT02C07	Algorithm and Data Structure	4	0	0	60	40	100	3
MCAIT02C08	Programming in Java	3	0	0	60	40	100	3
MCAIT02C09	Database Management Systems	3	0	0	60	40	100	3
MCAIT02C10	Web Technology	3	0	0	60	40	100	2
MCAIT02C11	Software Engineering	3	0	0	60	40	100	3
MCAIT02P03	Lab- III: i) Data Structure; ii) Java	0	5	0	60	40	100	3
MCAIT02P04	Lab- IV: i) DBMS; ii) WT	0	4	0	60	40	100	3
MCAIT02P05	Lab- V: MOOC course (Case Study)	0	0	5	0	50	50	2
Total		16	9	5	420	330	**750**	**22**

Number of Open Elective Course: 1

Open Elective Course		Theory	Practical	Tutorial	Credits
MCAIT02O01	Design and Analysis of Algorithms	3	1	0	4
Total credits in elective courses		3	1	0	4

SEMESTER III

Subject Code	Subject	Instructional Hours/Week			Marks			C
		L	P	T	ESA	CA	Tot	
MCAIT03C12	Machine Learning Techniques	3	0	0	60	40	100	3
MCAIT03C13	Theory of Computation	3	0	0	60	40	100	3
MCAIT03C14	Big Data Analytics	3	0	0	60	40	100	3
Offered by other departments	Open Elective	4	0	0	60	40	100	4
MCAIT03E01 /02/03/04/0 5/06	Elective – I	3	0	0	60	40	100	3
MCAIT03P06	Lab- VI: Machine Learning	0	5	3	60	40	100	2
MCAIT03P07	Mini Project	0	4	2	60	40	100	2
Total		16	9	5	420	280	**700**	**20**

SEMESTER IV

Subject Code	Subject	Instructional Hours/Week			Marks			C
		L	P	T	ESA	CA	Tot	
MCAIT04E07 /08/09/10/1 1/12	Elective II	3	0	0	60	40	100	3
MCAIT04E13 /14/15/16/1 7/18	Elective- III	3	0	0	60	40	100	3
MCAIT04E19 /20/21/22/2 3/24	Elective-IV	3	0	0	60	40	100	3
MCAIT04C15	Project	0	16	5	120	80	200	6
MCAIT04C16	General Viva	0	0	0	100	0	100	2
Total		9	16	5	400	200	**600**	**17**

ELECTIVES:

POOL A (Elective- I) Subject Code: MCAIT03EXX

MCAIT03E01 ARTIFICIAL INTELLIGENCE
MCAIT03E02 BIOINFORMATICS
MCAIT03E03 FUZZY SETS & SYSTEMS
MCAIT03E04 GRAPH THEORY & COMBINATORICS
MCAIT03E05 SOFTWARE ARCHITECTURE
MCAIT03E06 FOUNDATIONS OF NATURAL LANGUAGE PROCESSING

POOL B (Elective- II) Subject Code: MCAIT04EXX

MCAIT04E07 DATA MINING AND WAREHOUSING
MCAIT04E08 ALGORITHMS IN COMPUTATIONAL BIOLOGY
MCAIT04E09 OBJECT ORIENTED ANALYSIS AND DESIGN
MCAIT04E10 COMPUTER VISION
MCAIT04E11 SOFTWARE PROJECT MANAGEMENT
MCAIT04E12 VISUAL CRYPTOGRAPHY

POOL C (Elective- III) Subject Code: MCAIT04EXX

MCAIT04E13 PATTERN RECOGNITION
MCAIT04E14 CYBER FORENSICS
MCAIT04E15 NATURAL LANGUAGE PROCESSING WITH PYTHON
MCAIT04E16 GRID AND CLOUD COMPUTING
MCAIT04E17 INFORMATION SECURITY
MCAIT04E18 BIOMETRIC IMAGE PROCESSING

POOL D (Elective- IV) Subject Code: MCAIT04EXX

MCAIT04E19 DATA AND INFORMATION VISUALIZATION
MCAIT04E20 INFORMATION RETRIEVAL SYSTEM
MCAIT04E21 OPERATIONS RESEARCH
MCAIT04E22 COMPUTER GRAPHICS WITH OpenGL
MCAIT04E23 DESIGN AND ANALYSIS OF ALGORITHMS
MCAIT04E24 NATURE INSPIRED COMPUTING

Open Elective Course Subject Code: MCAIT02O01

MCAIT02O01 DESIGN AND ANALYSIS OF ALGORITHMS

BRIDGE COURSE

1. Introduction.

The bridge program comprises 80 hours teaching and learning activity. It consists of two theory papers and one laboratory papers. This course shall be conducted during the first semester of the MCA programme without affecting the actual work load of the semester. The course shall be offered in the department at which the candidates enrol for the MCA program. The mode conduct of the course is completely under the strict control the department at which the MCA program is offered. Total eighty (80) hours teaching and learning activities shall be completed before the notification of Ist semester examination by the University. The department has to complete the course by conducting classes and evaluation of the students before the commencement of the Ist semester MCA examination by the University. All those students who successfully complete the bridge course shall be given completion certificate by the department. The list of all successful candidates shall be forwarded to the University along with Continuous Assessment mark list of the Ist semester MCA program.

2. Conduct of classes

Department council shall schedule regular classes (may be online class – preferred MOOC) and completed eighty (80) hours programme before the Ist semester MCA end semester examination notification by the University. The classes shall be conducted either in the weekend mode or regular working day without affecting the actual regular teaching and learning activities of the Ist semester MCA curriculum.

3. Duration of the program

The course shall comprise three (03) papers i.e. two (02) theory papers and one (01) practical paper. Candidate has to appear examinations for all above papers at the end of the program conducted by the Department at which candidate has registered for the MCA program. The details of subjects and corresponding examination details are mentioned in the curriculum.

4. Conduct of examination

At the end of the course, department has to conduct the examinations on each theory paper with two (02) hours duration and complete the evaluation process of all those papers within two (02) weeks. The pattern of question papers and evaluation criteria for passing examinations are specified in the regulation.

5. Pattern of question paper for theory papers

Sl.	Question Type	Number Of Questions	No. Of Questions to be answered	Marks/ question	Max. Marks
1	Single word/MCQ/Fill in the blank	10	10	10 × 1 = 10	10
2	Short answer	05	05	05 × 2 = 10	10
3.	Short essay	05	03	03 × 4 = 12	12
4	Essay	04	02	02 × 9 = 18	18

6. Question Paper preparation

The faculty in-charge of the each course shall prepare three (03) unique set of question papers on the subject s(he) taught. Faculty should give utmost care in preparing the question paper. The question paper should contain four (04) different sections titled as Part A, Part B, Part C and Part D. In part A out of ten (10) questions, two (02) questions from each unit, for part B and Part C, out of five (05) questions, single (01) question from each unit and finally Part D comprises any four (04) questions from all the five (05) units. After preparing the question paper, faculty-in-charge shall submit these question papers to the Head of the

department in sealed cover. The Head of the department shall then constitute question paper scrutiny committee comprising the Head of the department and two more senior faculties other than faculty-in charge of any course in Bridge program for scrutinizing the question papers submitted by the faculty-in-charge and finalize the question papers for the examinations.

7. Conduct of practical examination

At the end of the course, department shall conduct a practical examination for the course BR03 Lab- C programming Language by appointing two faculties in the department and provide a printed question paper which comprises of list ten questions and out of which faculties have to assign one questions on checking the skill of C programming construct and another one related to the numerical methods taught in Module V of the BR02 Course. The evaluation of the practical examination shall be done as follows for each two questions given to the students:

Sl.	Components	Marks
1	Writing Algorithm/Flow Chart	10
2	Program writing and compilation using system	10
3	Correct output	05

8. Theory paper evaluation

The Head of the department shall constitute a Board of Examiners (BoE) by including all the faculties in the department (minimum three faculties) with the Head of the department as the chairman. The BoE prepare the scheme and criteria for the evaluation of the answer books of the students in the Bridge course and the evaluation shall be completed within two weeks after the examinations of the Bridge Program. Only single valuation is enough.

9. Finalizing the results of Bridge program

The BoE shall conduct a pass board meeting soon after completing the evaluation of the answer books and related tabulation works. The students who receive (40%) marks in each subject including the practical examination in total (50%) shall be placed as successful completion of the program. All the documents including the tabulation registers regarding the conduct of the examinations shall be kept in the department and the same shall be produced to the University as when needed/requested. All the successful students list shall be forwarded to the University soon after publishing the results.

10. Supplementary chance

A candidate who fails to secure minimum marks (40%) for a pass in a course will be permitted to write the same examination one more time after three months of the completion of program. The students who do not complete the bridge program within one year shall not be registered for IInd semester MCA end semester examination conducted by the University and no further promotion shall be allowed for subsequent semesters too.

11. Scheme and curriculum for Bridge Program

Subject Code	Subject	Instructional Hour/Week (30 Hours/paper)			Marks	Credit
		L	P	Total		
BR01	Basics of Computing	30	----	30	50	0
BR02	Mathematical Foundations	30	----	30	50	0
BR03	LAB – C Programming Language	---	20	20	50	0

BR01 BASICS OF COMPUTING

UNIT I

Introduction to Programming:- Algorithms- Problem -Solving aspect – Implementation of algorithms – Properties of algorithms – The efficiency of algorithms – Flow chart- Pseudo Code, Programs and Programming Languages - compiler – Interpreter, Loader and Linker - Program execution – Classification of Programming Language-Structured Programming Concept- Top-down and bottom-up approaches. (05 Hours)

UNIT II

Features of C, Evolution of C, Structure of a C Program, Compiling a C Program-C Character sets-identifiers-data types-keywords-statements- variable and constants- tokens-Operators- Storage classes-auto, register, static, extern, typedef- Type casting, I/O Functions. Control Constructs-Control statements-Conditional, switch Statements- Loops and Jumping statements - break, continue and goto Statement. (07 Hours)

UNIT III

Introduction to Functions, Function Declaration and Prototypes, Storage Classes, Recursion, call by value and call reference. Arrays-One Dimensional Array, Two Dimensional, Strings, Linear search and Binary search algorithms.

Understanding memory addresses- address operator- pointer- use of pointers- arrays and pointers – pointers and strings - array of pointers- pointer to pointer. Structure Definition-Structure Initialization- Arrays of Structures-Arrays within Structures-Structures within Structures-Structure Pointers. Union–Definition and Declaration- Accessing a Union Member-Initialization of a Union Variable- Use of User Defined Type Declarations. Introduction to File Handling in C- File- Defining and Opening a File- Reading and Writing in Files Reading and writing Data- Sequential File- Functions for Random Access to Files. (11 Hours)

UNIT IV

Introduction to computer- Components-architecture- types of computers-classification-CPU-types, speed, classification-memory: RAM, ROM, Cache, Secondary memory -I/O devices. Introduction to software-Operating systems-system software- types of software-types of operating systems. Network:- LAN,WAN,MAN, topology, networking devices. Internet:- IP address, classification, need of IP address, Working of IP address, WWW,URL, Domain names, Internet services and service providers, ISPs. Mobile Technology:-Cellular System Generations-Types of Mobile Devices, Types of mobile operating systems. IoT:-what and how, structure of IoT, IoT applications (Familiarity only).Block chain technology: - Basic awareness and definitions.(07 Hours)

Text Book:

1. V Rajaraman, Neeharika Adabala, Fundamentals of computers, 6th edition, PHI.
2. Balagurusamy, Programming in ANSI C, 5thedn, TMH.

3. Pradeep K Sinha and Priti Sinha, Computer Fundamentals.

BR02 MATHEMATICAL FOUNDATIONS

UNIT I

Number systems: Decimal numbers, binary numbers, decimal-to-binary conversion-Binary arithmetic-1's and 2's complements-signed numbers- Arithmetic operations with signed numbers- Octal numbers-Hexadecimal Numbers-BCD numbers- Digital codes. Digital and Analog quantities- Binary digit-Logic Level-Basic logic operators- Basic logic functions. Basic digital circuits:- Inverter- AND and OR gates- NAND and NOR gates- Excusive OR and Exclusive NAND gate- Boolean Algebra – operations and expressions- Laws and rules of Boolean Algebra- Demorgan's theorms- Simplifications using Boolean expressions and truth tables-Karnaugh map- SOP and POS minimizations- Simplification of Boolean expression using K-Map (up to four variables) – (8 Hours)

UNIT II

Measures of Central Tendency: Mean, Median, Mode. Measures of Dispersion: Range, Quartile Deviation, Mean Deviation, Variance, Standard Deviation, Coefficient of Variation.

