KANNUR UNIVERSITY

(Abstract)

Master of Computer Application (MCA) Programme in the Department of Information Technology, Mangattuparamba Campus - Modified Scheme (I st semester only), II nd, III rd and IV th Semester syllabus - Approved- Implemented w.e.f. 2023 admission--Orders issued

---

ACADEMIC C SECTION

ACAD C/ACAD C1/23379/2023 Dated: 06.04.2024

Read:-1. U. O. No. ACAD C/ACAD C1/23379/2023 dated 13.11.2023

2. Circular No. ACAD C/ACAD C3/22373/2019 dated 01/02/2024

3. Email dated 28.02.2024 from the Head, Department of Information Technology, Mangattuparamba Campus.

4.Minutes of the meeting of the Department Council held on 13.02.2024

5. Orders of the Vice chancellor dated 5-4-2024.

ORDER

1.The Scheme (All semesters) & Syllabus (I st Semester only) of Master of Computer Application (MCA) Programme under Choice Based Credit and Semester System in the Department of Information Technology, Mangattuparamba campus were implemented w.e.f. 2023 admissions vide paper read(1) above.

2. As per paper read (2) above, Heads of Teaching Departments who have not yet submitted the completed syllabus of PG Programmes were requested to submit the Syllabus of remaining semesters, prepared in tune with the approved Scheme / Credit distribution table along with a copy of the Department Council Minutes.

3. Subsequently, the HoD, Department of Information Technology submitted the remaining semester syllabus approved by the Department council(Paper read 4) He has also rectified the discrepancy between the name of Discipline Specific Elective courses in the scheme part and detailed part of the syllabus approved vide Paper read 1 above.

4.The Vice Chancellor after considering the matter in detail, and in exercise of the powers of the Academic Council conferred under section 11(1), Chapter III of Kannur University Act 1996, approved the **modified Scheme , II nd, III rd& IV th semester Syllabus of Master of Computer Application (MCA) Programme and accorded sanction to implement the same in the Department of Information Technology, Mangattuparamba Campus of the University w.e.f. 2023 admissions, subject to report to the Academic Council.**

5.The Scheme & Syllabus of Master of Computer Application Programme, under Choice Based Credit Semester System implemented in the Department of Information Technology, Mangattuparamba Campus w. e. f. 2023 admission, is appended and uploaded in the University Web Site.(www.kannuruniversity.ac.in)

6.Orders are issued accordingly.

Sd/-

Narayanadas K

DEPUTY REGISTRAR (ACAD)

For REGISTRAR

To: 1.The Head, Department of Information Technology, Mangattuparamba Campus.

2. Convener, Curriculum Committee.

Copy To: 1. The Examination branch (through PA to CE)
2. PS to VC/ PA to PVC/PA to R
3. DR/AR1/AR II (Acad), EXCI, EP IV
4. Web Manager (for uploading in the website)
5. Computer Programmer
6.SF/DF/FC

Forwarded / By Order

SECTION OFFICER

KANNUR UNIVERSITY
CIVIL STATION P.O.
KANNUR
Pin-670 002
THAVAKKARA

DEPARTMENT OF INFORMATION TECHNOLOGY

KANNUR UNIVERSITY

DEGREE OF

MASTER OF COMPUTER APPLICATIONS (MCA)

(CHOICE BASED CREDIT AND SEMESTER SYSTEM)

SYLLABUS

(FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2023 – 24 ONWARDS)

REGULATIONS FOR THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS

(MCA)

FOR THE STUDENTS ADMITTED FROM THE ACADEMIC YEAR 2023 – 24 ONWARDS

PROGRAMMESPECIFIC OUTCOMES

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications
PSO2	Enhance the knowledge and skills about System Software and Application Software
PSO3	Attain skills to design Algorithms and Programs
PSO4	Acquire the knowledge in the building, designing and managing IT and IT enabled infrastructure
PSO5	Design, build, and test software systems to meet the given specifications by following the principles of Software Engineering
PSO6	Inculcate research and development culture in the emerging areas of Computer Science and Information Technology
PSO7	Equip the candidate to take up challenging positions in the area of research and development linked with Information Technology and Computer Application

1. ADMISSION

Admission to the MCA programme will be carried out as per the Kannur University PG regulations – 2023.

Eligibility for Admission:

BCA/ Bachelor Degree in Computer Science/Engineering or Equivalent Degree with not less than 50% marks (45% for candidates belonging to reservation category) or equivalent grade.

OR

B.Sc./B.Com./B.A with Mathematics at 10+2 level or at Graduation level (such candidates shall undergo additional Bridge Courses as per the norms of Kannur University) with not less than 50% marks (45% for candidates belonging to reservation category) or equivalent grade.

2. PROGRAMME STRUCTURE

Duration of the MCA programme shall be 2 years, divided into 4 semesters. Each semester shall have 18 weeks. The minimum duration for the completion of the MCA programme is four (4) semesters from the date of registration. The maximum period of completion is eight semesters (4 years) from the date of registration. Every student has to opt the various categories of course [Discipline Specific Core (DSC), Discipline Specific Electives (DSE), Value Added Courses (VAC), Skill Enhancement Courses (SEC), Ability Enhancement Courses (AEC), Multi-Disciplinary Courses (MDC), Inter Disciplinary Courses (IDC), and MOOC] as per Kannur University PG Regulations-2023.

For MCA programme, all students have to take a MOOC (of minimum 2 credits) and submit the course completion certificate obtained from the MOOC provider at the end of third semester. Credit earned from the MOOC will be over and above the minimum credit required for completion of MCA Degree (Refer Kannur University PG Regulations-2023).

For the MCA programme, a minor project work with 2 credits (offered in the third semester) is opted by the department.

The minimum credits required for the successful completion of the MCA programme will be 84. Every student should earn a minimum of 8 credits offered by the other department

For all core courses an additional module (called as Module X) is added. Portions of this module need not be discussed by the faculty during the designated classroom hours for the given course. However, components of continuous evaluation such as assignment / seminar / viva may be based on the contents of Module X. Module X has to be considered as additional reading and self-exploration by the students. For end semester examinations, contents of Module X will not be considered.

MOOC

Kannur University PG Regulations-2023 will be applicable for MOOC.

For MCA programme, all students have to take a MOOC (of minimum 2 credits) and submit the course completion certificate obtained from the MOOC provider at the end of third semester. Department council will prepare a list of MOOCs from the online courses offered by the NPTEL / any other MOOC provider approved by the council from time to time at the beginning of semesters 1, 2 and 3. Students have to choose a MOOC from this list and earn his/her certificate of completion as per the guidelines stipulated by the MOOC provider concerned from time to time.

The credit earned by the student for the MOOC will be counted as such without any normalization.

3. EVALUATION

Evaluation (Both CE and ESE) will be conducted as per the Kannur University PG Regulations- 2023 for all theory and practical courses. However, the specific guidelines for conducting the evaluation of theory courses, practical courses, mini project and major project are mentioned below.

CONTINUOUS EVALUATION FOR THEORY COURSES (DSC, DSE, VAC, IDC, MDC, AEC, SEC)

CE includes assignments, seminars, viva and periodic written examinations. This should be done by the faculty who manages the course.

The weightage of each component under CE for theory courses shall be in the following proportions:

Components	% of Weightage
Test papers (minimum two Continuous Evaluation Tests)	40
Assignments	20
Seminar / Viva	40

Table 1: Weightage of each Component under Continuous Evaluation for Theory Courses

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 grades shall be taken for the award of CE grades. The dates of test papers shall be announced well in advance and the result should be displayed in the notice board. Tests such as multiple-choice objective type and open text book test (online or offline mode) also can be opted for conducting the test papers.

Assignments: For each course at least one assignment (including practical assignments, if necessary) shall be assigned to the students. The mode of submission and assessment of the assignments shall be decided by the faculty concerned. Assignment works can be conducted either offline/ online mode (as per the decision taken by the faculty concerned).

Viva: Faculty concerned can assign topics for comprehension (based on any portions in the syllabus) and ask the students to appear for individual viva sessions as per a declared schedule. Viva sessions can be conducted either in online or offline mode (as per the decision taken by the faculty concerned).

Seminar: Faculty concerned can assign topics for comprehension (based on the course concerned) and ask the students to prepare seminars based on the topics assigned to them. Each student has to prepare the seminar content and present it. Mode of preparation, submission, and presentation can be specified by the faculty member concerned. Seminar sessions and contents can be submitted and conducted either in online or offline mode (as per the decision taken by the faculty concerned).

TECHNOLOGY SPECIFIC ELECTIVE (ELECTIVE V)

In the fourth semester an Elective course (Elective V) are meant to foster the students with tools and technologies that they need to know and make use in the design and development of software applications. The department council will prepare the list of elective courses to be offered for Elective V at the end of every third semester. Seminar/Report/ Case study implementation report of the specific technology mentioned in the elective should be submitted by each student for the evaluation. The mode of evaluation of this course shall be based on the presentation, report and viva. Both CE and ESE for this course will be conducted by the Department.

CONTINUOUS EVALUATION FOR PRACTICAL COURSES

The components of CE for practical courses are as follows:

Components	% of Weightage
Lab test (minimum 1)	40
Completion of the list of Lab assignments prescribed by the faculty	20
Periodical assessment of assignment in the Lab	40

Table 2: Weightage of each Component under Continuous Evaluation for Practical Courses

EVALUATION FOR MINOR PROJECT WORK

The components of CE and ESE for minor project work are as follows:

Components	% of Weightage
Understanding of the problem / Concepts	20
Adhering to methodology	15
Quality of presentation and demonstration	15
Quantum of work / effort	25
Organization and content of Project report	5
Viva based on Project	20

Table 3: Weightage of each Component for CE/ ESE for Minor Project Work

CE and ESE of the minor project work shall be done by a departmental committee constituted by the HOD. The committee should consist of a minimum of two faculty members, including the guide. Phases of evaluation and evaluation criteria for each phase shall be framed by the departmental committee.

4. PROJECT WORK

Project Work offered in the fourth semester currently has 8 credits. 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 project work should be carried out in the department / Institution / industry / R & D organization of national repute. Project work shall be carried out under the supervision of a faculty member. If the student wishes to undertake his / her project outside the campus, then a co-guide shall be selected from the organization concerned. 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 approval of the project synopsis from the supervisor in the department before the commencement of the project. A co-guide should be an expert in the area in which the 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 these regulations / guidelines. Students have to submit the copies of the reports that are approved by the project supervisor(s) before the last date fixed by the department.

The end semester evaluation of the project work shall be done by a board of at least two examiners, in which one should be an external expert. For the evaluation of the project work, the candidate must present the work before the board of examiners which will be followed by a Viva-Voce. The end semester evaluation of the project will be based on the project report, the presentation of the project work undertaken by the student and Viva-Voce.

The weightages for CE and ESE of the project also shall be in the ratio 40:60.

EVALUATION OF PROJECT WORK

CE of the project work shall be done by a departmental committee constituted by the HOD. The committee should consist of a minimum of two faculty members, including the guide.

The assessment is based on presentation, interim report and viva voce. Each internal presentation shall be evaluated based on the following components:

Components	% of Weightage
Understanding of the problem /Concepts	20
Adhering to methodology	15
Quality of presentation and demonstration	15
Quantum of work / effort	25
Organization and content of Project report	5
Viva based on Project	20

Table 4: Components for Continuous Evaluation and the Corresponding Weightage (for Project Work)

End Semester Evaluation (ESE): A board of two examiners appointed by the university shall conduct ESE. 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 project reports that are duly certified by the guide and HOD. 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 ESE shall consist of the following components:

COMPONENTS	% of Weightage
Understanding of the problem/requirements/ concepts related to the project	15
Adhering to methodology (Software engineering phases or research methodology) and the candidates understanding of the components of methodology	15
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
Quality of presentation / demonstration	15
Quantum of work / effort - assessed through the content of report, presentation and viva	25
Organization and content of report	10

Table 5: Components for ESE with the Corresponding Weightage (for Project Work)

Guideline for Preparing Project Report (Both Minor Project Work and Project Work)

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
10) Conclusion
11) Publications based on the project work (if any)
12) Appendices
13) References

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 into several chapters and each chapter may be further divided into several divisions and subdivisions. 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. The tables and figures shall be introduced at appropriate places.

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.

ii) Page Dimension and Binding Specifications:

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

All the project reports submitted by the students should be plagiarism checked using software and the plagiarism report generated by the software should be verified and signed by the HOD.

Bridge Course

1) Introduction

The bridge course comprises 80 hours teaching and learning activity. It consists of two theory papers and one laboratory paper. 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 enroll 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. The list of all successful candidates shall be forwarded to the university along with CE mark list of the Ist semester MCA programme.

2) Conduct of Classes

Department council shall schedule regular classes (may be online class or MOOCs) for eighty (80) hours. The classes should be over 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

The course shall comprise three (03) courses - two (02) theory courses and one (01) practical course. Candidate has to appear examinations for these courses, 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 course with two (02) hours duration and complete the evaluation process of all those courses 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 two (02) unique set of question papers on the subject s(he) taught. 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 HOD in sealed cover. The HOD shall constitute a Board of Examiners (BoE) by including all the faculties in the department (minimum three faculties) with the HOD as the Chairman. BoEwill scrutinize 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 BoE (mentioned in item no 6 above) will 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 course. Only single valuation will be done.

9) Finalizing the Results of Bridge Course

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. The entire 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 Course

Code	Course	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

Module 1: 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)

Module 2: 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)

Module 3: 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)

Module 4: 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 Books:

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

Module 1: 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'stheorms - 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)

Module 2: 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)

Module 3: 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)

Module 4: Numerical Differentiation: Based on equal-interval Interpolation, Derivatives using Newton's backward difference formula. NumericalIntegration:-Trapezium rule, Trapezoidal rule, Simpson's rule. Differential equations: Preliminaries, Taylor series method, RungeKuttamethods-Statistical description and modelling of data. (08 Hours)

Text Books:

1. Thomas L Floyd - Digital Fundamentals, Pearson International Edition (9th Edition), Prentice Hall. (I and II Units)
2. Balachandra Rao, C K Shanghai – "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)

End of Bridge Course

MCA

LEARNING OUTCOME BASED CURRICULUM FRAME WORK &PROGRAMME STRUCTURE

Graduate Attributes

GRADUATE ATTRIBUTE 1 - SCHOLARSHIP

KANNUR UNIVERSITY graduates will be able inquire critically into their area of study, while being aware of changing state of knowledge both in their own chosen discipline as well as related disciplines

KANNUR UNIVERSITY graduates will have the ability to actively engage in the generation of innovative and relevant knowledge and understanding through inquiry, critique and synthesis going beyond their discipline of specialization.

GRADUATE ATTRIBUTE 2 - CRITICAL CITIZENSHIP AND THE SOCIAL GOOD

KANNUR UNIVERSITY graduates will be engaged, committed and accountable agents of social good. They must aspire to contribute to social justice and environmental sustainability, appreciative of the complexity of historical contexts and societal conditions through their roles as professionals and members of local and global communities.