Matrices and determinants:-matrix, types of matrices, operations on matrices, Determinants-properties of determinants-inverse of a matrix- Rank of a Matrix, Trace of a Matrix. Solving Linear Equations using Matrices.(07 Hours)

UNIT III

Errors and Approximations- Nonlinear equations–Bisection Method, Regula-Falsi Method, Secant Method, Newton-Raphson method. Eigen values and eigenvectors:- Power Method, Jacobi Method, Householder's Method. System of Linear equations:- Cramer's Rule, Gauss Elimination Method, Gauss-Jordan Method.(07 Hours)

UNIT IV

Numerical Differentiation: Based on equal-interval Interpolation, Derivatives using Newton's backward difference formula. Numerical Integration:-Trapezium rule, Trapezoidal rule, Simpson's rule. Differential equations: Preliminaries, Taylor series method, RungeKutta methods-Statistical description and modelling of data. (08 Hours)

Text Books:

1. Thomas L Flyod-Digital Fundamentals, Pearson International Edition (9th Edition), Prentice Hall. (I and II Units)
2. Balachandra Rao, C K Shantha – "Numerical Methods – with Programs in BASIC, FORTRAN, Pascal and C++". University Press (Unit V)
3. Babu Ram –"Numerical Methods", Pearson (Unit V)
4. M.K. Jain, S.R.K. Iyengar, R.K. Jain – Numerical Methods (Problems and Solutions), New Age International Publishers (Unit V)

CORE COURSES

MCAIT01C01 DIGITAL FUNDAMENTALS AND COMPUTER ORGANIZATION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

- **CO1:** Introduce the basic Digital Principles and applications
- **CO2:** Familiarize with basic building blocks of Digital systems, Digital Logic and Digital Circuits
- **CO3:** Understand the organization and design of basic digital computer
- **CO4:** Understand the organization of memory and techniques that computers use to communicate with I/O devices

UNIT I

Number systems and arithmetic operations, Different Binary codes, Gates, Boolean algebra & Laws, Combinational Circuits: Sum of product, Product of sum, simplification by Boolean methods-K-Map Simplification- up to six variables. Tabular method. Decoders, Multiplexer, De-multiplexer, Encoder, Binary Adders, Subtractors, Magnitude comparator. Sequential circuits: Flip-flops, Analysis of Clocked Sequential Circuits, State Reduction and assignments, FF excitation tables, Design procedure Registers: shift registers, SISO, SIPO,.PISO, PIPO, Universal Shift Registers, Ripple Counters, Synchronous counters, Ring counter, Shift Counter, Up-down counters.

UNIT II

Historical Background of microprocessors – Architecture of 8086 - Addressing modes - Instruction set - Assembly Language Programming with MASM. Basic computer structure: Basic operational concepts. Number representation and arithmetic operations-Character representations -Performance. Instruction set Architecture: Memory locations and addresses, memory operations, instructions and instruction sequencing, addressing modes. Assembly language, stacks, subroutines, RISC v/s CISC.

UNIT III

Basic Processing Unit: Fundamental concepts, Instruction execution, Hardware components, Instruction fetch and execution steps, control signals, Hardwired control, CICS style processors (3-bus organization, micro programmed control). Arithmetic - multiplication of unsigned numbers (array and sequential multipliers), multiplication of signed numbers (Booth algorithm), Fast multiplication (bit pair recoding), Floating point numbers and operations.

UNIT IV

Memory system : Basic concepts, Semiconductor RAMS, ROMs, DMA, Memory hierarchy, Cache memory, performance requirements, virtual memory, memory management requirements, secondary storage devices- Basic I/O: Accessing I/O devices (device interface, program controlled I/O), Interrupts (enabling and disabling, handling multiple interrupts, controlling I/O device behavior, Processor control registers,

exceptions). I/O organization: Bus structure, bus operation, arbitration, Interface circuits, interconnection standards (USB, PCI, Firewire, SCSI, SATA)- Introduction to pipeling and parallel processing architecture.

Text Book

1. Digital Fundamentals - T.L.Floyd and R.P.Jain
2. Microcomputer Systems: The 8086/8088 Family, Architecture, Programming, and Design – YuCheng Liu and Glenn A. Gibson, Prentice Hall, Inc.
3. Hamacher, Vranesic, Zaky, Manjikian, Computer Organization and Embedded Systems, 6thedn, Tata McGraw Hill.

References:

1. William Stallings, Computer Organization & Architecture – Designing for Performance, 9th Edn, Pearson
2. John P. Hayes, Computer Architecture and Organization, Third Edn, Tata McGraw Hill.
3. M. Morris Mano, Computer System Architecture, PHI 2003
4. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications, Walter A Triebel and Avtar Singh
5. The 8051 Micro Controller, Kenneth J Ayala.
6. Microprocessor and Interfacing: Programming and Hardware. - D.V Hall

MCAIT01C02 SYSTEM SOFTWARE AND OPERATING SYSTEMS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Attain the knowledge about System Software
CO1: Understand the basic concepts, structure and functions of operating systems.
CO2: Understand the principles behind the techniques in resource management
CO3: Knowledge about the basic design of the OS

UNIT I

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, Design of Macro pre-processor. Linkers: Linking and Relocation concepts, Design of linkers, Self relocating programs, Linking for over-lays, Loaders: introduction to loaders - functions of loaders- Compilers: Introduction to compilers -Different Phases-Lexical Analysis- role of the lexical analyzer, input buffering, specification of tokens, Recognition of tokens, lexical Analyzer generators, Lex.

UNIT II

Introduction to Operating systems: Different types of Operating system, Overview of Operating systems- Operating system structures -Process management: Processes, Process Scheduling – Inter Process communication - Communication in client server systems, Threads: Processes Vs Threads, Types of threads, Multicore and Multithreading. CPU Scheduling: Scheduling algorithms, Multiple Processor Scheduling, Algorithm Evaluation- Advanced CPU scheduling. Process synchronisation: – Critical section Problem, Mutual Exclusion, Requirements, Semaphores, Monitors, Producer Consumer Problem, Readers Writers Problem, Deadlock Prevention, Detection and Recovery

UNIT III

Memory Management- swapping, Contiguous memory allocation, Paging Segmentation, Segmentation with paging. Virtual memory- Demand paging, processes creation, page replacement, allocation of frames, thrashing. File system interface and Implementation- File concepts, access methods, directory structure, File system mounting, File sharing, Protection, File system structure, File system implementation, Directory

implementation, Allocation methods, Free space managements, Efficiency and performance, Recovery, Log-structured file system.

UNIT IV

I / O Systems - I / O hardware, Application I/O interface, Kernel I / O subsystem, Transforming I / O to hardware operations, STREAMS, Performances. Mass storage structure - Disk structure, Disk scheduling, Disk management, Swap space managements, RAID structure, Disk attachments, Stable storage implementation, Tertiary storage structure.

Text books:

1. D.M. Dhamdhere, "Systems Programming and Operating Systems", TMH, 2003.
2. Silberschatz, A., Galvin, P.B. & Gagne, G. “Operating System Concepts”, 9th Ed. John Wiley & Sons-India.

References:

1. Dhamdhere, D. M. “Operating Systems”, 2nd Ed. The McGraw - Hill Companies.
2. Ditel, Deital and Choffness, Operating Systems, Pearson, 3rdEdn
3. William Stallings, Operating Systems, Internals and Design Principles, 7th Edition Pearson,
4. SibsankarHaldar ,Alex a Aravind, “Operating Systems”, Pearson Education India, Second impression.
5. Andrew S.Tanenbaum , Albert S.Woodhull, “The Minix Book- Operating Systems Design and Implementation”, 3rd Edition Pearson(2016).

MCAIT01C03 PYTHON PROGRAMMING

Contact Hours/ week: 4 **Credit: 2**

COURSE OUTCOME

CO1: Learn Python for expressing computation
CO2: Familiarize with functions and modules in python
CO3: Understand object-oriented programming concepts in Python
CO4: Learn the GUI programming in Python

UNIT I

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, Dynamic and Strongly typed language. 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, default parameters, Keyword arguments, *args and **kwargs**,** Command Line Arguments, Strings- Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

UNIT II

Lists- Creating List, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions used on lists, list Methods, The *del* statement, Dictionaries- Creating Dictionary, Accessing and Modifying *key:value* Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods, The *del* statement. Tuples and Sets-Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Tuple Methods, Using *zip()* Function, Sets, Set Methods, Frozenset. Files- Types of Files, Creating and Reading Text Data, File Methods to

Read and Write Data, Reading and Writing Binary files, The Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules, Regular Expression Operations- Using Special Characters, Regular expression Methods, Named Groups in Python Regular Expressions, Regular expression with *glob* Module.

UNIT III

Object-Oriented Programming- Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attribute versus Data Attribute, Encapsulation, Inheritance, Polymorphism.
Exceptions: Errors in python program-Compile time errors, Runtime errors, logical errors- Exception handling-types of exception- The except block- assert statement- User-defined Exceptions- Logging the exceptions.

UNIT IV

GUIs in Python: Root Window-Fonts and colours – Working with containers-Canvas, Frames, Widgets, Button widgets, Arranging widgets in the Frame, Label widget, Message Widget, Text widget, scrollbar widget, Check button widget, Radio button widget, Entry widget, Spinbox widget, List box widget, Menu widget- Table creation.

Text Book-

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

Reference Books

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

MCAIT01C04 COMPUTER NETWORK AND LINUX ADMINISTRATION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: *Understand the basics of Computer Network*
CO2: *Familiarize with OSI reference model*
CO3: To study basic Linux commands and understand the file system structure
CO4: To learn different system services, maintenance and configuring
CO5: *Understand the concepts of* Inter Process Communication
CO6: *Comprehend the concepts of* Network Configuration

UNIT I

Introduction, Basic concepts- Line configuration, Topology, Transmission mode, Categories of networks, Internetworks, Transmission media - Twisted pair Cable, Coaxial Cable, Optical Fiber, Satellite Communication, Cellular Telephony, Terrestrial Microwave, OSI and TCP/IP models, Functions of Physical Layer, Data link layer, Network layer, Transport Layer, Session Layer, Presentation Layer and Application Layer.

UNIT II

Introduction: Important parts of kernel; Major services in a UNIX system: init, login from terminals, syslog, periodic command execution crown and at; Boot process: The LILO boot process: LILO parameters, /etc/lilo.conf; The GRUB boot process; The /boot directory and files; initrd file and mkinitrd; Run levels: /etc/inittab, start-up script /etc/rc.d/rc.sysinit; System Configuration: The /etc/sysconfig/… files,kernel modules; kernel daemon; /etc/conf. modules and module parameters; /lib/modules/… directory structure and contents. File system configuration: file system types, /etc/fstabloyout and meaning; Basic user environment: /etc/skel/… and home directories, Window manager configuration file locations; System Security: Host security: tcp_wrappers and /etc/hosts.allow and /etc/hosts. deny, /etc/security, shadow password, ile permissions, users groups and umask; Adding and deleting users; System maintenance: Syslogd, klogd and /etc/syslog.conf; Using a remote syslog; The system crontab, dailyscript, tmpwatch and logrotate; Using and managing the system log files; Basic system backup and restore operations; Emergency rescue operations.

UNIT III

Inter Process Communication programming : Create a process- fork() system call, Parent and Child Process, Process ID, User and Group ID Half Duplex Unix Pipes, Named Pipes, (First In First Out), Streams and messages, System V IPC :Message Queues, Semaphores, Shared memory, Sample programs for IPC that uses Pipes, FIFO; Socket Programming: Overview, socket address, Elementary Socket System Calls: socket, socket pair, bind, connect, listen, accept, send, sendto, recv, recvfrom, close, Byte ordering routines, Byte Operations, Address conversion routines, Simple client Programs that uses some reserved ports, Simple Client / Server Program using unreserved ports.

UNIT IV

TCP / IP Network Configuration: Introduction to TCP / IP network, Protocols, IP address, Hostname, Configuring a Host : setting the host name, assigning IP address, broad cast, net mask and name server address, Editing Host and network files, Interface Configuration: loop back interface, Ethernet interface, The SLIP and PPP interface, Configuring Gateway, Routing through gateway, Network commands: ifconfig, netstat, route. Network applications Configuration: File Transfer Protocol (FTP) and Trivial File Transfer Protocol (TFTP), Network File Systems (NFS), Network Information System(NIS),Hyper Text Transfer Protocol (HTTP) and Web server, Server Message Block (SMB) Protocol and Samba server, Dynamic Host configuration Protocol (DHCP) Firewalls, Remote booting. Domain Name Services (DNS) and Mail services: working of DNS, Host name Resolution Name lookup with DNS, Reverse Lookup, Domain Name Servers and Zones, DNS database: SOA, NS, MX, A and PTR records, Secondary and primary DNS, Zone change notification, root servers, internet root domains, configuring DNS, Using nslookup. Simple Mail Transfer Protocol (SMTP), Post office Protocol(POP) Multipurpose Internet Mail Extension (MIME), SMTP and POP3 command, Mail routing, Configuring A mail server.

References:

1. Data Communications and networking, Fourth Edition by Behrouz A. Forouzan, McGraw Hill 2017.
2. Computer Networks, Fifth Edition by Andrew S. Tanenbaum, Prentice-Hall 2011
3. Data and computer communication, Tenth Edition by William Stallings, Prentice-Hall 2014
4. Evi Nemeth , et al, Linux Administration Hand Book , PHI 2018
5. Nicholas Wells, Linux Installation and Administration, Thomson Vikas 2003.
6. Olaf Kirch& Terry Dawson, Linux Network Administrators Guide, O'relly, 2003
7. Hunt, Linux DNS server Administration, BPB Publication, 2003
8. W Richard Stevens, Unix Network Programming, PHI, 2002
9. Linux AdministrationA Beginner's Guide, Wale Soyinka , 7th Edition.

MCAIT01C05 DISCRETE STRUCTURES AND OPTIMIZATION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Accomplish knowledge in the fundamental mathematical concepts and terminology for Computer Science **.**
CO2: Acquire knowledge in Mathematical Logic.
CO3: Gain knowledge in Boolean algebra.
CO4: Acquire knowledge in set Theory, Relations and functions.
CO5: Awareness about basics of mathematical induction and counting principles.
CO6: Awareness of fundamentals of probability
CO7: Gain knowledge in Graph Theory and Group Theory
CO8: Awareness about the importance of Optimization in Computer Science

Unit I

Mathematical logic: Propositional and Predicate Logic, Propositional Equivalences, Normal Forms, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference. Boolean Algebra: Boolean Functions and its Representation, Simplifications of Boolean Functions. Set and relations: Set Operations, Representation and Properties of Relations

Unit II

Functions and Relations: Functions – Types of Functions, Composition of Functions and Inverse Functions. Relations - Relations and Their Properties, Functions as relations, Closure of Relations, Composition of relations, Equivalence Relations and Partitions. Partial Ordering, Hasse Diagram. Mathematical Induction. Recurrence Relation, Generating function. Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion- Exclusion Principle. Probability: Continuous and Discrete Probability: Bayes' Theorem.

Unit III

Group Theory: Groups, Subgroups, Semi Groups, Product and Quotients of Algebraic Structures, Isomorphism, Homomorphism, Automorphism, Rings, Integral Domains, Fields, Applications of Group Theory. Graph Theory- Basic concepts- Introduction, Directed Graph, Undirected Graph, Connected and Disconnected Graphs, Bipartite Graph, Complete Bipartite Graph, Isomorphic Graphs, Subgraph. Paths and Circuits. Minimum Spanning Trees. Shortest Paths in Weighted Graphs- Dijkstra's Algorithm. All pair shortest path- Floyd Warshall algorithm. Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Storage representation and manipulation of graphs.

Unit IV

Optimization: Linear Programming - Mathematical Model, Graphical Solution, Simplex and Dual Simplex Method, Sensitive Analysis; Integer Programming, Transportation and Assignment Models, PERT-CPM: Diagram Representation, Critical Path Calculations, Resource Leveling, Cost Consideration in Project Scheduling.

Reference Books:

1. Discreate Mathematics and Its Applications with Combinatorics and Graph Theory, Kamala Krithivasan, McGraw Hill Education, 2011 (Seventh Edition).
2. Discrete Mathematics and Applications, Kenneth H. Rosen, TMH 2003 seventh edition
3. Discrete Mathematical Structures with Applications to Computer Science, J.P.Tremblay and R Manohar, TMH 2001
4. Discreate Mathematics, N Ch S N Iyengar, V M Chandrasekharan, KA Venkatesh, PS Arunachalam, Vikas Publishing, 2003.
5. Discrete Mathematics with Applications. Susanna S.Epp
6. Linear programming LR Potty

MCAIT02C07 ALGORITHM AND DATA STRUCTURE

Contact Hours/ week: 4 **Credit: 3**

COURSE OUTCOME

CO1: Knowledge about important computational problems.
CO2: Attain knowledge to design the algorithm.
CO3: Knowledge to analyze a given algorithm.
CO4: Acquire knowledge to analyze algorithm control structures and solving recurrences.
CO5: Understand the concept of data structures and its relevance in computer science.
CO6: Familiarize with selected linear and nonlinear data structures.

Unit 1

Introduction to algorithm design: Definition, characteristics, Steps in developing algorithm, methods of specifying an algorithm, important problem types: Combinatorial problems, Geometric problems, Graph problems, Numerical problems, Searching, Sorting and String processing. Basic technique for design of efficient algorithm: Brute Force approach, Divide-and-Conquer approach, Branch-and-Bound technique. Greedy approach, Dynamic programming, Backtracking.

Unit II

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. Analyzing algorithm control structures, Solving recurrences: Iteration method, Substitution method, Recursion Tree method, Master's Theorem, problem solving using Master's Theorem case 1, case 2 and case 3. Case study: Analysis of Strassen's algorithm for matrix multiplication, Analysis of Merge sort. Complexity Classes: P, NP, NP Hard and NP Complete problems.

Unit III

Data structures: Definition and classification. Linear data structure: Array- operations, polynomial representation with arrays; concept of recursion, types of recursion. Stack: operations on stack. Case study with Tower of Hanoi problem. Application of stack i. postfix expression evaluation. ii. conversion of infix to postfix expression. Queues: operation on queue. Circular queue, dequeue and priority queue. Application of queue: job scheduling. Linked list: single linked list, structure and implementation; operations – traversing, add new node, delete node, reverse a list, search and merge two singly linked lists. Circular linked list– advantage. Queue as circular linked list. Doubly linked list, operations – add/delete nodes, advantages.

Unit IV

Non- linear data structure: Tree- basic terminologies and properties; representation of binary tree, operations on binary tree; type of binary tree, forest, B tree, B+ tree and Trie. Tree traversal: in order, pre order and post order traversals. Binary search tree. Application of tree, AVL tree, Huffman algorithm. Representations and operations of sets: Hash table, linked lists, tree and bit vector. Sorting Techniques: Insertion sort, Bubble sort, Selection sort, Quick sort and Merge sort. Comparison of sorting algorithms. Searching: basic terminologies, linear search: linear search with array, linear search linked lists. Non- linear search techniques, binary search, binary tree searching. Graphs: basic terminologies, representation of graph, matrix representation of graphs. Graph traversals - depth-first traversal – breadth-first traversal - applications of graphs – shortest-path algorithm – Dijkstra's algorithm -minimum spanning tree – Prim's and Kruskal's algorithms.

REFERENCE BOOKS:

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, Design and Analysis of Computer Algorithms, Addison Wesley.
3. Pallaw, V K, Design and Analysis of Algorithms, Asian Books Private Ltd, 2012.
4. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013
5. Oded Goldreich, P,NP and NP- Completeness, Cambridge University Press, 2011.
6. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume- 1
7. Samanta, Classic Data structures, Second Edition, PHI
8. Sahni and Mehta, Fundamentals of Data Structures in C++, 2nd Edn, University Press
9. Sahni, Rajasekaran, Fundamentals of Algorithms, 2ndEdn, University Press
10. M. A. Weis, Data Structures and Algorithm Analysis in C++, Pearson Edu. Asia, 2013
11. Anany Leviton, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson Education.

MCAIT02CO8 PROGRAMMING IN JAVA

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO 1: Understand the Enterprise Java platform
CO2: Learn APIs and runtime environment for developing and running large – scale projects.
CO3: Develops programming skills in multi – tiered, scalable, reliable and secure network application.
CO4: Understand the structure and development of a web application.

UNIT I

Object oriented Progrmming concepts ,Features of Java: - Bytecode, Java Virtual Machine (JVM), Java Applets and Applications, Java file name and directory structure; Packages of Java API. Data Types, Variables, Arrays, Type Conversion and Casting; Operators; Control Statements. Class:- Class Fundamentals, Declaring Objects, Constructors, access specifier, static, Nested and Inner Classes, Command-Line Arguments, this Keyword; Garbage Collection. String handling. Collection class. Inheritance, method overloading, Method Overriding, Dynamic Method Dispatch, Abstract Classes

UNIT II

Packages:- Importing Packages; Interface: Defining an Interface, Implementing Interfaces; Exception Handling: try, catch, throw, throws, and finally, Java's Built-in Exceptions; Thread:- Synchronization, Messaging, Runnable interface, Inter thread communication, Deadlock, Suspending, Resuming and stopping threads, Multithreading. I/O streams, File streams: I/O Streams, File Input Stream and File Output Stream, Data Input and O/P Streams, Buffered I/P and O/P Streams, FileClass, Reader and Writer Streams, Random Access File. Applets: Applet lifecycle, working with Applets, The HTML APPLET tag. Working with Graphics. Abstract Window Toolkit (AWT): AWT Classes, Window Fundamentals, Component, Container, Panel, Window, Frame. working with Frame Windows, AWT Controls, Layout Managers, and Menus. Event Handling: Events, Event Sources, Event Classes, Event Listener Interfaces, Adapter Classes.

UNIT III

Database connectivity:- JDBC architecture- drivers- database connections- statements- resultsets- transactions-metadata-stored procedures-error handling- blobs and clobs. JDBC with Mysql . Database application: Basic steps to create database application, Remote database. RMI-Database example. ,Database Transaction. JNDI.

UNIT IV

Java Servlets: Generic and http servlets, Get, post, head and other requests; Servlet responses; error handling; security; servlet chaining; cookies; session tracking; Working with Apache Tomcat Server,Steps to create a servlet in Tomcat, working of servlet, servlet in Netbeans. JavaServer Pages: JSP basics, JSP API, JSP in Netbeans, Scripting elements, Implicit Objects, Directive Elements, sharing data between JSPs, JSP actions, JSP application development: Generating dynamic content, Error Handling and Debugging.

Reference Book:

1. Herbert Schildt, The complete reference Java2 ,11thed, Released December 2018
   Publisher(s): McGraw-Hill ISBN: 9781260440249
2. David Flanagan, Java in a Nutshell A desktop quick Reference, 7 Edition,2018. OReilly&AssociatesInc
3. Rajkumar, Java programming, pearson, 2013
4. HarimohanPandey, Java Programming, Pearson, 2012
5. David Flanagan,Jim Parley, William Crawford & Kris Magnusson , Java Enterprise in a nutshell- A desktop Quick reference -O'REILLY, 2005
6. Stephen Ausbury and Scott R. Weiner, Developing Java Enterprise Applications, Wiley-2001
7. JasonHunder& William Crawford, Java Servlet Programming, O'REILLY, 2002
8. Database Programming with JDBC and Java, Reese George, OReilly

MCAIT02C09 DATABASE MANAGEMENT SYSTEMS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the concept of Database Management Systems
CO2: Familiarization with Relational Database design
CO3: Skill in writing queries using SQL
CO4: Understand the concepts of NoSql databases such as Key-value store, column store, document store

UNIT I

Introduction to Database Management Concepts: Database Systems versus File Systems. Three level Architecture of databases. Overview of relational, network, hierarchical data models. The ER and EER. Relational Algebra and Relational calculus**.** Relational database design - Functional dependencies -1st, 2nd, 3rd, 4th, BCNF, 5th Normal form.