KANNUR UNIVERSITY graduates will be committed to furthering gender and social equality and empathetically engage with all forms of difference including, conflicting intellectual traditions, religious and cultural practices, language, region and nationality.

GRADUATE ATTRIBUTE 3 - LIFELONG LEARNING

KANNUR UNIVERSITY graduates will be Lifelong Learners, committed to and capable of continuous collaborative and individual learning and critical reflection for the purpose of furthering their understanding of the world and their place in it.

Programme Outcomes (PO)

SL #	Outcome
PO1	**Critical Thinking**: Take informed actions after identifying the assumptions that frame our thinking and actions, checking out the degree to which these assumptions are accurate and valid, and looking at our ideas and decisions (intellectual, organizational, and personal) from different perspectives
PO2	**Problem Solving**: Identify, formulate, conduct investigations, and find solutions to problems based on in-depth knowledge of relevant domains
PO3	**Communication**: Speak, read, write and listen clearly in person and through electronic media in English/language of the discipline, and make meaning of the world by connecting people, ideas, books, media and technology.
PO4	**Responsible Citizenship**: Demonstrate empathetic social concern, and the ability to act with an informed awareness of issues
PO5	**Ethics**: Recognize different value systems including your own, understand the moral dimensions of your decisions, and accept responsibility for them.
PO6	**Self-directed and Life-long Learning**: Acquire the ability to engage in independent and life-long learning in the broadest context socio- technological changes
PO7	**Environmental Sustainability and Global Perspective:** - Develop an understanding of global standards to foster legal environment. Learn and practice to critically analyse the legal issues from local, national and international concerns

Programme Specific Outcomes (PSO)

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications
PSO2	Enhance the knowledge and skills about System Software and Application Software
PSO3	Attain skills to design Algorithms and Programs
PSO4	Acquire the knowledge in the building, designing and managing IT and IT enabled infrastructure
PSO5	Design, build, and test software systems to meet the given specifications by following the principles of Software Engineering
PSO6	Inculcate research and development culture in the emerging areas of Computer Science and Information Technology
PSO7	Equip the candidate to take up challenging positions in the area of research and development linked with Information Technology and Computer Science

PSO - PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7
PSO1	✔	✔	✔	✔	✔	✔	✔
PSO2	✔	✔	✔		✔	✔	✔
PSO3	✔	✔	✔	✔	✔	✔	✔
PSO4	✔	✔	✔	✔	✔	✔	✔
PSO5	✔	✔	✔	✔	✔	✔	✔
PSO6	✔	✔	✔	✔	✔	✔	✔
PSO7	✔	✔	✔	✔	✔	✔	✔

MCA

PROGRAMME STRUCTURE

LEGEND	
Item	**Description**
C	Credits
ESA	% of the Weightage of the End Semester Evaluation
CE	% of the Weightage of the Continuous Evaluation
P	Practical Hours
T	Total
Tt	Tutorial
DSC	Discipline Specific Core Course
DSE	Discipline Specific Elective Course
MDC	Multi Disciplinary Elective Course
IDC	Interdisciplinary Elective Course
VAC	Value Added Course
SEC	Skill Enhancement Course
AEC	Ability Enhancement Course

Course Structure

<table>
<tr><td colspan="10">Distribution of credits for the MCA Programme with effect from 2023- 24 Onwards</td></tr>
<tr><td></td><td>1</td><td colspan="2">2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>Total Credits</td></tr>
<tr><td></td><td colspan="2">Discipline Specific</td><td>Electives</td><td colspan="2">Offered for Other Departments</td><td></td><td></td><td></td><td></td></tr>
<tr><td>#</td><td>Core Courses (DSC)</td><td>Electives (DSE)</td><td>IDC/MDC</td><td>AEC 2Credits</td><td>SEC 2 Credits</td><td>VAC / MOOC 2 Credits</td><td>Minor Project 2 Credits</td><td>Major Project 8 Credits</td><td></td></tr>
<tr><td>1</td><td>MCCSA01DSC01(3C)
MCCSA01DSC02(4C)
MCCSA01DSC03(3C)
MCCSA01DSC04(3C)
MCCSA01DSC05(2C)
MCCSA01DSC06(2C)</td><td>Pool A
MCCSA01DSE01 to 03</td><td></td><td></td><td></td><td></td><td></td><td></td><td>21</td></tr>
<tr><td></td><td>17 Credits</td><td>4 Credits Each</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>MCCSA02DSC07(3C)
MCCSA02DSC08(3C)
MCCSA02DSC09(3C)
MCCSA02DSC10(3C)
MCCSA02DSC11(3C)
MCCSA02DSC12(2C)
MCCSA02DSC13(2C)</td><td></td><td></td><td>MCCSA02AECXX</td><td>MCCSA02SECXX</td><td></td><td></td><td></td><td>23</td></tr>
<tr><td></td><td>19 Credits</td><td></td><td></td><td>2Credits</td><td>2 Credits</td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td>MCCSA03DSC14(4C)
MCCSA03DSC15(4C)
MCCSA03DSC16(3C)
MCCSA03DSC17(2C)</td><td>Pool B
MCCSA03DSE04 to 11</td><td>MSCSC03MDC01to 07 (Offered for other departments)</td><td></td><td></td><td>MCCSA03VAC01</td><td>MCCSA03DSC18</td><td></td><td>22</td></tr>
<tr><td></td><td>13 Credits</td><td>3 Credits Each</td><td>4 Credits Each</td><td></td><td></td><td>2*</td><td>2 Credits</td><td></td><td></td></tr>
<tr><td>4</td><td></td><td colspan="2">Elective III (Pool C)</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>MCCSA04DSE12 to 19</td><td></td><td></td><td></td><td></td><td></td><td>MCCSA04DSC19(8C)</td><td>18</td></tr>
<tr><td></td><td></td><td colspan="2">4 Credits from Pool C</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Elective IV (Pool D)</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>MCCSA04DSE20 to 28</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">4 Credits from Pool D</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>MCCSA04DSE29
Elective V (2C)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">10 Credits</td><td></td><td></td><td></td><td></td><td>8 Credits</td><td></td></tr>
<tr><td></td><td colspan="8">Total Credits for MCA Programme</td><td>84</td></tr>
</table>

*MOOC Credit will not be counted for CGPA. However, it is compulsory. Credit earned by the students for the MOOC will be entered as provided by the MOOC provider.

Semester I

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
1.1	MCCSA01DSC01	Mathematical Foundations for Computer Science	3	3	0	0	60	40	100
1.2	MCCSA01DSC02	System software and Operating systems	4	4	0	1	60	40	100
1.3	MCCSA01DSC03	Computer Network and Linux Administration	3	3	0	1	60	40	100
1.4	MCCSA01DSC04	Web Technology	3	3	0	1	60	40	100
1.5	MCCSA01DSEXX	Elective I – DSE (Pool A)	4	4	0	0	60	40	100
1.6	MCCSA01DSC05	Lab I: Principles of Programming	2	0	4	1	60	40	100
1.7	MCCSA01DSC06	Lab II: Network and Linux Administration	2	0	4	1	60	40	100
		Total	**21**	**17**	**8**	**5**			

S1 - List of Courses for Elective I – DSE (Pool A)		
No	**Course Code**	**Course Name**
1.5a	MCCSA01DSE01	Principles of Programming using Python
1.5b	MCCSA01DSE02	Principles of Programming using CPP
1.5c	MCCSA01DSE03	Principles of Programming using C

Semester II

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
2.1	MCCSA02DSC07	Algorithms and Data Structure	3	3	0	1	60	40	100
2.2	MCCSA02DSC08	Programming in Java	3	3	0	0	60	40	100
2.3	MCCSA02DSC09	Database Management Systems	3	3	0	0	60	40	100
2.4	MCCSA02DSC10	Theory of Computation	3	3	0	1	60	40	100
2.5	MCCSA02DSC11	Software Engineering	3	3	0	1	60	40	100
2.6	MCCSA02DSC12	Lab III: i) Data Structure ii) Java	2	0	4	1	60	40	100
2.7	MCCSA02DSC13	Lab IV: i) DBMS; ii) WT	2	0	4	1	60	40	100
2.8	Offered by Other Dept.	SEC/AEC	2	2	0	0	60	40	100
2.9	Offered by Other Dept.	SEC/AEC	2	2	0	0	60	40	100
		Total	**23**	**19**	**8**	**5**			

AECs (offered for other Departments)

No	Course Code	Course Name	C	T	P	Tt
1	MCCSA02AEC01	Algorithm Analysis	2	2	0	2
2	MCCSA02AEC02	Artificial Intelligence and Daily Life	2	2	0	2
3	MCCSA02AEC03	Python Programming	2	2	0	2
4	MCCSA02AEC04	Data Processing with Python	2	2	0	2
5	MCCSA02AEC05	Fundamentals of Big Data	2	2	0	2
6	MCCSA02AEC06	Fundamentals of Programming Methodology	2	2	0	2
Total			**2**	**2**	**0**	**2**

SECs (offered for other Departments)

No	Course Code	Course Name	C	T	P	Tt
1	MCCSA02SEC01	Basics of Machine Learning	2	2	0	2
2	MCCSA02SEC02	Data Science Fundamentals	2	2	0	2
3	MCCSA02SEC03	Optimization Techniques	2	2	0	2
4	MCCSA02SEC04	Scientific Computing	2	2	0	2
5	MCCSA02SEC05	The Art of E - Documentation using Latex	2	2	0	2

6	MCCSA02SEC06	Fundamentals of Digital Skilling using Google Workspace for Education	2	2	0	2
7	MCCSA02SEC07	Image Processing using Python	2	2	0	0
8	MCCSA02SEC08	Fundamentals of Electrical and Electronics Engineering	2	2	0	0
9	MCCSA02SEC09	Data Analysis and Plotting	2	2	0	0
10	MCCSA02SEC10	Quantum computing	2	2	0	0
Total			**2**	**2**	**0**	**2**

Semester III

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
3.1	MCCSA03DSC14	Machine Learning Techniques	4	4	0	1	60	40	100
3.2	MCCSA03DSC15	Computer Graphics and Image Processing	4	4	0	1	60	40	100
3.3	MCCSA03DSC16	Quantum Computing	3	3	0	1	60	40	100
3.4	MCCSA03DSC17	Lab V: (i) Machine Learning (ii)CG and IP	2	0	4	1	60	40	100
3.5	MCCSA03DSE04 to 11	DSE (POOL B)	3	3	0	0	60	40	100
3.6	Offered by Other Dept.	IDC/MDC	4	4	0	0	60	40	100
3.7	MCCSA03VAC01	MOOC*	2*	0	0	0	60	40	100
3.8	MCCSA03DSC18	Minor Project	2	0	4	1	60	40	100
		Total	**22**	**18**	**8**	**5**			

- *MOOC credit will not be counted for CGPA however it is compulsory

Multi-Disciplinary Courses (offered for other Departments)						
No	**Course Code**	**Course Name**	**C**	**T**	**P**	**Tt**
1	MSCSC03MDC01	Design and Analysis of Algorithms	4	4	0	0
2	MSCSC03MDC02	Principle of Programming and Numerical Methods	4	4	0	0
3	MSCSC03MDC03	Java Programming	4	4	0	0
4	MSCSC03MDC04	Machine Learning	4	4	0	0
5	MSCSC03MDC05	Foundations in Data science	4	4	0	0
6	MSCSC03MDC06	Digital Signal Processing	4	4	0	0

7	MSCSC03MDC07	Quantum Computing and Information Theory	4	4	0	0
Total			**4**	**4**	**0**	**0**

S3 - List of Discipline Specific Electives for (Pool B)

No	Course Code	Course Name
3.5a	MCCSA03DSE04	Artificial Intelligence
3.5b	MCCSA03DSE05	Bioinformatics
3.5c	MCCSA03DSE06	Fuzzy Sets and Systems
3.5d	MCCSA03DSE07	Graph Theory and Combinatorics
3.5e	MCCSA03DSE08	Software Architecture
3.5f	MCCSA03DSE09	Foundations of Natural Language Processing
3.6g	MCCSA03DSE10	Computer Organization and Architecture
3.7h	MCCSA03DSE11	Foundations in Data Science

Semester IV

No	Course Code	Course Name	C	Hrs./wk.			Assessment Weightage (%)		
				L	P	Tt	ESA	CE	T
4.1	MCCSA04DSE12 to 19	Elective III – DSE (POOL C)	4	4	0	0	60	40	100
4.3	MCCSA04DSE20 to 28	Elective IV-DSE (POOL D)	4	4	0	0	60	40	100
4.5	MCCSA04DSE29	ELECTIVE V*	2	2	0	0	60	40	100
4.6	MCCSA04DSC19	Project	8	0	15	5	60	40	400
		Total	**18**	**10**	**15**	**5**			

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

S4 - List of Courses for Elective III – DSE (Pool C)			
No	**Course Code**	**Type of Elective**	**Course Name**
4.1a	MCCSA04DSE12	DSE	Big Data Analytics
4.1b	MCCSA04DSE13	DSE	Operations Research
4.1c	MCCSA04DSE14	DSE	Algorithms in Computational Biology
4.1d	MCCSA04DSE15	DSE	Object Oriented Analysis and Design
4.1e	MCCSA04DSE16	DSE	Computer Vision
4.1f	MCCSA04DSE17	DSE	Software Project Management
4.1g	MCCSA04DSE18	DSE	Visual Cryptography
4.1h	MCCSA04DSE19	DSE	Biometric Image Processing

S4 - List of Courses for Elective IV – DSE (Pool D)			
No	**Course Code**	**Type of Elective**	**Course Name**
4.2a	MCCSA04DSE20	DSE	Nature Inspired Computing
4.2b	MCCSA04DSE21	DSE	Pattern Recognition
4.2c	MCCSA04DSE22	DSE	Cyber Forensics
4.2d	MCCSA04DSE23	DSE	Natural Language Processing with Python
4.2e	MCCSA04DSE24	DSE	Grid and Cloud Computing
4.2f	MCCSA04DSE25	DSE	Information Security
4.2g	MCCSA04DSE26	DSE	Data and Information Visualization
4.2h	MCCSA04DSE27	DSE	Information Retrieval System
4.2i	MCCSA04DSE28	DSE	Design and Analysis of Algorithms

Semester I

SEMESTER I
CORE COURSE
MCCSA01DSC01 MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CA	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

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

Course Objectives:

- Impart knowledge on mathematical logic.
- Give basic idea set theory, relations, functions and graphs and their problem solving.
- Familiarize measures of central tendency and measures of dispersion.
- Impart knowledge on probability and its distributions.
- Familiarize matrices and its operations, vector space and Eigen vectors.