UNIT II

SQL: Built-in functions – numeric, date, string functions, set operations, sub-queries, correlated sub-queries, join, Exist, Any, All, joined relations. Integrity and security: domain constraints, referential integrity, assertion, triggers, authorization in SQL. PostgreSQL programming – PL/pgSQL. Case study: PostgreSql. Views: Introduction to views, data independence, security, updates on views, comparison between tables and views.

UNIT III

Transaction processing- desirable properties of transaction. Transactions and Schedules –Characterising Schedules based on Recoverability, Serializability of schedules. Concurrency Control in databases: Locking Techniques-time stamp ordering, multi version concurrency Control –granularity of data items.

UNIT IV

Basics of NoSql Database: BASE transactions and eventual consistency- Properties of NOSQL databases. Key-Value data-stores – Column Stores, Document data-stores - Architecture of Dynamo DB, Big Table, HBase, Cassandra and Mongo DB, Graph databases- Comparison of Twitter's FlockDB and Neo4j.

References

1. Silbersehatz, Korth and Sudarshan, Database system concepts, 6th edition MGH 2011
2. Ramakrishnan and Gehrke, Database Management Systems, 3rd Edn, Mc Graw Hill, 2003
3. Elmasri and Navathe, Fundementals of Database systems, 5th Edition, Pearson 2009
4. C.J.Date-A.Kannan, S.Swamynathan "An introduction to Database System" 8th Edition, Pearson education O'Reilly, Practical PostgreSQL Shroff Publishers(SPD) 2002.
5. Carlo Zaniolo,Stefano Ceri, Christos Faloat ,V.S.Subrahmanian,Roberto Ziart "Advanced Database System", Morn Kaufmann publishers 2005
6. Ian Robinson, Jim Webber, Emil Eifrem, Graph databases- , O'Reilly, 2015.
7. Fowler, Adam. NoSQL for dummies. John Wiley & Sons, 2015.

MCAIT02C10 WEB TECHNOLOGY

Contact Hours/ week: 3 **Credit: 2**

COURSE OUTCOME

CO1: Enable students to program for the World Wide Web using HTML and JavaScript
CO2: Create static and dynamic web pages using PHP.
CO3: Familiar with MySQL
CO4: Impart basic knowledge in Content Management System and Progressive Web Apps

UNIT I

HTML5: New Elements -Structural Elements, New Form/Input Elements, New Attributes Canvas, Video and Audio, Web Storage, Geolocation. The JavaScript Language- Introduction to JavaScript in Perspective-Syntax-Variables and Data Types-Statements-Operators-Literals-Functions-Objects-Arrays-Built-in Objects. Host Objects: Browsers and the DOM-Introduction 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. Introduction to Bootstrap and responsive web design basics

UNIT II

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 building in error checking, MySQL functions, displaying queries in tables.

UNIT III

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 IV

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. Mash-ups: Introduction. Progressive Web Apps – Introduction, Features, Advantages. Hybrid application development – Basics, Discover the platforms and frameworks used for hybrid application development

Reference Textbooks:

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. Bob Breedlove, et al, Web Programming Unleashed, Sams Net Publishing, 1stEdn

4. Steven Holzner, PHP: The Complete Reference, McGraw Hill Professional, 2008
5. Steve Suehring, Tim Converse, Joyce Park, PHP6 and MY SQL Bible, John Wiley & Sons,2009
6. Pedro Teixeira, Instant Node.js Starter, Packt Publishing Ltd., 2013
7. Anthony T. HoldenerIII, Ajax: The Definitive Guide, O'Reilly Media, 2008
8. Nirav Mehta, Choosing an Open Source CMS: Beginner's Guide Packt Publishing Ltd, 2009
9. James Snell, Programming Web Services with SOAP, O'Reilly 2002
10. Jacob Lett , Bootstrap Reference Guide, Bootstrap Creative 2018
11. Maximilian Schwarzmüller, Progressive Web Apps (PWA) - The Complete Guide, Packt Publishing 2018
12. Mahesh Panhale, Beginning Hybrid Mobile Application Development, Apress 2016

MCAIT02C11 SOFTWARE ENGINEERING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Understand the basic principle of Software Engineering
CO2: Integrate knowledge to uphold software quality.
CO3: Familiarize with requirement engineering and classical software design techniques.
CO4: Familiarize with various software testing Strategies.
CO5: Familiarize with various cost estimation techniques

Unit I

Software and Software Engineering: The Nature of Software, the Unique Nature of WebApps, Software Engineering, The Software Process, Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, Agile Process Models –Extreme Programming (XP), Adaptive Software Development, Dynamic system Development Method(DSDM),scrum, crystal, Feature driven development(FDD)

Unit II

Understanding Requirements: Requirements Engineering, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Validating Requirements. Requirements Modelling: Requirements Analysis, Scenario-Based Modelling, UML Models That Supplement the Use Case, Data Modelling Concepts, Class – Based Modelling, Flow-Oriented Modelling, creating behavioural model.

Unit III

Component-Level Design: What is a Component? Designing Class- Based Components. User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface analysis, Interface design steps, Web apps interface design, Design evaluation. Software Configuration Management: The SCM Repository, the SCM Process.

Unit IV

Estimation and scheduling: Software Sizing, LOC and FP based Estimations; Estimating Cost and Effort; Estimation Models, Constructive Cost Model (COCOMO), Project Scheduling and Staffing; Time-line Charts.

Software Testing: Verification and Validation; Error, Fault, Bug and Failure; Unit and Integration Testing; White-box and Black-box Testing; Basis Path Testing, Control Structure Testing, Deriving Test Cases, Alpha and Beta Testing; Regression Testing, Performance Testing, Stress Testing.

Text Book

Software Engineering – Roger S Pressman, 'Software Engineering: A Practitioner's Approach, 7 th Edition, McGraw-Hill International Edition, 2010.

Reference Books

1. Richard Fairey, 'Software Engineering concepts, Tata McGraw-Hill 2009 reprint
2. Ian Sommerville, 'Software Engineering'. 6th Ed., Addison Wesley
3. Waman S Jawadekar, 'Software Engineering Principles and Practice', Tata McGraw Hill, 2004
4. PankajJalote., Software Engineering - A precise Approach, Wiley India, 2011
5. Ammann and Offcut, Introduction to Software Testing, Cambridge University Press, 2008

MCAIT03C12 MACHINE LEARNING TECHNIQUES

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: To introduce the prominent methods for machine learning
CO2: To study the basics of supervised and unsupervised learning
CO3: To study the basics of neural networks
CO4: To study the basics of deep learning architectures

UNIT I

Introduction to Machine Learning: Concept of 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. Setting up your machine learning platform; training, validation and testing, over-fitting and under-fitting, different types of error calculation.

UNIT II

Supervised Learning: Introduction, learning a class from example, learning multiple classes, model selection and generalization, linear regression and feature selection, Bayesian and Decision Tree learning; classification tree and regression tree, multivariate methods for learning; multivariate classification and regression. Unsupervised Learning: Introduction, clustering; mixture densities, k-means clustering, expectation maximization algorithm, Latent Dirichlet Allocation, spectral and hierarchical clustering.

UNIT III

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 IV

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 Network.

References:

1. Ethem Alpaydin, Introduction to Machine Learning- 3rd Edition, PHI.
2. Tom M. Mitchell, Machine Learning, McGraw-Hill
3. Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning), MIT Press, 2016.

4. Kuntal Ganguly, Learning Generative Adversarial Networks, Packt Publishing, 2017

MCAIT03C13 THEORY OF COMPUTATION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Acquire knowledge about the fundamental concepts Theoretical Computer Science
CO2: Understand the concept of Finite Automata, Nondeterministic Finite Automata and Pushdown Automata.
CO3: Advance knowledge about Regular Languages and Context Free Grammer
CO4: Attain knowledge about Turning Machine

Unit I

Introduction to the Theory of computation and Finite Automata: Mathematical preliminaries and notation, Proof techniques, 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.

Unit II

Regular Languages and Regular grammars: Regular expressions, connection between regular expressions and regular languages , regular grammars. Properties of Regular Languages: closure properties of regular languages, identifying non regular Language

Unit III

Context-free grammars & languagesContext-free grammars, parsing and ambiguity. Simplification of Context free Grammars and Normal forms : methods of transforming grammars two important normal forms. Pushdown automata for context-free languagesNon 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 and linear languages, closure properties for context-free languages.

Unit IV

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. Limits of Algorithmic computation: Problems that cannot be solved by Turing machines, Undecidable Problems for Recursively enumerable Languages, The Post Correspondence problem.

Text Book:

1. An introduction to Formal Languages and Automata, Peter Linz, 4thedn, Narosa publishing House.

Reference Books

1. John C Martin, Introduction to Languages and the Theory of Automata, McGraw Hill 1997
2. Mishra &Chandrasekharan, Theory of Computer Science : Automata, Languages andComputation, 3rd edn, PHI
3. Hopcroft, Motwani and Ullman, Introduction to automata theory, Languages and

Computation, 3rdEdn,., Pearson

MCAIT03C14 BIG DATA ANALYTICS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Acquire knowledge about the importance of Big Data.
CO2: Information about Stream Data Model in Big Data
CO3: Achieve the knowledge about the Big Data Analytics.
CO4: Make awareness about Hadoop Distributed File System

Unit I

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 II

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.

Unit III

The Hadoop Distributed File System – Components of Hadoop- Analyzing the Data with Hadoop- Scaling Out- Hadoop Streaming- Design of HDFS-Java interfaces to HDFSBasics-Developing a Map Reduce Application-How Map Reduce Works-Anatomy of a Map ReduceJob run-Failures-Job Scheduling-Shuffle and Sort – Task execution - Map Reduce Types andFormats- Map Reduce Features. Setting up a Hadoop Cluster - Cluster specification - Cluster Setup and Installation – Hadoop Configuration-Security in Hadoop - Administering Hadoop – HDFS - Monitoring-Maintenance- Hadoop benchmarks- Hadoopin the cloud.

Unit IV

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

Reference Books:

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, McGrawHill Pub, 2012
4. AnandRajaraman& Jeffrey D Ullman, Mining of Massive Datasets, Cambridge University Pres,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 .

ELECTIVES

POOL A – (ELECTIVE-I)

POOL A

MCAIT03E01 ARTIFICIAL INTELLIGENCE
MCAIT03E02 BIOINFORMATICS
MCAIT03E03 FUZZY SETS & SYSTEMS
MCAIT03E04 GRAPH THEORY & COMBINATORICS
MCAIT03E05 SOFTWARE ARCHITECTURE
MCAIT03E06 FOUNDATIONS OF NATURAL LANGUAGE PROCESSING

MCAIT03E01 ARTIFICIAL INTELLIGENCE

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Attain the knowledge about the importance of Artificial Intelligence.
CO2: Understand about AI techniques
CO3: Acquire the knowledge about Expert System
CO4: Familiarize the knowledge about Machine Learning

Unit I

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 II

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.

Unit III

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 IV

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.

Text book:

1. George F Luger, Artificial Intelligence – Structures and Strategies for Complex probel solving, 5thEdn, pearson.

Reference Books:

1. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill.
2. S. Russel and p. Norvig, Artificial intelligence – A Modern Approach, 3rdEdn, pearson D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990

MCAIT03E02 BIOINFORMATICS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

- **CO1:** Expose students to the popular genomic and proteomic databases and to impart knowledge in processing and analyzing genomic data.
- **CO2:** Introduce advanced topics in Bioinformatics.
- **CO3:** Acquire knowledge about Sequence alignment
- **Co4:** Familiar with the major Biological Databases.