Course Outcomes:

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

SL #	Course Outcomes
CO1	Acquire knowledge about mathematical logic, set theory and relations
CO2	Use functions, partial ordering and counting techniques to solve problems.
CO3	Understand measures of central tendency, measures of dispersion, probability and its distributions.
CO4	Basic understanding of linear algebra

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔		✔	✔		✔	✔
CO2	✔		✔	✔		✔	✔
CO3	✔		✔	✔		✔	✔
CO4	✔		✔	✔		✔	✔

COURSE CONTENTS

Module 1: Mathematical logic: Propositional and Predicate Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Normal Forms**.** Set and relations: Set Operations, Properties of Relations, Representing relations- matrices & digraphs. Closure of Relations, Composition of relations, Equivalence Relations.

Module 2: Functions: Types of Functions, Composition of Functions and Inverse Functions, Some important functions- floor & ceiling, Recursive functions**.** Partitions - Partial Ordering, Hasse Diagram, Lattice –Types, Properties, Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion-Exclusion Principle.

Module 3: Measures of Central Tendency - Measures of Dispersion - Coefficient of Variation, Covariance. Probability – Random experiment, Sample point, Sample space, Events, Algebra of events, Statistical regularity, Frequency and Classical definitions, Axiomatic approach to probability, Probability Space and probability measure, Addition theorem, Conditional probability, Multiplication theorem, Independence of events, Bayes' theorem and applications. Discrete

Distributions – One point, two point distributions, Uniform, Point binomial, Poisson, Continuous Distributions – Normal, Exponential.

Module 4: Matrices and determinants: matrix, types of matrices, operations on matrices, transpose of a matrix, Determinants-properties of determinants- inverse of a matrix- Rank of a Matrix, Trace of a Matrix. Solving Linear Equations using Matrices – Matrix solution, Gauss Elimination Method, Vector Space, Subspace, Linear Dependence and Independence, Basis and Dimension, Linear Transformations, Matrices Related to Linear Transformations, Eigenvalues and Eigenvectors.

Module X (For Additional Reading and Comprehension by the Students): Mathematical Induction, Recurrence Relation, Generating function, Group Theory: Groups, Subgroups. Discrete Distributions – Geometric, Hyper geometric and Negative binomial distributions. Continuous Distribution - Rectangular, Beta, Gamma, log normal distribution. Consistency of a linear system, Diagonalization of a matrix, Diagonalization of a symmetric matrix.

Core Compulsory Readings

1. Kenneth H. Rosen, Discrete Mathematics and Applications, TMH 2003
2. Elementary Linear Algebra - Devi Prasad (Narosa Pub. House, 2006)
3. Fundamentals of Mathematical Statistics - S. C. Gupta & V. K. Kapoor (Sulthan Chand & Sons)

Core Suggested Readings

1. Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Kamala Krithivasan, McGraw Hill Education, 2011 (Seventh Edition).
2. J. P. Tremblay and R Manohar, Discrete Mathematical Structures with Applications to Computer Science, TMH 2001
3. Discrete Mathematics, N Ch S N Iyengar, V M Chandrasekharan, KA Venkatesh, PS Arunachalam, Vikas Publishing, 2003.
4. Introduction to Probability and Statistics for Engineers and Scientists- S.M. Ross (Elsevier)
5. Linear Algebra - A Geometric Approach - S. Kumaresan (Prentice Hall India)

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Define tautology and contradiction.
2. Show that for any two sets $A-(A\cap B) = A-B$
3. Explain Equivalence relation.
4. Let R be a partial ordering of the set of all devisors of 64. Construct the hasse diagram for it. Find the meet and join.
5. Three dice are rolled together. What is the probability of getting at least one '4'?
6. Use Gauss elimination to solve the following system of linear equations.

$$2X + Y + Z = 10$$
$$3X + 2Y + 3Z = 1$$
$$X + 4Y + 9Z = 16$$

SEMESTER I
CORE COURSE
MCCSA01DSC02 SYSTEM SOFTWARE AND OPERATING SYSTEMS

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4/1	0	5	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

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

Course Objectives:

- To know the design and implementation of assemblers, macro processor, linker, loader and compiler.
- To explain the main components of OS and their working.
- To familiarize the operations performed by OS as a resource Manager.
- To impart various scheduling policies of OS
- To teach the different memory management techniques.
- To explain file system, mass storage structure and input/output management.

Course Outcomes:

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

SL #	Course Outcomes
CO1	Acquire knowledge about Assembler, Linkers and Loaders.
CO2	Understand Process scheduling, process synchronization and methods to handle deadlocks.
CO3	Understand Memory Management and file management techniques.
CO4	Understand I/O systems, mass storage structure and different disk scheduling algorithms.

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔	✔		✔	✔
CO2	✔	✔	✔	✔		✔	✔
CO3	✔	✔	✔	✔		✔	✔
CO4	✔	✔	✔	✔		✔	✔

COURSE CONTENTS

Module 1: 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, Design of Macro pre-processor. **Linkers:** Linking and Relocation concepts, Design of linkers, Self relocating programs. **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.

Module 2: 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. **Process synchronization:** Critical section Problem, Mutual Exclusion, Requirements, Semaphores, Producer Consumer Problem, Readers Writers Problem,**Deadlock** :Prevention, Detection and Recovery.

Module 3: 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 implementation, Directory implementation, Allocation methods.

Module 4: 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.

Module X (For Additional Reading and Comprehension by the Students):
Macros-Advanced Macro facilities, **Linker-**Linking for over-lays, **CPU Scheduling-** Multiple Processor Scheduling, Algorithm Evaluation- Advanced CPU scheduling. **Process synchronization:** Monitors, **File system interface and Implementation**- File system mounting, File sharing, Protection, File system structure, Free space managements, Efficiency and performance, Recovery, Log- structured file system.

Core Compulsory Readings

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.

Core Suggested Readings

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. Sibsankar Haldar, 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)

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Design a two pass assembler with its necessary phases
2. With an example explain FCFS, SJF and Round Robin CPU Scheduling Algorithms
3. Describe Producer Consumer Problem
4. Illustrate banker's algorithm to avoid deadlocks
5. Describe Paging mechanism with examples
6. Implement any three-page replacement algorithms for the reference string given as follows and find the number of page faults
7. Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.
8. Explain various File Access methods involved with OS
9. Explain with a diagram a typical PC Bus structure with I/O Devices
10. Apply and illustrate SSTF, FCFS and SCAN disk scheduling algorithms with a request queue of 98, 183, 37, 122, 14, 124, 65, 67 with Head pointer at 53

SEMESTER I

Core Paper

MCCSA01DSC03 COMPUTER NETWORK AND LINUX ADMINISTRATION

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course provides an insight into the fundamental topics of Computer Networks and Linux Administration. A discussion on inter process communication and programming is provided. Configuration of Linux for network related activities are also discussed.

Course Objectives:

- To understand the basics of Computer Networks
- To acquire knowledge about the fundamentals of using Linux Operating System in a network environment
- To illustrate various Inter Process Communication mechanisms
- To develop programs using various Inter Process Communication primitives
- To learn network configuration basics in Linux

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the basics of Computer Networks and Linux Network Administration
CO2	Illustrate various Inter Process Communication mechanisms
CO3	Develop programs using various Inter Process Communication primitives
CO4	Explain network configuration basics in Linux

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔		✔		✔	✔
CO2	✔	✔	✔	✔		✔	✔
CO3	✔	✔		✔		✔	✔
CO4	✔	✔		✔		✔	✔

COURSE CONTENTS

Module 1: Introduction, Basic concepts - Line configuration, Topology, Transmission mode, Categories of networks, Internetworks, Transmission media - Twisted pair Cable, Coaxial Cable, Optical Fiber. OSI and TCP/IP models, Functions of Layers in OSI and TCP/IP models. Network Devices: Hub, Switch, Router, Bridge, Gateway, Modem, Repeater, Access Point.

Module 2: Introduction to Linux: Linux and Unix, Common Linux Features, Advantage of Linux, Overview of Linux architecture, Linux files system, Linux standard directories. Commands for files and directories cd, ls, cp, rm, mkdir, rmdir, pwd, file, more, less, grep. Creating and viewing files using cat. Common administrative tasks: Obtaining Supervisor privileges, Setting file and directory permissions, Managing links, Users: concept, /etc/passwd file, /etc/shadow file, users groups and umask, adding, deleting and modifying user accounts. Major services in a UNIX system: init, login from terminals, syslog. LILO boot process and GRUB boot process.

Module 3: 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, Shared memory. Sample programs for IPC that uses Pipes, Message Queues, and Shared memory. Socket Programming: Overview, TCP and UDP Sockets, 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 / Server Program using unreserved ports.

.

Module 4: 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: loopback interface, Ethernet interface, The SLIP and PPP interface, Configuring Gateway, Routing through gateway, Network commands: ifconfig, netstat, route. Dynamic IP Configuration: DHCP, Need for DHCP, Functions of DHCP. Monitor Network Connections: ping, traceroute, netstat with options i, r, l, p. Network Applications: File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Network File Systems (NFS), Network Information System(NIS), Hypertext Transfer Protocol (HTTP), Web Server.

Module X (For Additional Reading and Comprehension by the Students):
Transmission media: Satellite, Communication, Cellular Telephony, Terrestrial Microwave. History of Computer Networking and Internet. Core system services: init daemon, /etc/inittab file, telinit command, xinetd and inetd, enabling and disabling services. DHCP Server and Client Configuration. Email Protocols: Simple Mail Transfer Protocol (SMTP), Post office Protocol (POP), Multipurpose Internet Mail Extension (MIME). Domain Name Services (DNS): Working of DNS, Domain Name Servers and Zones, Secondary and primary DNS.

Core Compulsory Readings

1. Behrouz A. Forouzan, Data Communications and networking, Fourth Edition, McGraw Hill 2017
2. James F. Kurose and Keith W. Rose, Computer Networking A Top-Down Approach Featuring the Internet, Third Edition, Pearson Education
3. Wale Soyinka, Linux Administration A Beginner's Guide, Fifth Edition, TMH
4. Linux Administrator STREET SMARTS A Real World Guide to Linux Certification Skills

Core Suggested Readings

1. Andrew S. Tanenbaum, Computer Networks, Fifth Edition, Prentice-Hall 2011
2. William Stallings, Data and Computer Communication, Tenth Edition, Prentice-Hall 2014
3. Evi Nemeth , et al, Linux Administration Hand Book , PHI 2018
4. Nicholas Wells, Linux Installation and Administration, Thomson Vikas 2003
5. Olaf Kirch & Terry Dawson, Linux Network Administrators Guide, O'relly, 2003
6. W Richard Stevens, Unix Network Programming, PHI, 2002

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration, Team Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Write in brief about any one network device
2. State what is meant by LILO boot process
3. Identify how to address a socket
4. State the need for DHCP
5. Differentiate between switch and hub
6. Prepare short notes on managing links in Linux
7. Differentiate: Half Duplex Unix Pipes and Named Pipes
8. Differentiate: FTP and TFTP
9. Explain OSI model
10. Illustrate IPC using UDP sockets
11. Explain interface configuration
12. Explain commands for files and directories

SEMESTER I
CORE COURSE
MCCSA01DSC04 WEB TECHNOLOGY

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: Explain different components and technologies of World Wide Web as a platform. Design and develop websites using fundamental web languages, technologies, and tools. Distinguish between server-side and client-side web technologies. Describe various web technology and application development issues and trend

Course Objectives:

- Explain different components and technologies of World Wide Web as a platform
- Design and develop websites using fundamental web languages, technologies, and tools
- Distinguish between server-side and client-side web technologies
- Describe various web technology and application development issues and trends

Course Outcomes:

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

SL #	Course Outcomes
CO1	Enable students to program for the World Wide Web using HTML and JavaScript
CO2	Create static and dynamic web pages using PHP and MySQL
CO3	Compare web 1.0 and web 2.0 and Familiarize with XML and JSON
CO4	Impart basic knowledge in Content Management System and Progressive Web Apps

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔		✔	✔
CO4	✔	✔	✔	✔		✔	✔

COURSE CONTENTS

Module 1: HTML5: New Elements -Structural Elements, New Form/Input Elements, New Attributes Canvas, Video and Audio, Web Storage. Introduction to JavaScript -Syntax Variables and Data Types - Statements -Operators-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.

Module 2: 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.

Module 3: 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.

Module 4: Content Management System: Introduction, need of CMS, Understanding CMS technologies, Different types of CMS: Portals, Wikis, Blog etc., their features and possible uses. Web services: Introduction, Web service architecture - RPC, SOA, REST, Web service standards – SOAP, WSDL, UDDI. Progressive Web Apps – Introduction, Features, Advantages.

Module X (For Additional Reading and Comprehension by the Students):
HTML5 elements, Geolocation, DHTML, Introduction to Bootstrap and responsive web design basics, Mash-ups: Introduction, Hybrid application development –

Basics, Discover the platforms and frameworks used for hybrid application development.

Core Compulsory Readings

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

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	60
Continuous Evaluation	40
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Explain Form events in JS with example
2. Discuss the different string handling functions in PHP
3. Explain with an example on Generating XML with PHP
4. Explain JSON request and response with example
5. Discuss on how to Choose the Best CMS Platform for Your Website
6. Explain different web service standard

Semester I | Elective MCCSA01DSE (01 – 03) | POOL A

SEMESTER I
ELECTIVE COURSE (DSE)
MCCSA01DSE01 PRINCIPLES OF PROGRAMMING USING PYTHON

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course mainly focuses on introducing the fundamental programming concept to the students. The course mainly discusses the various control structures and data structures that will be useful for the programmers to learn the basic programming concept. Further this course also elaborates the object-oriented programming concepts. Structure of this course is well organized in such a way that it introduces the programing basic concept to advanced concepts such as modules, packages, GUI, NLP and exception handling mechanisms. After completing this course, the students acquire the ability to develop real life applications commonly useful for society in many walks of life.

Course Objectives:

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

Course Outcomes:

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

SL #	Course Outcomes
CO1	Acquire the basic skills in programming through different control structures
CO2	Understand basic data types and attain the knowledge of using functions and modules in python for developing general purpose applications
CO3	Understand and implement object-oriented programming concepts in Python
CO4	Learn the GUI programming in Python for developing easy user front end tools for developing General purpose applications

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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. **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.

Module 2: Lists - Creating List, Basic List Operations, Indexing and Slicing in Lists, Built - In Functions used on lists, list Methods, 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. 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.

Module 3: 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.

Module 4: GUIs in Python: Root Window - **Fonts and colors** – 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.

Module X (For Additional Reading and Comprehension by the Students):
History of Python Programming, Thrust Areas Of Python, Installing Anaconda Python Distribution, PyCharm IDE and Jupyter Notebook, Creating And Running Python Project, The Pickle Module, NumPy and Pandas for data analysis.

Core Compulsory Readings

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

Core Suggested Readings

1. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st Edition,O'Reilly Media, 2017. ISBN – 13: 978-1491962299.

2. Wesley J. Chun, Core Python Programming, Second Edition, Publisher: Prentice Hall Pub

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1.What are the fundamentals data types in Python?
2.What are the different control structures in Python?
3. Explain the function and syntax of for loop control structure in Python with example.
4. What are functions? Explain how it differs from modules?
5.Explain the differences between modules and packages.
6. What are the different string operations in python? Explain.
7. Differentiate between mutable and immutable objects in Python,
8. Explain the basic operations on List.
9. What are dictionaries? Explain its use.

10. What is inheritance?
11. How inheritance is implemented in Python?
12. What is polymorphism? Explain its implementation in Python.
13. What is encapsulation?
14. What are the different types of errors in a program? Explain each one.
15. Explain the differences between compile time and run-time errors in Python.
16. Describe the various GUI widgets in Tkinter module.
17. Explain how exceptions are handled in Python.
18. Give an account in re module in Python.
19. What is assert statement in Python? Give its syntax.
20. What is user-defined exception? Explain with suitable example.
21. What are objects? Explain how it differs from class.
22. What is CSV file?
23. Explain how you will read CSV files in Python.
24. What are Pandas? Explain its usage.
25. Write a note on glob module in Python.

SEMESTER I
ELECTIVE COURSE (DSE)
MCCSA01DSE02 PRINCIPLES OF PROGRAMMING USING CPP

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course mainly focuses on introducing the fundamental programming concept to the students. The course mainly discusses the various control structures and data structures that will be useful for the programmers to learn the basic programming concept. Further this course also elaborates the object-oriented programming concepts. Structure of this course is well organized in such a way that it introduces the basic concepts of programming using the basic C++ programming language. After completing this course, the students acquire the ability to develop real life applications commonly useful for society in many walks of life.

Course Objectives:

- Aims to impart basic programming skills to the learners in a simplest way
- Familiarize participants with the basic structure of C++ programs and its compilation process
- Acquire the knowledge to impart various control structures to implement programming logic
- Teach the fundamental data types, variables, operators, and control structures in C++
- Introduce object-oriented programming (OOP) concepts, including classes, objects, inheritance, polymorphism, and encapsulation.
- Explore memory management and discuss the proper use of pointers and dynamic memory allocation.
- Cover essential input/output operations for user interactions and file handling.
- Encourage problem-solving and algorithmic thinking through practical coding exercises and projects

Course Outcomes:

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

SL #	Course Outcomes
CO1	Attain the basics of programming language through a modern programming language
CO2	Ability design to implement object-oriented programming principles
CO3	Demonstrate a solid understanding of inheritance, exception handling
CO4	Utilize dynamic memory management using the new and delete operators, understanding object copying, copy constructors, and assignment operators

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1: Principles of object-oriented programming; OOP paradigm; Basic concepts of OOP; Benefits; applications. Introduction to C++, Structure of C++ program; Getting started with C++ syntax, Tokens, Keywords, identifiers and constants; Data types, symbolic constants; type compatibility; declaration and dynamic initialization of variables; reference variables. Operators, manipulators; type cast operators; Expressions, implicit conversions; operator precedence; Control structures, recursion

Module 2: Class and Object: Declaring objects – Defining Member Functions – Static Member Variables and Functions – Array of Object, Abstraction mechanism: private, public, constructors, destructors, member data, member functions, inline function, friend functions, virtual function, static members, and references, making an outside function inline; nesting of member functions; private member functions; arrays within a class; memory allocation for objects; static data members; static member functions; arrays of objects; objects as function arguments; friendly functions; returning objects; const member functions; pointer to members; Local classes.

Module 3: Inheritance: Class hierarchy, derived classes, single inheritance, multiple, multilevel, hybrid inheritance, role of virtual base class, constructor and destructor execution, base initialization using derived class constructors. Polymorphism: Binding, Static binding, Dynamic binding, Static polymorphism: Function Overloading, Ambiguity in function overloading, Dynamic polymorphism: Base class pointer, object slicing, late binding, method overriding with virtual functions, pure virtual functions, abstract classes. Operator Overloading: This pointer, applications of this pointer, Operator function, member and non-member operator function, operator overloading, I/O operators. Exception handling: Try, throw, and catch, exceptions and derived classes, function exception declaration, unexpected exceptions, exception when handling exceptions, resource capture and release.

Module 4: Dynamic memory management, new and delete operators, object copying, copy constructor, assignment operator, virtual destructor. Template: template classes, template functions. Namespaces: user defined namespaces, namespaces provided by library. Object Oriented Design, design and programming, role of classes. Files: File Stream Classes – File Modes – Sequential Read/ Write Operations –Binary and ASCII Files – Random Access Operation – Command Line Arguments.

Module X (For Additional Reading and Comprehension by the Students):
Standard Template Library: Fundamental idea about string, iterators, hashes, iostreams and other types.

Core Compulsory Readings

1. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India)
2. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education

Core Suggested Readings

1. Big C++ - Wiley India
2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India)
3. C++ and Object-Oriented Programming – Jana, PHI Learning.
4. Object Oriented Programming with C++ - Rajiv Sahay, Oxford
5. Mastering C++ - Venugopal, McGraw-Hill Education (India)

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	60
Continuous Evaluation	40
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Define Object-Oriented Programming (OOP) and explain how it differs from procedural programming.
2. Discuss the structure of a C++ program. Identify the essential components required to write a valid C++ program.
3. Explain the role of tokens, keywords, identifiers, and constants in C++ programming. Provide examples of each.
4. Discuss type cast operators in C++. When and how should type casting be used?
5. What is a class in C++? Explain the concept of data abstraction and encapsulation in a class.
6. Define member functions and explain how they are declared and defined in a class. Discuss the use of access specifiers.
7. Explain how to declare and work with an array of objects in C++.
8. Differentiate between private and public member data and member functions in a class. Provide examples to demonstrate their usage.
9. How are inline functions implemented in C++? What are the advantages of using inline functions?

10. Discuss the concept of friend functions in C++. How are they used to access private members of a class?
11. Explain the concept of virtual functions and their role in achieving dynamic binding in polymorphism.
12. Define inheritance and explain its importance in object-oriented programming. Provide an example of single inheritance.
13. Discuss the concept of class hierarchy and how derived classes inherit properties from base classes.
14. Explain the differences between single, multiple, multilevel, and hybrid inheritance in C++.
15. Describe the role of a virtual base class in avoiding multiple copies of base class data in derived classes.
16. Discuss the sequence of constructor and destructor execution in a derived class. How is base class initialization performed using derived class constructors?
17. Define polymorphism and explain the concepts of static and dynamic binding.
18. Explain the concept of dynamic polymorphism using base class pointers and method overriding.
19. What is object slicing in C++? How does it occur, and how can it be avoided?
20. Explain the concepts of exception handling in C++. How are try, throw, and catch used to manage exceptions?
21. Describe the process of dynamic memory allocation in C++ using new and delete operators.
22. Explain object copying, copy constructor, and the assignment operator in C++. How do they affect object behaviour?
23. What are namespaces in C++? How do they prevent naming conflicts in large projects?
23. Describe the concept of files in C++. How are file stream classes used for sequential read/write operations?
24. Explain the different file modes used for file handling in C++.
25. Discuss the role of command-line arguments in C++ programs. How can they be accessed and used?

SEMESTER I
ELECTIVE COURSE (DSE)
MCCSA01DSE03 PRINCIPLES OF PROGRAMMING USING C

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course mainly focuses on introducing the fundamental programming concept to the students. The course mainly discusses the various control structures and data structures that will be useful for the programmers to learn the basic programming concept. Further this course also elaborates the object-oriented programming concepts. Structure of this course is well organized in such a way that it introduces the programing basic concept to advanced concepts. After completing this course, the students acquire the ability to develop real life applications commonly useful for society in many walks of life using C programming language.

Course Objectives:

- Aims to impart basic programming skills to the learners in a simplest way
- Impart knowledge on fundamental and advanced data structure concepts
- Acquire the knowledge to impart various control structures to implement programming logic
- Reinforce and expand students' understanding of programming fundamentals through the lens of C
- Introduce advanced programming constructs and techniques to solve complex problems
- Develop students' proficiency in writing modular and efficient C code.
- Enhance problem-solving skills through algorithmic thinking and application of data structures
- Explore real-world applications of C programming and its role in software development

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand the Fundamentals of Programming
CO2	Explore user-defined data structures like arrays, structures and pointers in implementing solutions to problems
CO3	Familiarize the fundamental data types and its uses in programming environment
CO4	Design and Develop Solutions to problems using structured programming constructs such as functions and procedures

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1: Algorithms and Flowcharts: Definitions, Symbols, Program: structure, top- down design, source code, object code, executable file, file extensions. Introduction to C: Importance of C; Structure of C program, Files used in a C program, Compilers, Compiling and executing C programs, variables, constants, Input/output statements in C Features of C, Evolution of C, 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.

Module 2: Structure, Union, and Enumerated Data Type: Introduction, structures and functions, Unions, unions inside structures, Enumerated data type. Files: Introduction to files, using files in C, reading and writing data files. Detecting end of file Operators in C, Type conversion and typecasting. Decision control and Looping statements: Introduction to decision control, Conditional branching

statements, iterative statements, nested loops, break and continue statements, goto statement.

Module 3: Functions: Introduction using functions, Function definition, function declaration, function call, return statement, passing parameters to functions, scope of variables, storage classes, recursive functions. Arrays: Declaration of arrays, accessing the elements of an array, storing values in arrays, Operations on arrays, Passing arrays to functions, two dimensional arrays, operations on two-dimensional arrays, two-dimensional arrays to functions, multidimensional arrays, applications of arrays.

Module 4: Strings and Pointers: Introduction, string taxonomy, operations on strings, Miscellaneous string and character functions, arrays of strings. Pointers: Introduction to pointers, declaring pointer variables, Types of pointers, passing arguments to functions using pointers, Dynamic memory allocation: malloc(), calloc(),free(), realloc(). File Management: Text and binary files, Defining and opening a file, closing a file, input and output operations on file, error handling, random access file. Command line arguments.

Module X (For Additional Reading and Comprehension by the Students):
C Standard Library, Advanced I/O and File Handling, Preprocessor and Macros, Multithreading and Concurrency in C, C and Hardware Interaction, C for Systems Programming

Core Compulsory Readings

1. E. Balaguruswamy, Programming in ANSI C, 7th Edition, Tata McGraw-Hill
2. Computer fundamentals and programming in c, "Reema Thareja", Oxford University, Second edition, 2017.

Core Suggested Readings

1. Brian W. Kernighan and Dennis M. Ritchie, The 'C' Programming Language, Prentice Hall of India.

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

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Define an algorithm. Provide an example of a simple algorithm for finding the factorial of a number.
2. What is the purpose of a flowchart in programming? Draw a flowchart to represent the process of finding the largest among three numbers.
3.Write a simple C program to calculate the area of a rectangle using input/output statements.
4. What are the different data types in C? Provide examples of each.
5. Identify and explain the various control statements in C, such as conditional
6. Explain the concept of a structure in C. Create a C program that uses a structure to store and display information about a student.
7. What is the difference between a structure and a union? When would you use a union inside a structure?
8. Write a C program that writes the multiplication table of a given number to a file.
9. Describe the usage of decision control statements in C. Write a C program to check if a given number is positive, negative, or zero.
10. Implement a nested loop in C to generate the following pattern:
11. Explain the purpose of the "break" and "continue" statements in C. Provide examples for each.
12. Describe the process of passing arguments to functions in C. Write a C function to find the sum of two integers and return the result.

13. Define a function in C. Create a C program with a user-defined function to find the factorial of a given number.
14. What is the scope of a variable in C? Explain the concept of storage classes in C with examples.
15. Write a recursive function in C to calculate the nth Fibonacci number.
16. How do you declare and access elements in a one-dimensional array in C? Provide a code example.
17. Create a C program that performs matrix addition using two-dimensional arrays.
18. Explain the applications of arrays in real-world programming scenarios.
19. Compare and contrast arrays and pointers in C. How are they related, and how are they different?
20. Write a C function to reverse a string using pointers.
21. What are strings in C? Write a C program to count the number of characters in a given string.
22. Explain the concept of pointers in C. Write a C program to swap two numbers using pointers.
23. What is dynamic memory allocation in C? Describe the functions malloc(), calloc(), free(), and realloc().
24. How do you handle errors while working with files in C? Describe error handling techniques in file management.
25. What are command-line arguments in C? Write a C program that accepts command-line arguments and displays them on the screen.

End of Semester I

Semester II

SEMESTER II
CORE COURSE
MCCSA02DSC07 Algorithms and Data Structure

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CA	ESE	Total
3	0	3	3/1	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: This is one of the basic courses in computer science that deals with the representation and storage of data in the most efficient and effective ways. This course's primary objective is to study and analyze different algorithms and its complexities in manipulation and storage of data in computers in a most effective way.

Course Objectives:

- To impart knowledge about various data structures, their representation, and applications.
- To acquaint with various Algorithmic techniques and applications.
- To familiarize the design and analysis of algorithms
- To familiarize the usage of linear and non-linear data structures and its implementation.
- Acquire the knowledge of various searching and sorting techniques.

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

SL #	Course Outcomes
CO1	Acquire knowledge to analyze and design algorithm and its control structures
CO2	Attain the knowledge of solving recurrences and recursion
CO3	To impart knowledge about various data structures, their representation, and applications
CO4	Familiarize with selected linear and nonlinear data structures.

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1:Data Structures: Basic Terminologies – Algorithms: Definition, Pseudo code Representation – Time complexity and space complexity - **Algorithm analysis:** time and space complexity. Growth 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. **Basic technique for design of efficient algorithm:** Brute Force approach, Divide-and-Conquer approach, Branch-and-Bound technique. Greedy approach, Dynamic programming, Backtracking.

Module 2: Data structures: Definition and classification. Linear data structure: **Array**- one dimensional and two-dimensional arrays, representation, operations, polynomial representation with arrays; Searching and sorting operations (Linear search and binary search & Bubble sort): **String:** Representation, operations and pattern matching. **Recursion**: concept and types, implementation.

Module3:Stack: Representation of stack, operations, implementation. **Application of stack:** postfix expression evaluation. ii. conversion of infix to postfix expression. **Queues:** Representation of queue, operations, implementation. **Application of queue:** job scheduling. **Various types of queues**: Circular queue, dequeue and priority queue. **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.

,.

Module 4: Non- linear data structure: Tree- basic terminologies and properties; representation of binary tree, operations on binary tree; type of binary tree. **Tree traversal**: in order, pre order and post order traversals. Binary search tree. Application of tree. **AVL tree:** representations and operations. **Graphs:** Basic Terminologies, Representation, Operations, Traversals – Applications: Shortest

path problem, Topological sorting, Minimum Spanning trees. **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

Module X (For Additional Reading and Comprehension by the Students):
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. B-Tree, Huffman algorithm. Representations and operations of sets: Hash table, linked lists, tree and bit vector.