Unit I

Bioinformatics - introduction to - nature and scope of computational biology and Bioinformatics. Cells - prokaryotes and eukaryotes - DNA double helix - central dogma – RNA, Amino acids, Proteins - string representations. A glossary of Bioinformatics terms - file format for bio-molecular sequences, sequence alignment, phylogeny, gene finding, microarray analysis, homology and evolutionary relationships.

Unit II

Basic algorithms in Computational Biology - exhaustive search methods and their applications in Computational Biology - string matching algorithms. Motif finding - tandem repeats – concept of dynamic programming - graph algorithms - clustering algorithms.

Unit III

Sequence alignment - pair-wise sequence alignment, Sequence similarity, identity, and homology. Global and local alignment, Dot plots for sequence comparison, Dynamic programming. Need of scoring schemes - penalizing gaps, scoring matrices for amino acid sequence alignment, PAM probability matrix and log odds

matrix, BLOSUM, Dot-plot visualization, Needleman-Wunsch algorithm- effect of scoring schemes – e values - BLAST and FASTA, Smith – Waterman algorithm for local alignment. Multiple sequence alignment - sequence alignment using dynamic programming, N-dimensional dynamic programming. Tools for MSA - muscle and T-Coffee. Phylogenetic algorithms - evaluation of phylogenetic trees, significance.

Unit IV

Introduction to the major resources - NCBI, EBI and ExPASy - nucleic acid sequence databases - GenBank, EMBL, DDBJ – Protein sequence databases - SWISS-PROT, TrEMBL, PIR_PSD - genome databases at NCBI, EBI, TIGR, SANGER – procedures to access these databases and to make use of the tools available.

Reference books

1. Mount D, Bioinformatics: Sequence & Genome Analysis, 2nd Edition, Cold spring Harbor Press, ISBN: 978-087969712.
2. Dan Gusfield, Algorithms on Strings Trees and Sequences, 1st Edition, Cambridge University Press, ISBN: 0521585198.
3. Pevzner P A, Computational Molecular Biology: An Algorithmic Approach, MIT Press, Cambridge, MA, ISBN: ISBN: 9780262161978.
4. Jeremy J. Ramsden, Bioinformatics: An Introduction, Springer, ISBN: 9789401570961.
5. Sushmita M and Tinku A, Data Mining: Multimedia, Soft Computing and Bioinformatics, Wiley-Interscience, ISBN: 9780471460541.

MCAIT03E03 FUZZY SETS & SYSTEMS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the concept of Fuzzy Logic
CO2: Recognize the difference between Classical set and Fuzzy set
CO3: Cognize the properties of Fuzzy membership functions
CO4: Appreciate the applications of Fuzzy Logic

Unit I

Introduction: Fuzzy systems – Historical perspective, Utility and limitations, uncertainty and information, fuzzy sets and membership, Chance vs. Fuzziness. Classical sets and Fuzzy sets: Classical set (Operations, properties, mapping to functions). Fuzzy sets (operations, properties, Alternative fuzzy set operations).

Unit II

Classical Relations and Fuzzy relations: Cartesian product, crisp relations (cardinality, operations, properties, composition), Fuzzy relations (cardinality, operations, properties, Fuzzy Cartesian products and composition), Tolerance and equivalence relation, Crisp equivalence and tolerance relations, Fuzzy tolerance and equivalence relations, value assignments (Cosine amplitude , Max-min method), other similarity methods, other forms of composition Operation.

Unit III

Properties of membership functions, Fuzzification and Defuzzification: Features of the -cuts for fuzzy λ membership functions, various forms, Fuzzification, defuzzification to crisp sets, relations, Defuzzification to scalars. Logic and Fuzzy systems: Classical logic, proof, Fuzzy logic, approximate reasoning, other forms of the implication operation. Natural language, Linguistic hedges, Fuzzy rule based systems, Graphical techniques for inference.

Unit IV

Development of membership functions: Membership value assignments (intuition, inference, rank ordering, Neural network, Genetic algorithm, inductive reasoning.) Extension Principle: Crisp functions, mapping and relations, Functions of Fuzzy sets – extension principle, Fuzzy transform, practical considerations. Fuzzy arithmetic: Interval analysis, Approximate methods of extension – DSW and restricted DSW algorithms. Fuzzy classification: Classification by equivalence relation (crisp and Fuzzy), Cluster analysis, cluster validity, C-means clustering (Hard and Fuzzy), Fuzzy c-means algorithm.

Reference books:

1. Ross, Fuzzy Logic with Engineering Applications, 3rd Edn, Wiley India.
2. Hajek P, Metamathematics of Fuzzy Logic. Kluwer, 1998
3. Rajasekharan and Viajayalakshmipai, Neural Networks, Fuzzy Logic and Genetic Algorithm, PHI, 2003. 4. Sivanandan and Deepa, Principles of Soft Computing, John wiley.

MCAIT03E04 GRAPH THEORY & COMBINATORICS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Accomplish knowledge about Graph Theory
CO2: Understand the applications of Graph Theory
CO3: Acquire knowledge about Combinatorics
CO4: Cognizance about Recurrence and nonhomogeneous recurrence relations

Unit I

Introduction to Graphs, definitions, sub graphs, paths and cycles, matrix representation of graphs, Euler tours, Chinese postman problem, planar graphs, Euler's formula, platonic bodies, applications of Kuratowski's theorem, Hamiltonian graphs, graph colouring and chromatic polynomials, map colouring.

Unit II

Trees: definition and properties, rooted trees, trees and sorting, weighted trees and prefix codes, biconnected components and articulation points. Kruskal's and Prim's algorithms for minimal spanning trees. Disjkstra's shortest path algorithm, Bellman – Ford algorithm, all-pairs shortest paths, Floyed – Warshall algorithms, the max-flow min-cut theorem, maximum bipartite matching.

Unit III

Fundamental principles of counting, permutations and combinations, binomial theorem, combinations with repetition, combinatorial numbers, Principle of inclusion, derangements, arrangements with forbidden positions.

Unit IV

Generating functions, partitions of integers, the exponential generating function, the summation operator. Recurrence relations, first order and second order, nonhomogeneous recurrence relations, method of generating functions.

Reference Books:

1. Grimaldi R.P., —Discrete and Combinatorial Mathematics : an applied Introduction||, 3e, Addison Wesley, 1994
2. Corman T. H., Leiserson C. E., Rivest R. L., —Introduction to algorithms:, Prentice Hall India, 1990

3. Mott J.L., Kandel A. and Baker T.P., —Discrete Mathematics for Computer Scientists and Mathematicians||, 2e, PHI
4. Rosen K.H., —Discrete Mathematics and its Applications||, 3e, McGraw Hill 5. Clark J. and Holton D. A., —A first look at Graph theory||, World Scientific.

MCAIT03E05 SOFTWARE ARCHITECTURE

Contact Hours/ week: 3 **Credit: 3**

CO1: Accomplish knowledge about Software Architecture
CO2: Cognizance about Service Oriented Architecture
CO3: Acquire knowledge about Interactive Systems
CO4: Integrate the software Engineering skills.

UNIT I

Software Architecture - Foundations - Software architecture in the context of the overall software life cycle – Key architectural Principles, Common Application Architecture, Design Principles, Architectural Styles - CASE study of Architectures Designing, Describing, and Using Software Architecture - IS2000: The Advanced Imaging Solution - Global Analysis – Factors affecting the architecture development of a software.

UNIT II

Conceptual Architecture View, Module Architecture View, Styles of the Module Viewtype - Execution Architecture View, Code Architecture - View. Component-and-Connector Viewtype - Styles of Component-and-Connector Viewtype - Allocation Viewtype and Styles – Documenting Software Interfaces, Documenting Behavior - Building the Documentation Package.

Unit III

Archetypes and Archetype Patterns. Model Driven Architecture with Archetype Patterns. Literate Modelling, Archetype Pattern. , Customer Relationship Management (CRM) Archetype Pattern, Product Archetype Pattern, Quantity Archetype Pattern, Rule Archetype Pattern. Design Patterns, Creational Patterns, Patterns for Organization of Work, Access Control Patterns.

UNIT IV

Service Oriented Architecture, Service Variation Patterns, Service Extension Patterns, Object Management Patterns Adaptation Patterns, Communication Patterns, Architectural Patterns, Structural Patterns, Patterns for Distribution. Patterns for Interactive Systems. Adaptable Systems, Frameworks and Patterns, Analysis Patterns, Patterns for Concurrent and Networked Objects, Patterns for Resource Management, Pattern Languages, Patterns for Distributed Computing.

REFERENCE BOOKS

1. Hofmeister, Nord, Soni, Applied Software Architecture, Addison-Wesley
2. Paul Clements et al., Documenting-software-architectures-views-and-beyond, 2ndedn, Pearson 3. Arlow&Neustadt, Enterprise Patterns And MDA-Building Better Software With Archetype Pattern An UML, Pearson, 2004
4. Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Pattern-Oriented Software Architecture, Vol 1 - A System Of Patterns, Wiley.

5. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns, Pearson

MCAIT03E06 FOUNDATIONS OF NATURAL LANGUAGE PROCESSING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: To introduce the fundamentals of Language processing from an algorithmic viewpoint.
CO2: To discuss various issues that make natural language processing a hard task.
CO3: To understand how the machines can deal with Natural Languages
CO4: To discuss some applications of Natural Language Processing (NLP).

UNIT I

Introduction to Language: Linguistic Knowledge, Grammar, Language and Thought, computational linguistics vs NLP, why NLP is hard, why NLP is useful, classical problems. Words of Language, Content Words and Function Words, Lexical categories, Regular expressions and automata. Morphology: Morphemes, Rules of Word Formation, Morphological parsing and Finite state transducers.

UNIT II

N-grams: simple N-grams, smoothing, Applications, language modelling.
Word classes and POS tagging: tag sets, techniques: rule based, stochastic and transformation based. Introduction to Natural Language Understanding- Levels of language analysis- Syntax, Semantics, Pragmatics.

UNIT III

Grammars and Parsing- Grammars for Natural Language: CFG, Probabilistic Context Free Grammar, statistical parsing. Features and Unification: Feature structures and Unification of feature structures. Lexical semantics, formal semantics and discourse. WSD, Information retrieval: Boolean, vector space and statistical models. Knowledge Representation and Reasoning- FOPC, Elements of FOPC.

UNIT IV

Discourse processing: monologue, dialogue, reference resolution, Conversational Agent. Text coherence. Dialogue acts: Interpretation of dialogue acts, plan inference model, clue-based model. Semantics: Representing meaning, Semantic analysis, Lexical semantics. Applications: Machine Translation, Natural Language Generation: architecture, surface realization and discourse planning.

Textbook:

1. Daniel Jurafsky and James H Martin. Speech and Language Processing
2. Hobson Lane, Cole Howard, Hannes Hapke. Natural Language Processing in action
3. Victoria fromkin, Robert Rodman and Nina Hyams, An Introduction to language, Tenth Edition. Downloadable freely at: https://ukhtt3nee.files.wordpress.com/2019/04/an_introduction_to_language.pdf

POOL B – (ELECTIVE-II)

POOL B

MCAIT04E07 DATA MINING AND WAREHOUSING
MCAIT04E08 ALGORITHMS IN COMPUTATIONAL BIOLOGY
MCAIT04E09 OBJECT ORIENTED ANALYSIS AND DESIGN
MCAIT04E10 COMPUTER VISION
MCAIT04E11 SOFTWARE PROJECT MANAGEMENT
MCAIT04E12 VISUAL CRYPTOGRAPHY

MCAIT04E07 DATA MINING AND WAREHOUSING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Acquire knowledge about the importance of Data Warehousing.
CO2: Understanding the importance of Data Mining
CO3: Attain knowledge about Clustering techniques
CO4: Understand the applications of Decision Trees

Unit I:

Introduction; data warehousing –Multidimensional data model, OLAP operations, warehouse schema, Data warehousing Architecture, warehouse server, Metadata, OLAP engine, data warehouse Backend Process.

Unit II:

Data mining – what is, KDD vs data mining, DBMS vs data mining, DM Techniques, issues and challenges, Applications. Association rules – What is, Methods, a priori algorithm, partition algorithm, Pincer- search algorithm, FP-tree growth algorithm, incremental and Border algorithms, Generalized Association rule.