Core Compulsory Readings

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. Sahni and Mehta, Fundamentals of Data Structures in C++, 2nd Edn, University Press
3. Samanta, Classic Data structures, Second Edition, PH
4. Sahni, Rajasekaran, Fundamentals of Algorithms, 2ndEdn, University Press
5. Sahni, Rajasekaran, Fundamentals of Algorithms, 2ndEdn, University Press
6. M. A. Weis, Data Structures and Algorithm Analysis in C++, Pearson Edu. Asia, 2013

Core Suggested Readings

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Design and Analysis of Computer Algorithms, Addison Wesley.
2. Pallaw, V K, Design and Analysis of Algorithms, Asian Books Private Ltd, 2012
3. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013
4. Oded Goldreich, P, NP and NP- Completeness, Cambridge University Press, 2011.
5. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume- 1
6. Anany Leviton, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson Education.

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

MODE OF TRANSACTION
Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio,

Video, Print.

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. What is data structure?
2. What are the characteristics of an algorithm?
3. Differentiate between time and space complexity of an algorithm?
4. Explain the various techniques for designing an efficient algorithm.
5. Distinguish between linear and non-linear data structure?
6. Explain the different types of arrays and its operations.
7. What is bubble sort?
8. What is binary search?
9. What is recursion? Explain how it is implemented?
10. What is stack? Explain its basic operation.
11. Explain the applications of stack?
12. What is meant by FIFO?
13. What is meant by LIFO?
15. What are priority queues?
16. What are linked list?
17. Explain the different operations on linked list.
18. What is doubly linked list?
19. Explain the applications of queue.
20. What are the different types of lists? Explain the applications of each one.
21. What is BST?
22. Explain quick sort algorithm.
23. What are the different tree traversing algorithms? Explain.
23. Explain selection sort algorithm?
25. Compare and contrast selection sort and insertion algorithm with example.

SEMESTER II
CORE COURSE
MCCSA02DSC08 Programming in Java

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: This course introduces JAVA programming language with object-oriented programming principles. Emphasis is placed on developing skills on implementing classes, inheritance, polymorphism, exception handling and multithreading. This course covers GUI programming using applets and AWT. Particular emphasis is given to Java database connectivity, Servlet programming and creating web applications using JSP.

Course Objective

- Understand object oriented programming
- Use of conditional statements and looping statements to solve problems associated with decision making and repetitions
- Write programs using more advanced JAVA features such as composition of objects, operator overloading, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling etc.
- Improve the problem solving skills
- Get knowledge about the basic concept of writing a program

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

SL #	Course Outcomes
CO1	Use the basic programming constructs of Java and OOP concepts to develop Java programs for a given scenario
CO2	Develop Java programs using threads, exception handling, AWT and applets
CO3	Develop Java programs using database connection for the web applications
CO4	Develop Java programs using servlets and Java Server Pages

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Features of Java, Byte code, JDK, JRE, JVM, Data Types -Type Conversion and Casting, Variables, Operators, Control Statements, Looping Statements, Arrays, Strings, **Object Oriented Programming Concepts** -Class, Objects-Declaring Objects, Access specifier, Static, Nested and Inner Classes, this Keyword, Garbage Collection**,** Methods-method overloading, **Constructors, Inheritance-**Method Overriding, Dynamic Method Dispatch, Abstract Classes, **Interface**- Defining an Interface, Implementing Interfaces.

Module 2: Packages - Declaring a package, Importing Packages, subpackages, **Exception Handling -** Types of Exceptions,try-catch, throw, throws, and finally, **Thread -** Life cycle of a thread, Synchronization, Thread class & Runnable interface, Multithreading. I/O Streams, **File Handling**, **Applets-** Applet life cycle, working with Applets, The <APPLET> tag. Working with Graphics. **Abstract Window Toolkit (AWT)** - Components: Container, Panel, Window, Frame. AWT Controls, Listeners, Layout Managers, Event Handling - Events, Event Sources, Event Classes, Event Listener Interfaces, Adapter Classes. **Swing (JFC) -** Difference between AWT and SWING, Swing components – JApplet, JButton, JFrame, etc.

Module 3: Database connectivity - JDBC architecture, JDBC Drivers, Steps to connect to Database- Connectivity with MySQL, Driver Manager, Types of JDBC statements: Statement, Prepared statement, Callable statement, ResultSet - Types of ResultSet, blobs and clobs, metadata - Database Metadata, Resultset Metadata, transactions, stored procedures, error handling

Module 4: Java Servlets: Introduction to servlet, Servlet life cycle, Developing and Deploying Servlets, Generic and http servlets, GET, POST, HEAD and other requests, Servlet responses, error handling, security, servlet chaining, cookies, session tracking, Working with Apache Tomcat Server, Steps to create a servlet in Tomcat, working of servlet, servlet in Netbeans. **Java Server 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.

Module X (For Additional Reading and Comprehension by the Students)
Collection class, Java's Built-in Exceptions, Inter thread communication, Deadlock, Suspending, Resuming and stopping threads, 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, File Class, Reader and Writer Streams, Random Access File. RMI. JNDI.

Core Compulsory Readings

1. Herbert Schildt, The complete reference Java2 ,11thed, Released December 2018 Publisher(s): McGraw-Hill ISBN: 9781260440249
2. JasonHunder& William Crawford, Java Servlet Programming, O'REILLY, 2002
3. Marty Hall, Larry Brown Core Servlets and Java server pages. Vol 1: Core Technologies. 2nd Edition.

Core Suggested Readings

1. David Flanagan, Java in a Nutshell A desktop quick Reference, 7 Edition, 2018. OReilly & Associates Inc
2. Rajkumar, Java programming, Pearson, 2013
3. Harimohan Pandey, Java Programming, Pearson, 2012
4. David Flanagan, Jim Parley, William Crawford & Kris Magnusson , Java Enterprise in a nutshell- A desktop Quick reference -O'REILLY, 2005
5. Stephen Ausbury and Scott R. Weiner, Developing Java Enterprise Applications, Wiley-2001
6. Database Programming with JDBC and Java, Reese George, Oreilly

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Create a class named 'Student' with string variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as 2 and name as John by creating an object of Student class.
2. Create a class to perform method overloading by changing the number of parameters
3. Design a simple program to print multiples of 5 with one thread and multiples of 10 with another thread using thread synchronization.
4. Create a class with a method to check if a number is less than 0 or not. If the number is less than 0, throw Arithmetic Exception, otherwise, print division is possible
5. Create a simple program to read a text file MyFile.txt line by line using BufferedReader class.
6. Create a registration form using AWT
7. Formulate the different steps to connect to the database in java.
8. Formulate the different steps to store BLOB data type into a table in database
9. Create a simple hello world servlet program by extending HttpServlet class.
10. Create a simple JSP web form to take the input of a student and submit it to a second JSP file which may simply print the values of form submission.

SEMESTER II

Core Paper

MCCSA02DSC09 Database Management Systems

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/0	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: This course is designed to provide a comprehensive understanding of the principles and practices of managing databases. It also focuses on the theoretical and practical applications of DBMS, which play a crucial role in modern information systems.

Course Objectives:

- To introduce students to the basic concepts and the principles of Database Management Systems
- To understand the value of data integrity, security and privacy concepts
- To explore normalization, transaction management, indexing and concurrency control
- To provide a hands-on experience in designing, implementation and management of databases
- To introduce emerging technologies in DBMS field

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

SL #	Course Outcomes
CO1	Familiarization with the concepts of Database Management Systems, Relational Database design and SQL
CO2	Familiarization with transaction processing and concurrency controls in databases
CO3	Explain the fundamentals of NOSQL databases
CO4	Explain the fundamentals of mapreduce

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔			✔	✔	✔
CO4	✔	✔			✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Database Management Concepts: The ER and EER. Relational Algebra and Relational calculus. Functional dependencies -1st, 2nd, 3rd, 4th, BCNF, 5th Normal form. Sub-queries: correlated sub-queries, join, Exist, Any, All, joined relations. Integrity and security: domain constraints, Referential integrity, assertion, triggers. Views: Introduction to views, data independence, security, updates on views, comparison between tables and views.

Module 2: Transaction processing - desirable properties of transaction. Transactions and Schedules – Characterizing Schedules based on Recoverability, Serializability of schedules. Concurrency Control in databases: Locking Techniques-Time stamp ordering, Multi version concurrency Control – Granularity of data items.

Module 3: Overview and History of NoSQL Databases. Definition of the Four Types of NoSQL Databases, The Value of Relational Databases, Getting at Persistent Data, Concurrency, Integration, Impedance Mismatch, The Emergence of NoSQL, Aggregate Data Models; Aggregates, Example of Relations and Aggregates, Consequences of Aggregate Orientation, Application and Integration Databases, Attack of the Clusters.

Module 4: Map-Reduce: Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations, A Two Stage Map-Reduce Example, Incremental Map-Reduce Key-Value Databases, What Is a Key-Value Store, Key-Value Store Features, Consistency, Transactions, Query Features, Structure of Data, Scaling, Suitable Use Cases, Storing Session Information, User Profiles, Preference, Shopping Cart Data, When Not to Use, Relationships among Data, Multioperation Transactions, Query by Data, Operations by Sets.

Module X (For additional reading and comprehension by the students):

NoSQL Key/Value databases using MongoDB, Document Databases, Document oriented Database Features, Consistency, Transactions, Availability, Query Features, Scaling, Suitable Use Cases, Complex Transactions Spanning Different Operations, Queries against Varying Aggregate Structure.

Core Compulsory Readings

1. HSilbersehatz, 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, Fundamentals 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. Sadalage, P. & Fowler, M. (2012). NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. (1st Ed.). Upper Saddle River, NJ: Pearson Education, Inc. ISBN- 13: 978-0321826626 ISBN-10: 0321826620
6. Redmond, E. & Wilson, J. (2012). Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (1st Ed.). Raleigh, NC: The Pragmatic Programmers, LLC. ISBN-13: 978-1934356920 ISBN-10: 1934356921

TEACHING LEARNING STRATEGIES

Lecturing

MODE OF TRANSACTION

Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40

Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Explain the concept of normalization in the context of database design. Why is it important? Give an example.
2. With corresponding examples, explain the difference between a primary key and a foreign key in a relational database. How it can be represented in a table.
3. Discuss the concept of ACID properties in transaction management. How do they ensure data integrity and consistency?
4. Write the advantages and disadvantages of various database indexes.
5. Write an SQL query to retrieve all students who have got more than 75 percentage of mark in the department of science from the student table.
6. Explain the purpose of a trigger in database. Give an example scenario where a trigger can be useful.
7. Discuss how database concurrency control used to ensure the consistency of data in a multi-user environment.
8. Discuss the role of data warehousing in decision support systems. What are the common challenges to be faces while implementing a data warehouse?
9. Compare and contrast distributed database system and centralized database system.
10. What are NoSQL databases? How it is different from traditional databases? Discuss a scenario where NoSQL database is more suitable than the traditional database.

SEMESTER II

Core Paper

MCCSA02DSC10 THEORY OF COMPUTATION

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course deals with the concepts of formal languages, grammar and automata. Topics covered include formal languages like regular language, context free language, context sensitive language and recursive enumerable language and their associated grammars. Students will learn how to construct abstract machines like finite automata, pushdown automata and standard Turing machine.

Course Objectives:

- Give basic idea on theory of computation and its applications.
- Impart knowledge on Finite Automata, regular languages, regular expression and regular grammar.
- Provide understanding on context free languages and pushdown automata
- Generate basic understanding on standard Turing machine and its variations.

Course Outcomes:

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

SL #	Course Outcomes
CO1	Attain knowledge about Finite Automata.
CO2	Understand regular expressions, regular language and regular grammar
CO3	Obtain knowledge on context free grammar and pushdown automata
CO4	Attain knowledge about standard Turning Machine and hierarchy of formal languages

Mapping of COs to PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✔	✔	✔		✔	✔
CO2	✔	✔	✔		✔	✔
CO3	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

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

Module 2: Regular Languages and Regular grammars: Regular expressions, connection between regular expressions and regular languages- state elimination method, Arden's lemma, regular grammars. Properties of Regular Languages: closure properties of regular languages, identifying non regular Language

Module 3: Context-free grammars & languages: Context-free grammars, parsing and ambiguity. Simplification of Context free Grammars and Normal forms : methods of transforming grammars, two important normal forms – CNF & GNF Pushdown automata for context-free languages Non deterministic pushdown automata, PDA and context-free languages, deterministic pushdown automata and deterministic context-free languages. Properties of Context-Free Languages: pumping lemmas for context free languages, closure properties for context-free languages.

Module 4: Turing machine: Standard Turing machine, combining Turing machines for complicated tasks, Turing's thesis. Other models of Turing machine : Minor variations on the Turing machine theme, Turing machine with
complex storage, nondeterministic Turing machine, a universal Turing machine, Linear bounded automata.

A Hierarchy of Formal Languages and Automata: Recursive and Recursively Enumerable Languages, Unrestricted Grammars, Context-Sensitive Grammars and Languages, The Chomsky Hierarchy

Module X (For Additional Reading and Comprehension by the Students):
Mathematical preliminaries and notation, Proof techniques -Pumping lemma for linear languages-Limits of Algorithmic computation: Problems that cannot be solved by Turing machines, Undecidable Problems for Recursively enumerable Languages, The Post Correspondence problem-Turing Machine Models and Complexity: Language Families and Complexity Classes, The Complexity Classes P and NP-Finite State Transducer – Mealy Machines and Moore Machines

Core Compulsory Readings

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

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning.

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar…..	40

Sample Questions to test Outcomes.

1. What are formal languages? Give example.
2. Construct a minimal FA that accepts all the strings of a's and b's where every string ends with ab. ∑={a,b}
3. Explain the process of NFA to DFA conversion with the help of an example.
4. Construct a regular expression that generate all the strings of a's and b's where number of a's in the string is even.
5. Construct a grammar that generates all the strings of a's and b's where every string contain at most 2a's.
6. What is Chomsky Normal Form? Give example.
7. Explain the architecture of PDA.
8. Define Turing Machine. Explain its architecture.

SEMESTER II
CORE COURSE
MCCSA02DSC11 Software Engineering

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

- To introduce the basic concepts of software engineering
- To introduce various software process models
- To build an understanding on various processes of software development

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

SL #	Course Outcomes
CO1	Identify the apt process model to build a software project
CO2	Illustrate the gathering of requirements and translating them into an appropriate model
CO3	Illustrate the principles of various aspects of software design
CO4	Identify the testing strategies suitable for a project and illustrate various methods for project estimation

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Fundamentals of Software Engineering: Evolution, Software Crisis, Definition, Layered Technology, Software Process, Framework Activities, Kinds of Process Flows. Process Models: Waterfall Model, V Model, Incremental Process Model, Evolutionary Process Models, Concurrent Model, Concepts of Agile Development, Agile Process Models: Extreme Programming (XP), Adaptive Software Development, Dynamic system Development Method(DSDM), Scrum

Module 2: Requirements Engineering: Tasks, Requirements Elicitation: Collaborative Requirement Gathering, QFD, Usage Scenarios. UML Fundamentals. Developing Use Cases, Developing Use Case Diagrams. Requirements Modeling: Elements, Scenario based Modelling, Data Modeling, Developing ER Diagrams, Fundamentals of DFD, Flow Oriented Modeling

Module 3: Design within the Context of Software Engineering, Design Concepts: Abstraction, Architecture, Patterns, Separation of Concerns, Modularity, Information Hiding, Functional Independence, Refinement, Refactoring. Design Model. Component-Level Design: Software Component, Different Views of a Component, Designing Class- Based Components. User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface analysis, Interface design steps

Module 4: Software Project Estimation: Estimation based on LOC, FP, Process and Use Cases, COCOMO II Model, Estimation for Web Applications. Scheduling: Time-line Charts (GANTT Charts), Scheduling for Web App Projects. Software Testing: Unit Testing, Integration Testing, Regression Testing, Smoke Testing.

Module X (For Additional Reading and Comprehension by the Students): Specialized Process Models, Agile Process Models: Crystal, Feature Driven Development (FDD). Requirement Modeling: Requirements Modeling for Web Apps. Design: Web App Interface Design. Design Evaluation. Project Evaluation:

Design for Object Oriented Projects and Agile Development. Project Estimation: Web App Projects and Agile Development. Testing Strategies for Object Oriented Software and Web App Software, Validation Testing, System Testing.

Core Compulsory Readings

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

Core Suggested Readings

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Identity the framework activities of a software process
2. Identity the elements UML diagram
3. State the golden rules of interface design
4. State the goals of software project estimation
5. Differentiate between incremental process model and evolutionary process model

6. Prepare the use case for a sample project work (Assume a suitable project work and a sample event involving an actor)
7. Prepare short note on any two design concepts
8. Illustrate LOC based estimation with a suitable example
9. Explain various process models
10. Explain the creation of flow oriented model with a suitable example
11. Explain the creation of data model with a suitable example
12. Illustrate any one golden rules of interface design

Semester II | AEC |MCCSA02AEC (01 – 06)

SEMESTER II
AEC
MCCSA02AEC01 Algorithm Analysis

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

- To write the performance of algorithms in mathematical terms
- To acquire knowledge for analyzing a given algorithm.

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

SL #	Course Outcomes
CO1	Understand about Time and Space Complexity
CO2	Acquire knowledge about the structure of the algorithm.
CO3	Obtain knowledge to analyze algorithm control structures and solving recurrence.
CO4	Attain information about Complexity Classes

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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)

Module 2: 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

Module 3: 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

Module 4: 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.

Core Compulsory Readings

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.

Core Suggested Readings

1. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013
2. Upadhyay, N, Design and Analysis of Algorithms, Sk Kataria & Sons, 2008.

3. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India
5. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill
6. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd
7. Oded Goldreich, P,NP and NP- Completeness, Cambridge University Press, 2011.
8. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms,Volume- 1, Addison Wesley, 1997.
9. Sanjeev Arora and Boaz Borak, Computational Complexity- A Modern Approach, Cambridge University Press; 2009

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	60
Continuous Evaluation	40
Tests	40
Assignment	20
Seminar	40

SEMESTER II
AEC
MCCSA02AEC02 Artificial Intelligence and Daily Life

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

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

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

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

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Artificial Intelligence- History of AI- Advantages and Disadvantages of AI- Applications- AI domains.AI in real life, Expert system - Expert system development- Modern expert systems

Module 2: Search and Control Strategies- State- Space representation- Problem Solving - Heuristic Techniques – Hill Climbing – Simulated Annealing

Module 3: Machine Learning- Supervised and Unsupervised Algorithms- Neural Networks- Classification and Predictions model – Applications

Module 4: Natural Language Processing - Natural Language Processing Tasks - NLP Applications- Recommender System - Sentimental Analysis

Core Compulsory Readings

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach Third Edition Pearson Education 2010 Inc. ISBN: 978-0-13-604259-5.
2. D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990

Core Suggested Readings

1. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill.
2.https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER II
AEC
MCCSA02AEC03 Python Programming

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description:

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

Course Objectives:

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

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

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

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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

Module 2: 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

Module 3: Lists- Creating List, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions used on lists, list Methods. **Dictionaries**- Creating Dictionary,

Accessing and Modifying *key:value* Pairs in Dictionaries, Built-In Functions used on Dictionaries, Dictionary Methods. **Tuples and Sets-** Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions used on Tuples, Relation between Tuples and Lists, Sets and Set Methods, Frozenset. **Strings-** Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings

Module 4: Files- Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary files, Reading and Writing CSV Files, Introduction to NumPy and Pandas.

GUIs in Python: Root Window-Fonts and colors- Working with containers and canvas, Frames, Widgets, Button widgets, Arranging widgets in the Frame, Label Widget, Message Widget, Text widget, Scrollbar widgets, Check button widget, Radio button widget, Entry Widget, Spin box Widget, List Box Widget, Menu Widget

Core Compulsory Readings

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

Core Suggested Readings

1. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st Edition,O'Reilly Media, 2017. ISBN – 13: 978-1491962299.

2. Wesley J. Chun, Core Python Programming, Second Edition, Publisher: Prentice Hall Pub

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1.What are the fundamentals data types in Python?

2.What are the different control structures in Python?

3. Explain the function and syntax of for loop control structure in Python with example.

4. What are functions? Explain how it differ from modules?

5.Explain the differences between modules and packages.

6. What are the different string operations in python? Explain.

7. Differentiate between mutable and immutable objects in Python,

8. Explain the basic operations on List.

9. What are the different types errors in a program? Explain each one.

10. What is CSV file?

11. Explain how will you read CSV files in Python.

12. What are Pandas? Explain its usage.
13. What are GUIs?
14. Explain the different Widgets in Python.
15. What is Canvas? Explain how it differ from Frames.

SEMESTER II
AEC
MCCSA02AEC04 Data Processing with Python

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

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

Course Objectives:

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

Course Outcomes:

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

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

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Python Fundamentals: Basic Concepts, Naming Variables, Operators and operands, Expressions, importing libraries, Redirecting the output, Data types, Lists, Working with files, Working with Directories; Big Data vs Data Science

Module 2: Introduction to probability: discrete and continuous variables, probability distribution, Bayes Rule, Variance, Standard Deviation, Types of probability distributions, Algorithms designed using probability,

Module 3: Finding datasets, Jupiter notebooks and loading data, pandas vs numpy, Saving, Dropping Null Values, User Defined Functions, Cleaning Dataset, Graphs and Statistics, Histogram, Working with rows and columns.

Module 4: Introduction to basic plots, pandas vs matplotlib, visualizing 1D distributions, visualizing 2D distributions, Higher dimension visualizations.

Core Compulsory Readings

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

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio,

Video, Print.

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions

1. What skills are necessary for a Data Scientist?
   a) Statistics
   b) Deep Learning
   c) Linear Algebra
   d) All of the above
2. CLI stands for …….
   a) Command Line Interface
   b) Command Language Interface
   c) Command Line intercom
   d) None of the Above

3. Give one example where both false positives and false negatives are important equally?
4. Write the difference between data analytics and data science.
5. What are data science tools?
6. List different data types available in python with examples for each.

SEMESTER II
AEC
MCCSA02AEC05 Fundamentals of Big Data

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: This course mainly focuses to introduce the fundamentals of big data analytics concept using Python programming language. First part of this course familiarizes basic concepts of python programming language and subsequently introduces the big data analytics concepts and machine learning algorithms for developing mathematical/statistical models for analyzing big data using NumPy and Pandas in Python. After completing this course, the students acquire the basic ability to develop real life applications for extracting hidden patterns/trends in a big data environment that is commonly useful for the society in may walk of life.

Course Objectives:

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

Course Outcomes

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

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

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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,

Module 2: 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

Module 3: Files: - Types of Files: -Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary files, Reading and Writing CSV Files, Introduction to NumPy and Pandas.

Data Analytics Overview - Statistical Computing - Mathematical Computing Using NumPy - Data Processing with Pandas - Data Visualization with Python - Introduction to Model Building for Evaluation

Module 4: Supervised Learning - Classification, Naive Bayes, KNN, Linear Regression. Unsupervised Learning - Clustering, Hierarchical algorithms – Agglomerative algorithm, Partitional algorithms -K- Means. Association Rule Mining - Apriori algorithm.

Core Compulsory Readings

1. Bart Baesens," Analytics in a Big Data World: The Essential Guide to Data Science and its Business Intelligence and Analytic Trends", John Wiley & Sons, 2013.

2. David Dietrich, "EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", John Wiley & Sons, 2015.

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

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1.What are the fundamentals data types in Python?

2.What are the different control structures in Python?

3. Explain the function and syntax of for loop control structure in Python with example.

4. What are functions? Explain how it differ from modules?

5.Explain the differences between modules and packages.

6. What are the different string operations in python? Explain.

7. What is CSV file?

8. Explain how will you read CSV files in Python.

9.What are Pandas? Explain its usage.

10. What is NumPy?

11. Explain the use of pandas in Big data analytics.

12.What is Big Data?

13. What are the different tools in Python for data visualization? Explain.

14. What is KNN?

15. Explain about supervised and unsupervised learning with suitable examples.

SEMESTER II
AEC
MCCSA02AEC06 Fundamentals of Programming Methodology

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

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

Course Outcomes

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

SL #	Course Outcomes
CO1	Identify the historical evolution of Computer Science in the perspective of problem solving
CO2	Illustrate the foundations of developing solutions using flowcharts and algorithms
CO3	Develop solutions using various selection constructs and implement them in C programming language
CO4	Develop solutions using various iteration constructs implement them in C programming language

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: History and evolution of Computer Science in the perspective of problem solving - Problem Solving – Fundamental Constructs for Building Solutions. Flowchart - System Chart - Program Charts – Variables, data names, programming statements – Steps for Developing Solutions - Preparing Solutions using Flowcharts: Conventions - Structure - Symbols. Preparing Solutions using Algorithms - Conventions - Top-Down Design – Structure of Algorithms. Case Study: Developing flowcharts and algorithms for problems using sequence construct

Module 2: Program: Characteristics - Modular Approach - Style - Documentation and Maintenance - Compilers and Interpreters - Preparing, Running and Debugging Programs - Types of Errors. Fundamentals of C Language: Evolution and Features - Program Structure - Elements - Constructs. Character Set, Tokens,

Keywords, Identifier. Data Types, Constants, Symbolic Constants, Variables, Data Input and Output, Statements - Assignment statements. Operators in C: arithmetic, relational, logical, assignment, auto increment, auto decrement, conditional, comma operators. Precedence of operators - expressions – evaluation of expressions, type conversion in expressions – precedence and associativity. Case Study: Preparing C programs for solutions developed in Unit I.

Module 3: Selection Constructs: Simple if - if else - if else if ladder - switch. Branching statements: break, go-to. Case study: Developing solutions (flowcharts and algorithms) for problems using various selection constructs - Comparative Study of various Selection Constructs - Converting a solution using one selection construct with other selection constructs.

Module 4: Iteration Constructs: Top Tested Vs Bottom Tested - while - for - do while - Nesting of loops - skipping breaking loops. Arrays - 1D and 2D - Case study: Developing solutions (flowcharts and algorithms) for problems using various iteration constructs - Comparative Study of various iteration constructs - Converting a solution using one iteration construct with other iteration constructs.

Core Compulsory Readings

1. J.B Dixit, Computer Fundamentals and Programming in C, Firewall Media

2. Anil Bikas Chaudhuri, The Art Of Programming Through Flowcharts Algorithms, Laxmi Publications, New Delhi.

3. Maureen Spraknle and Jim Hubbard, Problem Solving and Programming Concepts, Pearson

4. E Balagruswamy, Programming in ANSI C, TMH, 5th Edition

Core Supplementary Reading

1. R G Dromey, How to Solve by Computer, Pearson Education, 5th Edition

2. Brian W. Kernighan and Dennis M. Ritchie, C Programming Language, PHI

Kanetkar, Let Us C, BPB Publications, 8th Edition

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

Semester II | SEC |MCCSA02SEC (01 – 10)

SEMESTER II
SEC

MCCSA02SEC01 Basics of Machine Learning

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Machine Learning is the study of how to build computer systems that learn from experience. It is a subfield of Artificial Intelligence and intersects with statistics, cognitive science, information theory, optimization, and probability theory. The course will explain how to build systems that learn and adapt using examples from real-world applications. The main topics include neural networks, decision trees, KNN, unsupervised learning, and reinforcement learning.

Course Objectives:

- To understand the principles and techniques in Machine Learning.
- Helps to understand statistical techniques and different algorithms in machine learning.
- Produce Python code to statistically analyze a dataset using classification and regression techniques.
- Compare and contrast evaluation metrics

Learning Outcomes

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

SL #	Course Outcomes
CO1	To understand the basic concepts in machine as well as python programming language
CO2	To familiarize statistical methods and how to apply the same in machine learning
CO3	To learn different types of machine learning algorithms
CO4	To experience the implementation of machine learning algorithms and calculation of accuracy.

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Introduction to Machine Learning; Components of Learning; Categories of Machine Learning-Supervised Learning, Unsupervised Learning and Reinforcement Learning; Introduction to Python: NumPy, Matplotlib, Pandas

Module 2:

Statistical Techniques-Review on probability, Regression, Classification, Clustering, Probability Theories, Decision Trees; IDEs, Dimensionality Reduction; Main challenges of Machine Learning, Deep Learning

Module 3:

Decision Tree Classifiers, Classification and Regression trees; Regression-Linear Regression, Logistic Regression; Neural Networks-Introduction, Perception; Support Vector Machines; K Nearest Neighbors; Introduction to Clustering-K Means Clustering

Module 4:

Machine Learning Steps: Collection of data, Preparation, Model Selection, Training, Evaluation, Parameter Tuning, Making Predictions; Implementation; Confusion Matrix, Accuracy, Precision, Recall, ROC curve and AUC.

Core Compulsory Readings

1. Steven S. Skiena, "The Data Science Design Manual", Springer 2017.

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

Sample Questions

1. Machine Learning is a subset of which of the following:
   a) Artificial Intelligence
   b) Data Learning
   c) Deep Learning
   d) None of the Above
2. Decision Tree cannot be used for clustering. Choose whether true or false
   a) True
   b) False
3. Which technique is used to predict categorical responses?
4. List any four classification algorithm.
5. How can outlier values be treated?
6. Define ensemble learning.
7. What are the assumptions required for a linear regression?
8. List the differences between supervised and unsupervised learning.
9. What is deep learning? What is the difference between deep learning and machine learning?
10. List some of the importance of dimensionality reduction.
11. Explain Neural Network fundamentals.

SEMESTER II
SEC
MCCSA02SEC02 Data Science Fundamentals

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

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

Learning Outcomes

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

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

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Introduction to Data Science: Definition, Big data, populations and samples, exploratory data analysis, properties of data, Mathematical and Statistical Skills, Data Analysis Tools, Data Science Applications

Module 2:

Types of data, Application areas of Data Science, Data Science process, Machine Learning: Introduction, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Introduction to Statistics.

Module 3:

Basic Machine Learning Algorithms: Linear Regression, SVM, Naïve Bayes, example program implementation, Data acquisition, Transformation into standardized format, Data Cleaning, Data Reduction, Data Integration, Data Transformation, Normalization.