Unit III:

Clustering techniques – Paradigms, Partitioning Algorithms, k – Medoid algorithms, CLARA, CLARANS, hierarchical clustering, DBSCAN, Categorical Clustering, STIRR.

Unit IV:

Decision trees – what is, tree construction principles, Best split, Splitting indices, Splitting criteria, decision tree construction algorithms, CART, ID3, C4.5, CHAID. Introduction to web, spatial and temporal data mining.

References:

1. Data Mining Techniques, A K Pujari, University press.

2. J. Han, M. Kamber, "Data Mining Concepts and Techniques", Harcourt India Pvt Ltd.

3. M. Dunham, " Data Mining : introductory and Advanced Topics", Pearson Pub.

MCAIT04E08 ALGORITHMS IN COMPUTATIONAL BIOLOGY

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the Algorithms in Computational Biology
CO2: Familiar with the application of string metric.
CO3: Advance knowledge about Sequence Alignment
CO4: Understand the significance of phylogenetic trees

UNIT I

Basic Algorithms in Computational Biology: Exhaustive search methods and their applications in Computational Biology- Motif finding- Tandem repeats.

UNIT II

String matching algorithms: pattern matching in strings, suffix, prefix, factor, substring, exact string matching, exact tandem repeat, Data matrices: Measure of similarity, binary data measures, count data measures, continuous data measures, proximity matrices, string matric, clustering algorithm.

UNIT III

Sequence Alignment: Pair-wise sequence alignment, Need of Scoring schemes- Penalizing gaps; Scoring matrices for amino acid, PAM Probability matrix and Log odds matrix; BLOSUM; Dot-plot visualization; Needleman-Wunsch algorithm- effect of scoring schemes- e- values; BLAST and FASTA, Smith – Waterman algorithm for local alignment.

UNIT IV

Multiple sequence alignment: n dimensional dynamic programming. Tools for MSA: Muscle and T-Coffee. Phylogenetic Algorithms: Clustering based methods- UPGMA and neighbor joining, Optimality based: Fitch-Margoliash and minimum evolution algorithm; Character based methods- Maximum Parsimony and Maximum Likelihood methods; Evaluation of phylogenetic trees- significance.

TEXT BOOK:

1. Dan Gusfiled, Algorithms on Strings Trees and Sequences, Cambridge University Press.
2. Pevzner P A, Computational Molecular Biology: An Algorithmic Approach, MIT Press Cambridge, MA, 2000.
3. John D MacCuish and Norah E. MacCuish, Clustering in Bioinformatics and Drug Discovery, CRC Press 2011.

REFERENCE:

1. Richard M. Karp, Mathematical challenges from genomics and molecular biology, Notices of the American Mathematical Society, vol. 49, no. 5, pp. 544-553
2. Mount D, Bioinformatics: Sequence & Genome Analysis, Cold spring Harbor press.

3. Jeremy J. Ramsden, Bioinformatics: An Introduction, Springer.
4. Glyn Moody, Digital Code of Life: How Bioinformatics is Revolutionizing Science, John Wiley & Sons Inc
5. Tao Jiang, Ying Xu and Michael Q. Zhang, Current Topics in Computational Molecular Biology, Ane Books.
1. Sushmita M and Tinku A, Data Mining Multimedia, soft computing and Bioinformatics, John Wiley & Sons, Inc., 2003
2. Andrzej K. Konopka and M. James C. Crabbe, Compact Handbook of Computational Biology, CRC Press.
8. Bellman R E, Dynamic Programming, Princeton University Press.
9. Needleman S B and Wunsch C D, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol., 48 (1970) 443–453.
10. Smith T F and Waterman M S, Identification of Common Molecular Subsequences, J. Mol. Bio. 147 (1981) 195–197.
11. Watson J D and Crick F H C, A Structure for Deoxyribose Nucleic Acid, Nature, 171 (1953) 737–738

MCAIT04E09 OBJECT ORIENTED ANALYSIS AND DESIGN

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Understand the basic principle of Object Oriented Systems
CO2: Become familiar with the Unified modelling Language and different diagrams in design
CO3: Acquire knowledge to create Architecture diagrams
CO4: Learn in detail classes and types
CO5: Understand the concepts of components
CO6: Learn Impact of object orientation on Testing

UNIT I

Overview of object-oriented systems, objects, attributes, encapsulation, class hierarchy, polymorphism, inheritance, messages, history of object orientation. Introduction to UML, basic expression of classes, attributes, and operations, Class diagrams: generalization and association constructs composition and aggregation. Use case diagrams, Object interaction diagrams: collaboration diagrams, sequence diagrams, asynchronous messages and concurrent execution. State diagrams: basic state diagrams, nested states, concurrent states and synchronization, transient states. Activity diagrams.

UNIT II

Architecture diagrams: packages, deployment diagrams for hardware artefacts and software constructs . Interface diagrams: window-layout and window-navigation diagrams.

UNIT III

Encapsulation structure, connascence, domains of object classes, encumbrance, class cohesion, state-spaces and behavior of classes and subclasses, class invariants, pre-conditions and post-conditions, class versus type, principle of type conformance, principle of closed behavior.

UNIT IV

Abuses of inheritance, danger of polymorphism, mix-in classes, rings of operations, components and objects, design of a component, light weight and heavy weight components, advantages and disadvantages of using components. Software Quality Assurance – Impact of object orientation on Testing – Develop Test Cases and Test Plans

Reference books

1. Page-Jones .M, Fundamentals of object-oriented design in UML, Addison Wesely

2. Booch. G, Rumbaugh J, and Jacobson. I, The Unified Modeling Language User Guide, Addison Wesely.
3.Bahrami.A, Object Oriented System Development, McGraw Hill.
4.Booch. G, Rumbaugh J, and Jacobson. I, The Unified Modeling Language Reference Manual, Addison Wesely.
5.Larman.C, Applying UML & Patterns: An Introduction to Object Oriented Analysis & Design, Addison Wesley
6.Pooley R & Stevens P, Using UML: Software Engineering with Objects & Components, Addison Wesley.
7. Fowler Martin, —UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third edition, Addison Wesley, 2003.
8. Erich Gamma, a n d Richard Helm, Ralph Johnson, John Vlissides, —Design patterns: Elements of Reusable Object-Oriented Software||, Addison-Wesley, 1995.

MCAIT04E10 COMPUTER VISION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Awareness about the importance of Computer Vision.
CO2: Familiar with object representation
CO3: Acquire the Knowledge about 3D vision
CO4: Gain information about the applications of motion analysis techniques.

Unit I

Introduction: Motivation, Difficulty, Image analysis tasks, Image representations, Image digitization, Image properties, Color images, Cameras. Data Structures: Levels of image data representation - Traditional image data structures - Hierarchical data structures. Texture: Statistical texture description, Syntactic texture description methods, Hybrid texture description methods, Texture recognition method applications.

Unit II

Object Recognition: Knowledge representation, Statistical pattern recognition, Neural nets, Syntactic pattern recognition, Recognition as graph matching, Optimization techniques in recognition, Fuzzy systems.

Unit III

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 - 2D view based 3D representation.

Unit IV

Motion Analysis: Differential motion analysis methods, Optical flow, Analysis based on interest points, Detection of specific motion patterns, Video Tracking, Motion models to aid tracking.

ReferenceBooks:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image Processing, Analysis and Machine Vision”, Cengage Learning, New Delhi, 2014.
2. Wesley E. Synder and Hairong Qi, ”Machine Vision”, Cambridge University Press, USA, 2010.
3. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer-Verlag, London, 2011.
4. Rafael C Gonzalez, Richard E Woods, Steven L Eddins, “Digital Image Processing”, Pearson Education, New Delhi, 2009.

MCAIT04E11 SOFTWARE PROJECT MANAGEMENT

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Understand the importance of Software Project Management
CO2: Advance knowledge about Project Planning
CO3: Understand the Dimensions of Project Monitoring
CO4: Appreciate the significance of Software Quality Assurance

UNIT I

Software Project and Characteristics, Project Constraints, Project Life Cycle and Process Life Cycle. Factors in Designing a Project Structure, Types of Project Organization Structures, Different Management Styles. Project Enabling Processes and Project Facilitating Processes. Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, Software Project Management activities, SPM Framework, Common problems with software projects.

UNIT II

Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Elements of a Project Plan.Steps to a Well-Defined Project Plan. Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Methods of representing WBS, Application of the WBS. Structure of a Software Project Management Plan. Software project estimation, Software Effort estimation techniques. Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques, Activity Planning, Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts. Project Schedule Management. Ways to Organize Personnel.

UNIT III

Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index(SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Deskchecks, Walkthroughs, Code Reviews, Pair Programming.

UNIT IV

Concept of Software Quality, Activities of Software: Quality Planning, Quality Assurance, Quality Control, Tools and techniques for Quality Control. Software Quality Attributes, Software Quality Indicators, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring

REFERENCE BOOKS:

1. Manish Kumar Jha, Software Project Management, DhanpatRai& Co
2. Bob Hughes, Mike Cotterell, Software Project Management, Rajib Mall : Tata McGraw Hill

MCAIT04E12 VISUAL CRYPTOGRAPHY

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Acquire knowledge about the real world applications of Visual Cryptography
CO2: Understand the principles of steganography and digital watermarking
CO3: Awareness about the real world applications of Visual Cryptography
CO4: Familiar with Visual Cryptography Schemes

UNIT I

Digital image Processing: Fundamentals:- Digital Image Representation-coordinate conversions, images as matrices, Image Types- intensity images, binary images, RGB images; Color Image Processing:-, Colour Image Representation- RGB model, CMY model, CMYK model, HSI model. Image file formats.

UNIT II

Principles of steganography and digital watermarking and their applications. Secret Sharing- Introduction, History of secret sharing, principle of secret splitting, phases of secret sharing, Access Structures, Threshold Schemes, Shamir's Scheme, Applications.

UNIT III

Visual Cryptography- Introduction- History of Visual Cryptography, Construction of Visual Cryptography Schemes, basis matrices, Construction of 2-out-of-2 Visual Cryptography Schemes, Construction of 2-out-of-2 Visual Cryptography Schemes with Square Pixel Expansion, Construction of Visual Cryptography Schemes with Consistent Image Size. Visual Cryptography Schemes- Construction of 2-out-of-n Visual Cryptography Schemes, Basis Matrices for 2-out-of-n Visual Cryptography Schemes, Construction of n-out-of-n Visual Cryptography Schemes, Basis Matrices for n-out-of-n Visual Cryptography Schemes, Construction of k-out-of-n Visual Cryptography Schemes, Basis Matrices for k-out-of-n Visual Cryptography Schemes.

UNIT IV

Colour Visual Cryptography – subpixel layout of colour visual cryptography, Variations of colour visual cryptography Schemes- Constructing a '2 out of 2' colour Visual Cryptography Schemes, Constructing a '2 out of n' colour Visual Cryptography Schemes, Applications of Visual Cryptography.

REFERENCE:

1. Borko Furht, Edin Muharemagic and Daniel Socek, Multimedia Encryption and Watermarking, Springer.
2. 2.Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson Education.
3. 3. Jen- Shyang Pan, Hsiang- Cheh Huang and Lakhi C. Jain, Intelligent Watermarking Techniques, World Scientific.
4. 4.Josef Pieprzyk, Thomas hardjino and Jennifer Sebberry, Fundamentals of computer security, Springer International Edition 2008.

POOL C (ELECTIVE- III)

POOL C

MCAIT04E13 PATTERN RECOGNITION
MCAIT04E14 CYBER FORENSICS
MCAIT04E15 NATURAL LANGUAGE PROCESSING WITH PYTHON
MCAIT04E16 GRID AND CLOUD COMPUTING
MCAIT04E17 INFORMATION SECURITY
MCAIT04E18 BIOMETRIC IMAGE PROCESSING

MCAIT04E13 PATTERN RECOGNITION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the importance of Pattern Recognition.
CO2: Familiarize with the Pattern Recognition Techniques
CO3: Gain knowledge about clustering algorithms.
CO4: Awareness about soft computing techniques for Pattern Recognition

Unit I

Pattern recognition systems- Definitions, data representation, representations of patterns and classes. Types of pattern recognition systems. Applications of pattern recognition systems. Bayesian decision making and Bayes Classifier for continuous and discrete features.