Module 4:

Algorithm design and analysis, Data, Database Table, Python, Confusion Matrix, Data Handling and Visualization, Different Chart types, Plotting Functions, Solving problem with Data Science.

Core Compulsory Readings

Text Book

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

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%

Seminar	40%
Total	**100%**

Sample Questions:

1. Identify the language which is used in data science.
   a) R
   b) C++
   c) Java
   d) Ruby
2. Total groups in which data can be characterized is:
   a) 4
   b) 2
   c) 1
   d) 3
3. What are some of the techniques used for sampling?
4. Explain the major Components of Data Science.
5. What is Data Science? List the differences between supervised and unsupervised learning.
6. List some of the applications of data science in the real world scenario.
7. Write the difference between a box plot and a histogram.

SEMESTER II
SEC
MCCSA02SEC03 OPTIMIZATION TECHNIQUES

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

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

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

SL #	Course Outcomes
CO1	Understand the importance of optimization technique
CO2	Discuss Optimum design concepts
CO3	Solve the Linear Programming models using graphical and simplex methods
CO4	Evaluate different algorithmic methods for solving constrained and unconstrained optimization problems

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Optimization: Introduction, Statement of an Optimization problem, formulation of Optimal Problem, Types of Optimization problem.

Module 2:

Optimum design concepts: Definition of Global and Local optima, Optimality criteria, Convexity and concavity of functions of one and two variables, Lagrangian function, Hessian matrix formulation

Module 3:

Linear programming: Standard form of Linear Programming Problem, Canonical form, Elementary operations, Graphical method for two variable optimization problem, Simplex method, Karmarkar's projective scaling method.

Module 4:

Optimization algorithms for solving unconstrained optimization problems – Gradient based method: Cauchy's steepest descent method, Newton's method, Conjugate gradient method. Optimization algorithms for solving constrained optimization problems– direct methods – penalty function methods – steepest descent method.

Core Compulsory Readings

1. G. Hadley, Linear programming, Narosa Publishing House, New Delhi, ISBN

13: 9788185015910.

2. Singiresu S. Rao, Engineering Optimization: Theory and Practice by John Wiley and Sons, (5th edition),ISBN: 978-1-119-55479-3

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

SEMESTER II
SEC
MCCSA02SEC04 Scientific Computing

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course on Scientific Computing is designed to provide students with a comprehensive introduction to computational techniques used in various scientific disciplines. Students will gain practical knowledge of programming languages commonly used in scientific computing, numerical methods, data processing, and high-performance computing. The course will emphasize hands-on experience through programming exercises and projects, enabling students to apply the acquired skills to solve real-world scientific problems.

Course Objectives:

- Introduce students to the fundamentals of scientific computing and its significance in diverse scientific domains.
- Familiarize students with programming languages and libraries commonly used for scientific computations, such as Python, Julia, or MATLAB.
- Equip students with essential numerical analysis techniques and optimization methods for solving scientific problems efficiently and accurately.
- Develop students' skills in data processing, analysis, and visualization to gain insights from scientific datasets.

Learning Outcomes:

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

SL #	Course Outcomes
CO1	Demonstrate a solid understanding of scientific computing principles, methodologies, and applications.
CO2	Choose an appropriate programming language and effectively implement numerical methods to solve scientific problems.
CO3	Analyze and interpret scientific data using statistical techniques and visualization tools.
CO4	Apply numerical techniques to solve linear and non-linear equations, eigenvalue problems, and differential equations.

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Introduction to Scientific Computing. Overview of Scientific Computing: Role, significance, and applications in various scientific disciplines. Essential programming concepts: Variables, data types, loops, conditionals, functions, and basic I/O. Introduction to a programming language (Python recommended): Syntax, data structures, and libraries for scientific computing. Numerical methods: Root finding, interpolation, integration, and differentiation. Data visualization: Plotting techniques and tools for presenting scientific data effectively.

Module 2:

Data handling and manipulation using libraries such as NumPy, pandas, or equivalent. Statistical analysis of data, including hypothesis testing and regression. Advanced data visualization techniques for scientific presentations. Time-series analysis and Fourier transforms for signal processing. Data cleaning and preprocessing for scientific datasets.

Module 3:

Numerical Linear Algebra. Matrix and vector operations: Addition, subtraction, multiplication, and division. Solving linear systems: Gaussian elimination, LU decomposition, and iterative methods (Jacobi, Gauss-Seidel). Eigenvalue and eigenvector computation.

Module 4:

Differential Equations and Optimization. Ordinary Differential Equations (ODEs): First-order and higher-order ODEs, initial value problems, and boundary value problems. Numerical integration methods: Euler's method, Runge-Kutta methods (finite difference, finite element, etc.), and applications.

Core Compulsory Readings

1. Sastry S.S, INTRODUCTORY METHODS OF NUMERICAL ANALYSIS, 5TH EDN,Prentice Hall India Learning Private Limited, 2012, ISBN: 978-8120345928
2. Germund Dahlquist, Ake Bjorck, Numerical Methods in Scientific Computing, SIAM, 2008, ISBN:9780898716443

Core Suggested Readings

1. Bertil Gustafsson, Fundamentals of Scientific Computing,Springer Science & Business Media, 2011, ISBN: 9783642194948

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

SEMESTER II
SEC
MCCSA02SEC05 The Art of E - Documentation using Latex

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

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

Learning Outcomes:

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

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

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: WYSIWYG Editors Vs Latex. Latex: History - Advantages - Editors (Windows / Mac / Ubuntu - Online). Anatomy of Latex Document - Common Document Classes - Paper sizes - Commonly Used Packages - Setting margins. Document Creation - Beginning Document - Sections - Adding Text - Inserting blank lines - Ending Document

Module 2: Typesetting Document - Fonts, symbols, indenting, paragraphs, line spacing, word spacing, titles and subtitles, adding colors to text and entire page, adding bullets and numbered items. Adding header and footer, changing the page orientation, dividing the document into multiple columns

Module 3: Creating Tables: Setting Columns - Merging Rows and Columns -

Various Styles and Orientation. Adding Images - Graphics Packages - Rotation - Scaling. Drawing Matrices - Adding Mathematical Symbols and Equations. Adding Bibliography, Cross References and Citations - Adding Table of Contents, Figures and Tables

Module 4: Beamer - Creating Slides - Adding frames - Dividing the slide into multiple columns - Adding different blocks - Adding Tables and Graphics. Case Study: Preparation of Resume, Official Letters, Book, Article, Homework assignment

Core Compulsory Readings

1. Free online introduction to LaTeX Available at https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)
2. Learning Latex, D.F.Griffits, D.J.Higham, Siam, Philadelpha, 1997
3. Latex, https://en.wikibooks.org/wiki/LaTeX

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

SEMESTER II
SEC
MCCSA02SEC06 Fundamentals of Digital Skilling using Google Workspace for Education

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

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

Learning Outcomes:

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

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

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Google Workplace for Education: Gmail, Google Calendar, Google Drive, Google Docs, Google Sheets, Google Slides. Gmail - Creation of ID - Composing Messages - Folders - Labels - Simple Customizations.

Module 2:. Google Drive: Uses - Accessing - Viewing Drive Contents - Creating Folders - Creating Documents - Making Copies - Uploading Documents - Downloading Folder Contents - Sharing Folders - Erasing Documents - Creating and Managing Workspaces - Searching Contents - Managing Storage.

Module 3: Google Doc: Creation - Saving - Typesetting and Formatting - Images - Tables - Charts - Line - Emoji - Smart Chips - Dropdown - Footnote - Header and Footer - Page Numbers - Indentation - Lists - Checklists - Creating Links -

Watermark - Page Orientation - Spelling and Grammar Check - Citations - Modes (View / Review / Edit) - Voice Typing - Print - Download - Sharing.

Module 3: Google Sheet: Creation - Components - Cells and Addressing - Navigating. Entering and Editing Data: Text - Equations - Typesetting and Formatting. Saving Worksheet. Editing Worksheet Data - Moving and copying data - Moving and copying equations - Inserting and deleting ranges, rows, and columns. Formatting: Text - Numbers - Row and Column -- Conditional Format - Borders and Colors. Worksheets: Naming, Copying, Creating, Removing Charts: Types - Pie Chart - Bar Chart. Data - Sorting - Filters - Validation - Removal of Duplicates. - Page Orientation - Spelling and Grammar Check - Print - Download - Sharing.

Module 4: Google Slide: - Creating Slides - Adding Image, Text Box, Audio, Video, Shape, Chart, Diagram. Typesetting and Formatting - Lists - Checkboxes. Slides: Ordering - Changing Background - Deleting - Transition. Share - Print - Download

Core Compulsory Readings

1. Learn the basics, https://workspace.google.com/intl/en_in/training/

Core Suggested Readings

1. Online materials related to Google Workplace for Education
2. Digital Skilling Course (NPTEL), https://drive.google.com/file/d/1eKZP5LchZqrf3Ba9MjjeMDUh7cjbDOR9/view?usp=sharing

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

SEMESTER II
SEC
MCCSA02SEC07 Image Processing using Python

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: This comprehensive four-module course on " Image Processing using Python " is designed to provide students with a deep understanding and practical skills in image manipulation, enhancement, and analysis. Through hands-on exercises and real-world applications, students will navigate the powerful landscape of Python libraries, including PIL and OpenCV, to unlock the potential of images for various domains.

Course Objectives:

- Introduce students to Python libraries for image processing, emphasizing PIL and OpenCV.
- Provide hands-on experience in basic image manipulation and enhancement techniques.
- Master basic and advanced image filtering techniques for enhanced image quality.
- Build an image classification model using TensorFlow for categorizing images.
- Understand image transformation techniques, focusing on Fourier Transform.
- Provide an introduction to deep learning for image processing.

Learning Outcomes:

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

SL #	Course Outcomes
CO1	proficiency in using Python libraries for image processing
CO2	Comprehensive understanding of image histograms, image smoothing, sharpening techniques, and various filtering methods
CO3	Proficiency in thresholding and region-based segmentation techniques.
CO4	Ability to complete a real-world image processing project, highlighting students' ability to solve practical problems in diverse scenario

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Introduction to Python libraries for image processing, Basic image manipulation and enhancement techniques. Introduction to Python for Image Processing. Image representation and basic operations using Python libraries (e.g., PIL or OpenCV). Reading and displaying images in Python. Basic image processing operations: resizing, cropping, and rotating.

Module 2:

Understanding image histograms, applying image smoothing and sharpening techniques, understanding and applying basic and advanced image filtering techniques. Image restoration techniques, Edge detection methods. Feature

extraction and representation. Practical applications of segmentation and feature extraction. Object detection and recognition.

Module 3:

Thresholding and region-based segmentation techniques. Image classification model with TensorFlow. Preprocessing, Segmentation and Registration of medical images. Understanding 3D image processing, image visualization and manipulation.

Module 4:

Image compression technique, JPEG and Wavelet-based compression technique. Introduction to image steganography, hiding data and extracting hidden data from images using Python. Image transformation techniques (e.g., Fourier Transform). Introduction to deep learning for image processing.

Core Compulsory Readings

1. "Python Imaging Library (PIL) Handbook" by Fredrik Lundh
2. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
3. "OpenCV with Python By Example" by Prateek Joshi
4. "Hands-On Image Processing with Python" by Sandipan Dey

Core Suggested Readings

1. "Image Processing in Python with OpenCV" by Gabriel Garrido Calvo
2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili
3. "Image Processing and Acquisition using Python" by Ravishankar Chityala, Sridevi Pudipeddi, and G.V. Sridhar
4. "Introduction to Deep Learning" by Andreas C. Müller and Sarah Guido

TEACHING LEARNING STRATEGIES

●Lecturing, Demonstration

MODE OF TRANSACTION

●Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

Sample Questions:

1. Define an image histogram and explain how it is useful in image processing. Provide a simple example.
2. Explain the steps involved in building a basic image classification model using TensorFlow.
3. Explain the basic concept of Fourier Transform in image transformation. How does it alter the representation of an image?
4. What is the fundamental idea behind deep learning in image processing?
5. Provide a basic example of how deep learning can enhance traditional image processing methods.

SEMESTER II
SEC
MCCSA02SEC08 Fundamentals of Electrical and Electronics Engineering

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

The "Fundamentals of Electrical and Electronics" course is a comprehensive skill development program designed to introduce participants to the core principles of electrical and electronic engineering. The course aims to provide a solid foundation in electrical concepts, electronic devices, electrical machines, and power systems. Through a combination of theoretical knowledge and hands-on practical exercises, participants will gain the necessary skills to analyze, design, and troubleshoot basic electrical and electronic circuits. The course will also cover safety measures and regulations to ensure a safe working environment.

Course Objectives:

- Understand Basic Electrical Concepts: The course aims to familiarize participants with the fundamental principles of electricity, including voltage, current, resistance, and power, enabling them to analyze simple electrical circuits.
- Explore Electronic Devices and Applications: Participants will learn about semiconductor devices such as diodes and transistors and their applications in rectification, amplification, and digital logic circuits.
- Acquire Knowledge of Electrical Machines and Power Systems: The course will introduce different types of electrical machines, transformers, and power transmission systems, enabling participants to comprehend their working principles and applications.

- Develop Circuit Design and Troubleshooting Skills: Participants will be exposed to circuit design fundamentals, schematic creation, and PCB layout using software tools. Additionally, they will learn troubleshooting techniques using multimeters and oscilloscopes to identify and fix common circuit issues.

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

SL #	Course Outcomes
CO1	Upon completion of this course, participants will be able to explain the basic concepts of electricity, including voltage, current, resistance, and power, and apply Ohm's Law to calculate these values in simple circuits.
CO2	Participants will be able to identify and describe various electronic components and understand their applications in practical circuits, such as rectifiers, amplifiers, and digital logic circuits.
CO3	After completing the course, participants will be able to recognize different types of electrical machines, including DC motors, AC induction motors, and transformers, and comprehend their roles in various industrial and domestic applications.
CO4	By the end of the course, participants will have acquired the skills to design basic electrical and electronic circuits, create schematics and PCB layouts using software tools, and troubleshoot common circuit issues using multimeters

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Electrical Concepts

Basics of Electricity: Introduction to electrons, protons, and atomic structure. Understanding electric charge, current, and voltage. Differentiating between

AC and DC currents. 1.2 Circuit Components and Symbols: Identify and describe passive components (resistors, capacitors, inductors). Recognize and explain active components (diodes, transistors). Introduction to circuit symbols and diagrams. Ohm's Law and Power: Understanding Ohm's Law and its applications. Calculating voltage, current, and resistance in circuits. Power calculations and power rating of components. 1.4 Circuit Analysis Techniques: Series and parallel circuits analysis. Kirchhoff's laws and their application. Thevenin and Norton equivalents

Module 2: Electronic Devices and Applications. Semiconductors and Diodes: Introduction to semiconductors and their properties. Understanding diodes and their applications (rectifiers, zener diodes). 2.2 Transistors and Amplifiers: Types of transistors (BJT and MOSFET) and their characteristics. Transistor as an amplifier: common emitter, common collector, common base configurations. Digital Electronics: Basics of digital circuits and logic gates. Electronic Sensors and Actuators: Understanding different types of sensors (temperature, light, proximity) Introduction to actuators (motors, relays) and their applications.