Unit II

Min-max and Neymann-Pearson classifiers, Discrminant functions, Decision surfaces. Maximum likelihood estimation and Bayesian parameter estimation. Overview of Nonparametric density estimation- Histogram based approach, classification using Parzen window. K-nearest neighbor estimation and classification. Classification of clustering algorithm- hierarchical clustering- agglomerative clustering. Partitional clustering- Forgy's algorithm. K-means clustering.

Unit III

Introduction to feature selection –filter method- sequential forward and backward selection algorithms. Wrappers method and embedded methods. Feature extraction methods- Principal component analysis, fisher linear discriminant analysis, ICA.

Unit IV

Neural network structures for Pattern Recognition – Neural network based Pattern associators – Unsupervised learning in neural Pattern Recognition – Self-organizing networks – Fuzzy logic – Fuzzy pattern classifiers – Pattern classification using Genetic Algorithms.

Reference Books:

1.Duda R.O., and Har P.E., Pattern Classification and Scene Analysis, Wiley, New York, 1973.
2. Bishop C.M, *Pattern recognition and machine learning,* Springer, 2nd Edition,2006
3. Theodoridis .S, Pikrakis .A, Koutroumbas .K, Cavouras .D, *Introduction to Pattern Recognition:AMatlab approach,* Academics Press 2010

MCAIT04E14 CYBER FORENSICS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the significance of Computer Forensics.
CO2: Accomplish the knowledge about Computer Forensics analysis and validation techniques
CO3: Familiarize with the Computer Forensic tools
CO4: Awareness about mobile device forensics.

Unit I

Computer Forensics Fundamentals: What is Computer Forensics?, Use of Computer Forensics in Law Enforcement, Computer Forensics Assistance to Human Resources/Employment Proceedings, Computer Forensics Services, Benefits of Professional Forensics Methodology, Steps taken by Computer Forensics Specialists. Types of Computer Forensics Technology: Types of Military Computer Forensic Technology, Types of Law Enforcement, Computer Forensic Technology - Types of Business Computer Forensic Technology.

Unit II

Evidence Collection and Data Seizure: Why Collect Evidence? Collection Options obstacles-- Types of Evidence - The Rules of Evidence-Volatile Evidence - General Procedure - Collection and Archiving - Methods of Collection -Artifacts - Collection Steps - Controlling Contamination: The Chain of Custody. Duplication and Preservation of Digital Evidence: Preserving the Digital Crime Scene – Computer Evidence Processing Steps - Legal Aspects of Collecting and Preserving Computer Forensic Evidence Computer Image Verification and Authentication: Special Needs of Evidential Authentication – Practical Consideration -Practical Implementation. Computer Forensics Evidence and Capture: Data Back-up and Recovery. The Role of Back-up in Data recovery. Recovering Graphics Files- Recognizing, locating and recovering graphic files, copy rights issues with graphics. Understanding data compression, identifying unknown file formats.

Unit III

Computer Forensics analysis and validation: Determining what data to collect and analyse, validating forensic data. Addressing data-hiding techniques, performing remote acquisitions. Network Forensics: Network forensics overview, performing live acquisitions, developing standard procedures for network forensics, using network tools, examining the honey net project. Processing Crime and Incident Scenes: Identifying digital evidence. Collecting evidence in private sector incident scenes, processing law enforcement crime scenes, preparing for a search, securing a computer incident or crime scene, seizing digital evidence at the scene, storing digital evidence, obtaining a digital hash, reviewing a case.

Unit IV

Current Computer Forensic tools: evaluating computer forensic tool needs, computer forensics software tools, computer forensics hardware tools, validating and testing forensics software E-Mail Investigations: Exploring the role of e-mail in investigation, exploring the roles of the client and server in email, investigating e-mail crimes and violations, understanding e-mail servers, using specialized e-mail forensic tools Cell phone and mobile device forensics: Understanding mobile device forensics, understanding acquisition procedures for cell phones arid mobile devices.

Reference Books:

1. Jhon R. Vacca, Computer Forensics, Computer Crime Investigation, Second Edition, Firewall Media, New Delhi, 2004
2. Bill Nelson, Amelia Phillips, Frank Enfinger, Christofer Steuart , "Computer Forensics and Investigations", Second Indian Reprint , Cengage Learning India Private Limited,2009
3. Britz, Computer Forensics and Cyber Crime – An Introduction, 2ndEdn, Pearson.

MCAIT04E15 NATURAL LANGUAGE PROCESSING WITH PYTHON

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: To understand the fundamental concepts of language processing with Python

CO2: Acquire knowledge in text categorization using Python

CO3: Gain knowledge and expertise in top NLP applications

CO4: To make the students aware about the information extraction from natural language text.

UNIT I

Language Processing and Python: Computing with Language: Texts and Words, Texts as Lists of Words, Simple Statistics, Making Decisions and Taking Control, Automatic Natural Language Understanding. Accessing Text Corpora and Lexical Resources: Accessing Text Corpora, Conditional Frequency Distributions, Reusing Code, Lexical Resources, WordNet.

UNIT II

Processing Raw Text: Accessing Text from the Web and from Disk, Strings: Text Processing at the Lowest Level, Text Processing with Unicode, Applications of Regular Expressions, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation. Formatting: From Lists to Strings-Writing Structured Programs: Basics, Sequences, Questions of Style, Functions: The Foundation of Structured Programming, Doing More with Functions, Program Development, Algorithm Design, A Sample of Python Libraries.

UNIT III

Categorizing and Tagging Words: Using a Tagger, Tagged Corpora, Mapping Words to Properties Using Python Dictionaries, Automatic Tagging, N-Gram Tagging, Transformation-Based Tagging, How to Determine the Category of a Word?

Learning to Classify Text: Supervised Classification, Further Examples of Supervised Classification, Evaluation, Decision Trees, Naive Bayes Classifiers, Maximum Entropy Classifiers, Modeling Linguistic Patterns.

UNIT IV

Extracting Information from Text: Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction, Sentiment Analysis. Analyzing Sentence Structure: Dependencies and Dependency Grammar, Grammar Development. Working with XML, Working with Toolbox Data. NLP applications: Sentiment Analysis, Text Summarization and Question answering

TEXTBOOK:

1. Steven Bird, EwanKlein, and Edward Loper, Natural Language Processing with Python O'Reilly Media, Inc.2009. Freely accessible at: https://www.nltk.org/book/
2. Dipanjan Sarkar , Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data, Bangalore, Karnataka India. 2016. Freely downloadable at: https://www.academia.edu/37026239/Text_Analytics_with_Python_A_Practical_Real-World_Approach_to_Gaining_Actionable_Insights_from_Your_Data_Dipanjan_Sarkar

MCAIT04E16 GRID AND CLOUD COMPUTING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: To understand the fundamental concepts of Distributed Computing, Grid Computing and its large scale scientific applications
CO2: Attain knowledge on the concept of virtualization that is fundamental to cloud computing
CO3: Learn Different Cloud deployment models and Cloud computing categories
CO4: Learn Implementations of Open source grid middleware packages and Hadoop Framework
CO5: Understand the security issues in the grid and the cloud environment.

Unit I

Evolution of Distributed computing: Scalable computing over the Internet – Technologies for network based systems – clusters of cooperative computers – Grid computing Infrastructures – cloud computing – service oriented architecture – Introduction to Grid Architecture and standards – Elements of Grid – Overview of Grid Architecture.Introduction to Open Grid Services Architecture (OGSA) – Motivation – Functionality Requirements – Practical & Detailed view of OGSA/OGSI – Data intensive grid service models – OGSA services.

Unit II

Cloud deployment models: public, private, hybrid, community – Categories of cloud computing: Everything as a service: Infrastructure, platform, software, Security as a Service – Pros and Cons of cloud computing – Implementation levels of virtualization – virtualization structure – virtualization of CPU, Memory and I/O devices, Desktop Virtualization – virtual clusters and Resource Management – Virtualization for data center automation. Tools and Products available for Virtualization.

Unit III

Open source grid middleware packages – Globus Toolkit (GT4) Architecture , Configuration – Usage of Globus – Main components and Programming model – Introduction to Hadoop Framework – Mapreduce, Input splitting, map and reduce functions, specifying input and output parameters, configuring and running a job – Design of Hadoop file system, HDFS concepts, command line and java interface, dataflow of File read & File write.

Unit IV

Trust models for Grid security environment – Authentication and Authorization methods – Grid security infrastructure – Cloud Infrastructure security: network, host and application level – aspects of data security, provider data and its security, Identity and access management architecture, IAM practices in the cloud, SaaS, PaaS, IaaS availability in the cloud, Key privacy issues in the cloud.

References Books:

1. Kai Hwang, Geoffery C. Fox and Jack J. Dongarra, "Distributed and Cloud Computing: Clusters, Grids, Clouds and the Future of Internet", First Edition, Morgan Kaufman Publisher, an Imprint of Elsevier, 2012.
2. Jason Venner, "Pro Hadoop- Build Scalable, Distributed Applications in the Cloud", A Press, 2009
3. Tom White, "HadoopThe Definitive Guide", First Edition. O"Reilly, 2009.
4. Bart Jacob (Editor), "Introduction to Grid Computing", IBM Red Books, Vervante, 2005
5. Ian Foster, Carl Kesselman, "The Grid: Blueprint for a New Computing Infrastructure", 2nd Edition, Morgan Kaufmann.
6. Frederic Magoules and Jie Pan, "Introduction to Grid Computing" CRC Press, 2009.
7. Daniel Minoli, "A Networking Approach to Grid Computing", John Wiley Publication, 2005.
8. Barry Wilkinson, "Grid Computing: Techniques and Applications", Chapman and Hall, CRC, Taylor and Francis Group, 2010.
9. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, "Virtualization, Business Models, Mobile, Security and more, Jones & Bartlett Learning Company, 2013
10. R. BUYYA, C. VECCHIOLA, S T. SELVI, Matering Cloud Computing, Mc Graw Hill (India) Pvt Ltd., 2013

MCAIT04E17 INFORMATION SECURITY

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Accomplish Knowledge about computer security.
CO2: Acquire skill to encrypt and decrypt data.
CO3: Study to different cryptographic technique.
CO4: Obtain knowledge to Message Authentication
CO5: Accomplish knowledge about Digital Signature

Unit I

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 II

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. Pseudo Random Number generators - design of stream cipher, RC4.

Unit III

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

Reference books

1. William Stallings, Cryptography and Network Security, Pearson 2004
2. Foorouzan and Mukhopadhyay, Cryptographiy and Network security, 2ndedn
3. BuceSchneier., Applied cryptography – protocols and algorithms, Springer Verlag 2003
4. William stallings, Network Security Essentials, , 4thedn, Pearson
5. Pfleeger and Pfleeger, Security in Computing, 4thEdn, Pearson

MCAIT04E18 BIOMETRIC IMAGE PROCESSING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME:

CO1: Acquire knowledge about Biometric Image Processing
CO2: Familiar with Image enhancement
CO3: Gain knowledge about Hand and Iris Biometrics
CO4: Familiar with Morphological image processing.

Unit I

Digital image representation, Fundamental steps in image processing, Elements of digital image processing system, Image sensing and acquisition, Sampling and quantization, Basic relationship between pixels, Transformation technology: Fourier transform - Discrete cosine transform.

Unit II

Image enhancement: Spatial domain methods: Basic grey level transformations - Histogram equalization - Smoothing spatial filter - Sharpening spatial filters - Laplacian, Frequency domain methods: Smoothing and sharpening filters – Ideal - Butterworth - Gaussian filters. Image Segmentation: Point- Line and edge detection - Thresholding - Global and multiple thresholding, Region splitting and merging.

Unit III

Morphological image processing: Fundamental concepts and operations, Dilation and Erosion, Compound operations, Morphological filtering, Basic morphological algorithms, Grayscale morphology. 2D and 3D face biometrics: Global face recognition techniques: Principal component analysis - Face recognition using PCA - Linear discriminant analysis - Face recognition using LDA, Local face recognition techniques: Geometric techniques - Elastic graph matching techniques, Hybrid face recognition techniques. 3D Face Image: Acquisition, Pre-processing and normalization, 3D face

Unit IV

Hand and Iris Biometrics: Characterization by minutiae extraction: Histogram equalization, Binarization, Skeletonization, Detection of minutiae, Matching, Performance evaluation, Preprocessing of iris images: Extraction of region of interest - Construction of noise mask – Normalization - Features extraction and encoding - Similarity measures between two iris codes. Fusion in biometrics: Multi-biometrics, Levels of fusion: Sensor level - Feature level - Rank level - Decision level fusion - Score level fusion.

References Books:

1. Rafael C Gonzalez, Richard E Woods and Steven L Eddins, “Digital Image Processing”, Pearson Education, New Delhi, 2013.
2. Amine Nait Ali and Regis Fournier, “Signal and Image Processing for Biometrics”, John Wiley and Sons, UK, 2012.
3. Arun A Ross, KarthikNandakumar and Jain A K, “Handbook of Multi-biometrics”, Springer, New Delhi 2011.
4. Oge Marques, “Practical Image and Video Processing using MATLAB”, John Wiley and Sons, New Jersey, 2011.

POOL D (ELECTIVE- IV)

POOL D

MCAIT04E19 DATA AND INFORMATION VISUALIZATION
MCAIT04E20 INFORMATION RETRIEVAL SYSTEM
MCAIT04E21 OPERATIONS RESEARCH
MCAIT04E22 COMPUTER GRAPHICS WITH OpenGL
MCAIT04E23 DESIGN AND ANALYSIS OF ALGORITHMS
MCA IT04E24 NATURE INSPIRED COMPUTING

MCAIT04E19 DATA AND INFORMATION VISUALIZATION

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the significance of data representation.
CO2: Accomplish skills to represent the facts and information.
CO3: Comprehend the methods for information visualization
CO4: Familiarize with scientific visualization techniques.

Unit I

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, Treemap, Gantt chart, Stripe graphic, Animated spiral graphic. Visualization based on types of data: Visualizing Categorical data , Visualizing Time series data , Visualizing Spatial data.

Unit II

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.

Unit III

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. Case study with real world problem.

Unit IV

Scientific visualization: Introduction, Methods for visualizing two-dimensional 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.

Reference Books:

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
4. Riccardo Mazza. Introduction to Information Visualization, Springer, 2009
5. Gowri shankar S, Veena A, "Introduction to Python Programming", 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372
6. Alberto Fernandez Villan, Mastering OpenCV 4 with Python, Packt Publishing Ltd
7. Dr. R Nageswara Rao, Core Python Programming, 2nd edition, Dreamtech Publisher, 2019
8. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st Edition,O'Reilly Media, 2017.
9. Wesley J. Chun, Core Python Programming, Second Edition, Publisher: Prent Hall Pub
10. Introduction to Computer Science using Python - Charles Dierbach, Wiley,2015

MCAIT04E20 INFORMATION RETRIEVAL SYSTEM

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understand the significance of Information Retrieval System.
CO2: Familiarize with information retrieval strategies
CO3: Accomplish knowledge about Distributed Information Retrieval
CO4: Awareness about the Information Retrieval algorithms.

Unit I

Introduction: Retrieval strategies: vector space model, Probabilistic retrieval strategies: Simple term weights, Non binary independence model, Language models.

Unit II

Retrieval Utilities: Relevance feedback, clustering, N-grams, Regression analysis, Thesauri.

Unit III

Semantic networks, parsing Cross –Language: Information Retrieval: Introduction, Crossing the Language barrier.

Unit IV

Integrating structured data and text. A historical progression, Information retrieval as relational application, Semi Structured search using a relational schema. Distributed Information Retrieval: A theoretical Model of Distributed retrieval, web search.

Text book:

1. David A. Grossman, Ophir Frieder, Information Retrieval – Algorithms and Heuristics, Springer, 2nd Edition (Distributed by Universal Press), 2004

Reference books:

1. Gerald J Kowalski, Mark T Maybury,"Information Storage and Retrieval Systems: Theory and Implementation", Springer, 2004.
2. SoumenChakrabarti, "Mining the Web: Discovering Knowledge from Hypertext Data", Morgan – Kaufmann Publishers, 2002.
3. Christopher D Manning, PrabhakarRaghavan, HinrichSchutze, "An Introduction to Information Retrieval" , Cambridge University Press, England, 2009.

MCAIT04E21 OPERATIONS RESEARCH

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Recognize the need of Operations Research
CO2: Understand about the applications of Linear Programming
CO3: Acquire knowledge about Integer Programming
CO4: Obtain knowledge about project scheduling
CO5: Obtain knowledge about simulation fundamentals

Unit I

Linear programming: Formulation, Graphical Solution-2 variables, Development of Simplex Method, Artificial Variable Techniques, Big- M method, Two-Phase method, Reversed Simplex method. Duality in LPP and its formulation, Dual Simplex Method, Bounded variable method, Applications of LPP.

Unit II

Transportation problems, Assignment Problem, Traveling Sales persons problem. Integer Programming problem (IPP), Cutting Plane algorithm, Branch and bound method of solving IPP, Dynamic programming problems and it's characteristics, Deterministic Dynamic Programming Problem.

Unit III

Sequencing Problem, Processing **n** jobs through two machines and their mechanics, Processing **n** jobs through **m** machines, Processing **2** jobs through **m** machines, Project scheduling by PERT / CPM, Difference between PERT / CPM, Constructing the network, Critical path analysis, Float of an activity, Three time estimated for PERT, project cost by CPM.

Unit IV

Simulation: simulation concepts, simulation of a queuing system using event list, pseudo random numbers, multiplication congruential algorithm, inverse transformation method, Basic ideas of Monte-Carlo simulation.

Reference Books:

1. Thaha H.A.- Operation Research, 9THEdn, Pearson
2. Sharm J.K, Mathematical Models in Operation Research, TMGH, 1989.
3. Trivedi,. Probability, Statistics with Reliability, Queuing and Computer Science Applications, PHI
4. Winston, Operations Research Applications and Algorithms, 4thedn, CENGAGE, 2003Sons, 2007.

MCAIT04E22 COMPUTER GRAPHICS WITH OpenGL

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Acquire knowledge about the Graphics systems and Working with OPENGL
CO2: Attain the knowledge about Geometric Transformation (2D and 3D)
CO3: Learn in detail about 2D and 3D viewing pipelines
CO4: Learn Different types of Projections
CO5: Advance knowledge about 3D Object Representation
CO6: Learn Illumination models Surface Rendering and Ray tracing

UNIT I

Computer Graphics and Overview of Graphics systems: Applications of Computer Graphics, Video display devices, Raster scan systems, Graphic workstations and viewing systems, Input devices, Graphics software, introduction to OpenGL. Graphics Output Primitives: Coordinate reference frames, Line drawing algorithms (DDA and Bresenham's), OpenGL curve functions, Circle generating algorithms (Midpoint circle and Bresenham's), Pixel addressing and Object geometry, fill area primitives, Polygon fill areas. Attributes of graphics primitives: Color and Gray scale, point attributes, Line attributes, Fill-Area attributes, General Scan-line polygon fill algorithm, Scan-Line fill of convex-polygons, Boundary fill and flood fill algorithms, Antialiasing and antialiasing functions in OPENGL. Two-dimensional viewing: 2D viewing pipeline, Clipping window, normalization and viewport transformation, Clipping algorithms, point clipping, line clipping (Cohen-Sutherland, Nichol-Lee Nichol), Polygon Fill-area clipping (Sutherland – Hodgeman), Text Clipping.

UNIT II

Geometric Transformations: Basic 2D transformation, Matrix representation and Homogeneous coordinates, Inverse transformations, 2D composite transformations, Reflection and shear, Raster methods for geometric transformations, Transformations between 2D coordinate systems. 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 III

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), Perspective projections (transformation coordinates, perspective-projection equations, vanishing 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)

UNIT IV

3D Object representation: Quadric surfaces, super quadrics, blobby objects, spline representations. Visible surface detection methods: Classification, Back-face detection, depth-Buffer method, A buffer method. Wireframe visibility methods. Illumination models and surface rendering methods: Light sources, Surface lighting effects, Basic illumination models (Ambient light, Diffuse reflection, Specular reflection and the Phong model), polygon rendering methods (constant intensity surface rendering, Gouraud surface rendering, Phong surface rendering), Ray tracing methods – basic Ray-tracing algorithm.

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.

MCAIT04E23 DESIGN AND ANALYSIS OF ALGORITHMS

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Accomplish Knowledge about important computational problems.
CO2: Acquire knowledge to design the algorithm.
CO3: Study to analyze a given algorithm.
CO4: Obtain knowledge to analyze algorithm control structures and solving recurrence.
CO5: Attain information about Complexity Classes
CO6: Accomplish knowledge about Parallel Algorithms

Unit I

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).

Unit II

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.

Unit III

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 IV

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.

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, Sk Kataria & 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

MCAIT04E24 NATURE INSPIRED COMPUTING

Contact Hours/ week: 3 **Credit: 3**

COURSE OUTCOME

CO1: Understanding of Natural Inspirations in problem solving
CO2: Acquire knowledge about Ant Colony Optimization
CO3: Obtain knowledge about Swarm Intelligence
CO4: Gain knowledge about Genetic algorithms
CO5: Attain knowledge about DNA Computing

Unit I

Introduction: Nature Inspired Computing. Natural to Artificial Systems- Biological Inspirations in problem solving- Behavior of Social Insects: Foraging- Division of labor- Task Allocation – Cemetery Organization and Brood Sorting – Nest Building – Cooperative Transport.

Unit II

Ant Colony Optimization : Ant Behavior – Towards artificial Ants - Ant Colony Optimization – Problem solving using ACO – Extensions of Ant Systems – Applications. Swarm Intelligence: Introduction to Swarm Intelligence – Working of Swarm Intelligence –Particle Swarms Optimization– Applications.

Unit III

Introduction to Genetic algorithms – Population Initialization – Choosing a Fitness Function – Selection – Crossover – Mutation – Reinsertion – Applications of Genetic Algorithms – Evolutionary Algorithms. Other Biological computing Methods – Immune System Algorithms – Cellular Automata – Linden Meyer Systems – Artificial Neural Networks – Simulated Annealing.

Unit IV

Computing With New Natural Materials: DNA Computing: Motivation, DNA Molecule , Adelman's experiment , Test tube programming language, Universal DNA Computers , PAM Model , Splicing Systems, Lipton's Solution to SAT Problem , Scope of DNA Computing , From Classical to DNA Computing.

Text books

1. Stephen Olariu and Albert Y.Zomaya, "Handbook of Bio-Inspired and Algorithms and Applications", Chapman and Hall, 2006.
2. Marco Dorrigo, Thomas Stutzle," Ant Colony Optimization", PHI,2004
3. Eric Bonabeau, Marco Dorrigo, Guy Theraulaz, "Swarm Intelligence: From Natural to Artificial Syatems", Oxford University Press,2000
4. Mitchell, Melanie, "Introduction to Genetic algorithms",ISBN:0262133164,MIT Press,1996
5. Leandro Nunes de Castro, " Fundamentals of Natural Computing, Basic Concepts, Algorithms and Applications", Chapman & Hall/ CRC, Taylor and Francis Group, 2006

Reference books

1. Floreano D. and Mattiussi C., "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies", MIT Press, Cambridge, MA, 2008.
2. Leandro Nunes de Castro and Fernando .J, " Recent Developments in Biological Inspired Computing", MIT Press,2005

Open Elective Course

MCAIT02O01 DESIGN AND ANALYSIS OF ALGORITHMS

Contact Hours/ week: (Theory: 3 + Practical: 1) **Credit: 4**

COURSE OUTCOME

CO1: Accomplish Knowledge about important computational problems.
CO2: Acquire knowledge to design the algorithm.
CO3: Study to analyze a given algorithm.
CO4: Obtain knowledge to analyze algorithm control structures and solving recurrence.
CO5: Attain information about Complexity Classes
CO6: Accomplish knowledge about Parallel Algorithms

Unit I

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).

Unit II

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.

Unit III

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 IV

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.

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, Sk Kataria & 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