Module 3: Electrical Machines and Power Systems

3.1 Introduction to Electrical Machines: Overview of different types of electrical machines (DC motors, AC motors, transformers). Working principles and applications. Transformers and Power Transmission: Understanding transformers and their types Introduction to power transmission and distribution systems. 3.3 Electric Motors: Basics of DC motors and their characteristics. Introduction to AC induction motors and synchronous motors

Module 4: Circuit Design and Troubleshooting. Circuit Design Fundamentals: Overview of circuit design process and best practices. Schematic and PCB design introduction using software tools. Troubleshooting Techniques: Identifying and fixing common circuit issues.

Core Compulsory Readings

"Electric Circuits" Author: James W. Nilsson, Susan A. Riedel Publisher: Pearson Year: 2020

1. "Electronic Devices and Circuit Theory" Author: Robert L. Boylestad, Louis Nashelsky Publisher: Pearson Year: 2019

2. "Electric Machinery Fundamentals" Author: Stephen J. Chapman Publisher: McGraw-Hill Education Year: 2021

Core Suggested Readings

"Electrical Engineering: Principles & Applications" Author: Allan R. Hambley Publisher: Pearson Year: 2019

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

SEMESTER II
SEC
MCCSA02SEC09 Data Analysis and Plotting

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course provides a comprehensive introduction to data analysis and plotting using Origin Software. Participants will learn fundamental data processing techniques, exploratory data analysis (EDA), and advanced data analysis methods. Through hands-on exercises and projects, students will gain proficiency in creating informative and visually appealing graphs, making data-driven decisions, and presenting their findings effectively.

Course Objectives:

- To equip participants with essential data analysis skills using Origin Software.
- To enhance participants' ability to interpret and visualize complex datasets accurately.
- To enable participants to apply statistical techniques for data-driven decision-making.
- To empower participants to create dynamic and interactive graphs for effective data communication.

Learning Outcomes

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

SL #	Course Outcomes
CO1	Identify and apply data analysis techniques: Participants will demonstrate an understanding of various data analysis techniques, including data processing, descriptive statistics, filtering, and basic plotting. They will proficiently apply these techniques using Origin Software to gain insights from raw data.
CO2	Analyze and interpret data using Origin Software: Participants will be able to perform exploratory data analysis (EDA) using Origin Software. They will learn to create and interpret histograms, box plots, scatter plots, and other visualization tools to assess data distributions, relationships, and patterns.
CO3	Apply advanced data analysis methods: Participants will learn and implement advanced data analysis methods, such as statistical hypothesis testing, curve fitting, and peak analysis, using Origin Software. They will understand how to make data-driven decisions based on these analyses.
CO4	Design and create compelling data visualizations: Participants will acquire skills to design publication-quality graphs and visualizations using Origin Software. They will learn to customize graph styles, colors, annotations, and multimedia elements to create engaging and impactful presentations of their data analysis results.

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1:

Introduction to Data Analysis and plotting Software Understanding the importance of data analysis in various fields. Overview of plotting Software and its capabilities. Installing and setting up plotting Software. Importing data into software from different sources. Data processing and cleaning techniques in plotting Software. Introduction to basic plotting and visualization options in plotting Software.

Module 2:

Exploratory Data Analysis (EDA) with Origin. Understanding the concept of EDA and its role in data analysis. Utilizing descriptive statistics to summarize data. Visualizing data distribution using histograms, box plots, and density plots. Exploring relationships between variables using scatter plots and correlation analysis. Implementing data filtering and conditional formatting in plotting software. Customizing plot styles, colors, and annotations for better visualization.

Module 3:

Advanced Data Analysis. Performing statistical analyses in Origin, including t-tests, ANOVA, and regression. Using built-in analysis tools like peak analysis, curve fitting, and smoothing. Handling missing data and dealing with outliers in Origin. Applying data transformations and normalization techniques. Spectral analysis of data.

Module 4:

Advanced Plotting and Data Visualization. Understanding different plot types: 2D and 3D plots, contour plots, and heatmaps. Creating publication-quality graphs and exporting them to various formats. Animating graphs and visualizing time-series data. Final project: Students work on a real-world data analysis project and present their findings using Origin Software.

Core Compulsory Readings

1. Wes McKinney, Python for Data Analysis, O'Reilly, 2012, ISBN: 9781449319793

2. Claus O. Wilke, Fundamentals of Data Visualization: A Primer on Making, O'Reilly, 2019, ISBN: 9781492031086

Core Suggested Readings

1. Stephanie D. H. Evergreen, Effective Data Visualization The Right Chart for the Right Data, SAGE Publications, 2016, ISBN:9781506303079

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

SEMESTER II
SEC
MCCSA02SEC10 Quantum computing

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
2	0	2	2	0	2	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

- Understand the basic principles of quantum mechanics.
- Explore the fundamental concepts of quantum computing.
- Gain knowledge of quantum algorithms and their applications.
- Develop an understanding of quantum gates and quantum circuits.

Learning Outcomes

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

SL #	Course Outcomes
CO1	Develop a foundational understanding of quantum mechanics and its relevance to computing.
CO2	Gain knowledge of quantum algorithms and their potential applications.
CO3	Acquire practical skills in quantum programming and working with quantum circuits
CO4	Understand the challenges and future directions in quantum computing research

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Quantum Mechanics -Overview of classical computing and its limitations -Introduction to quantum mechanics -Wave-particle duality-Superposition and measurement-Quantum entanglement

Module 2: Quantum Computing Basics -Introduction to qubits and quantum gates -Quantum states and quantum operations - Quantum circuits and circuit model of Computation-Measurement and quantum measurement postulate

Module 3: Quantum Algorithms-Quantum parallelism and superposition - The Deutsch-Jozsa algorithm - Grover's algorithm for unstructured search -Shor's algorithm for factoring large numbers

Module 4: Quantum Simulators and Hardware- Overview of quantum simulators and their role in quantum computing research Introduction to quantum hardware (e.g., qubits, quantum gates) - Comparison of different quantum computing technologies (e.g., superconducting qubits, trapped ions) - **Quantum Programming Languages and Frameworks**- Introduction to quantum programming languages (e.g., Q#, Qiskit) - Basics of quantum program structure and execution - Quantum gates and operations in programming languages - Hands-on exercises using quantum programming frameworks

Core Compulsory Readings

Quantum Computing for Everyone" Author: Chris Bernhardt Publisher: The MIT Press Year: 2019

Core Suggested Readings

Programming Quantum Computers: Essential Algorithms and Code Samples" by Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia (O'Reilly Media)

1. Quantum Computing for Computer Scientists" by Noson S. Yanofsky and Mirco A. Mannucci (Cambridge University Press)
2. Quantum Computing: An Applied Approach" by Jack D. Hidary (Springer)

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar	40%
Total	**100%**

End of Semester II

Semester III

SEMESTER III
DISCIPLINE SPECIFIC CORE
MCCSA03DSC14 MACHINE LEARNING TECHNIQUES

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4/1	0	5	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course is intended to familiarize the learners about the fundamental concepts in machine learning and deep learning mechanism for applying and analyzing the huge volume of data generates in real life. The main purpose of this course is to provide foundational understanding of machine learning models and demonstrate how it will be useful for solving complex problems in real world.

Course Objectives:

- Aims to impart basic concepts of Artificial intelligence
- Impart knowledge on different learning scenario in machine learning
- Acquire the knowledge of various machine learning models and its implementation
- Understand various neural networks models architecture and its training algorithms
- Awareness of various dimensionality reduction techniques for optimizing the feature extraction techniques for improving the performance of various classification models
- Acquire the ability to analysis of data using various deep leaning techniques

Course Outcomes:

At the end of the course, the student will be able to:

SL #	Course Outcomes
CO1	Understand basic concepts of machine learning
CO2	Acquire knowledge on supervised and unsupervised learning
CO3	Obtain knowledge on the basics of neural networks
CO4	Understand concepts of deep learning architectures

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔		✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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

Core Suggested Readings

1. Tom M. Mitchell, Machine Learning, McGraw-Hill
2. Kuntal Ganguly, Learning Generative Adversarial Networks, Packt Publishing, 2017.

TEACHING LEARNING STRATEGIES

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

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

Sample Questions to test Outcomes.

1. What is Machine learning?
2. What are the different types of learning?
3. What is supervised learning?
4. What is reinforcement learning?
5. What is unsupervised learning?
6. What do you mean by training in Machine learning?
7. What is training dataset?
8. What do you mean by testing in Machine learning?
9. What is test data?
10. What is validation test?
11. What is meant by overfitting?
12. What do you mean by underfitting?
13. What is regression?
14. What is linear regression?

15. What is feature selection?
16. Explain classification tree.
17. Explain regression tree.
18. Explain about decision tree classifier.
19. What are the different multivariate methods for learning? Explain.
20. Explain the architecture of Artificial Neural Network.
21. Explain the training and testing algorithm for Backpropagation neural network.
22. Explain the different activation functions.
23. What is deep learning? Explain how it differs from machine learning.
24. What is LSTM?
25. What is CNN?

SEMESTER III

DISCIPLINE SPECIFIC CORE

MCCSA03DSC15 COMPUTER GRAPHICS AND IMAGE PROCESSING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4/1	0	5	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course is designed to provide students with a comprehensive understanding of the principles, techniques and algorithms mainly used in Computer Graphics and in Image Processing. It explores the basic concepts and applications of creating, manipulating and rendering of digital images. It covers theoretical as well as practical implementation aspects of digital image processing.

Course Objectives:

- Give To introduce students to the basic concepts and principles of computer graphics and image processing.
- To develop skills in image processing techniques including image pre-processing, image seg9mentation, image enhancement and image filtering techniques.
- To understand the principles of modelling in computer graphics.
- To provide a hands-on experience and implementing image processing algorithms.
- To develop skills in creating and rendering 2D and 3D graphics.
- To understand the principles of geometric transformations and modelling in computer graphics.

Course Outcomes:

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

SL #	Course Outcomes
CO1	To understand the basics on computer graphics and transformation algorithms.
CO2	Learn different clipping algorithms on an image
CO3	To learn the operations on 2D digital images
CO4	Understand filtering and segmentation techniques on digital images.

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔			✔	✔	
CO2	✔	✔			✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Basic Concepts in Computer Graphics. Input devices. Display devices. Line and circle drawing Algorithms. Solid area scan-conversion. Polygon filling. Projections – Parallel, Perspective. Hidden Line Elimination Algorithms. Image processing – digital image representation – edge detection – Robert, Sobel, Canny edge detectors. Scene segmentation and labeling – region- labeling algorithm – perimeter measurement. Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling. Two dimensional transformations-Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Module 2: Projections – Parallel and perspective projections – vanishing points. Visible surface detection methods– Back face removal- Z-Buffer algorithm, A-buffer algorithm, Depth-sorting method, Scan line algorithm. Window to viewport transformation. Cohen Sutherland Line clipping algorithm. Sutherland

Hodgeman Polygon clipping algorithm. Three-dimensional viewing pipeline, Visible surface detection algorithms- Depth buffer algorithm.

Module 3: Introduction to Image processing and applications. Image as 2D data. Image representation in gray scale, Binary and Color images. Fundamental steps in image processing. Components of image processing system. Coordinate conventions. Sampling and quantization. Spatial and Gray Level Resolution. Basic relationship between pixels– neighborhood, adjacency, connectivity. Fundamentals of spatial domain-convolution operation.

Module 4: Basic gray level transformation functions - Log transformations, Power-Law transformations, Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter, Linear and nonlinear filters, and sharpening spatial filters-Gradient and Laplacian. Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging. Edge Detection - Edge Operators- Sobel and Prewitt.

Module X (For Additional Reading and Comprehension by the Students):
Physical based rendering, real time rendering, ambient occlusion, deferred shading, smoothing, remeshing, convex hulls, Super sampling and multi sampling anti-aliasing techniques, GPU programming languages and frameworks :CUDA, OpenCL and OpenGL Shading Language

Core Compulsory Readings

1. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996
2. E. Gose, R. Johnsonbaugh and S. Jost., Pattern Recognition and Image Analysis, PHIPTR, 1996 (Module IV – Image Processing part)
3. James D Foley, Andries van Dam, Steven K Feiner, John F Hughes, Computer Graphics: Principles and Practice
4. Rafael C Gonalez and Richard E Woods, Digital Image Processing

Core Suggested Readings

1. JZhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 1986.
2. William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2e, 1979
3. Berd Jahne, Digital Image Processing: Concepts, Algorithms and Scintific Applications

TEACHING LEARNING STRATEGIES

- Lecturing, Interactive discussions, hands-on exercises, lab sessions, collaborative learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion Lecture, Seminar, Discussion, audio and video presentation, demonstration, practical assignments and exercises

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions to test Outcomes.

1. Explain the concept of anti-aliasing and the techniques used in Computer Graphics. How it will be affected in the quality of an image.
2. Discuss the concept of hidden surface removal in Computer Graphics processing. Explain Depth Buffer algorithm along with its advantages as well as limitations.
3. What are the different steps involved in ray tracing? Explain its advantages and ligations.
4. Explain how computer vision is related to Computer Graphics. Discuss the challenges of computer vision in real world scenarios.
5. What is the role of color models in Computer Graphics? Discuss RGB and CMYK color models and their application in the contest of real-world scenario.

6. Explain different image segmentation techniques. Also discuss about image segmentation evaluation metrics.
7. Explain image enhancement in image processing. Discuss different techniques used for image enhancement.
8. Explain basic steps involved in image processing by explaining each. Discuss the importance of each one.
9. Explain the concept of image registration in image processing. Explain how noise removal and image deblurring affects the overall performance of a digital image.
10. What is image compression in image processing. Differentiate lossless and lossy compression and its importance.

SEMESTER III
DISCIPLINE SPECIFIC CORE
MCCSA03DSC16 QUANTUM COMPUTING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3/1	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

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

Course Outcomes:

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

SL #	Course Outcomes
CO1	Develop a foundational understanding of quantum mechanics and its relevance to computing.
CO2	Gain knowledge of quantum algorithms and their potential applications.
CO3	Acquire practical skills in quantum programming and working with quantum circuits.
CO4	Understand the challenges and future directions in quantum computing research.

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔			✔	✔	✔	✔
CO2		✔		✔	✔	✔	✔
CO3		✔	✔	✔	✔	✔	✔
CO4		✔		✔	✔		✔

COURSE CONTENTS

Module 1: Introduction to Quantum Mechanics -Overview of classical computing and its limitations -Introduction to quantum mechanics -Wave-particle duality-Superposition and measurement-Quantum entanglement

Module 2: Quantum Computing Basics -Introduction to qubits and quantum gates -Quantum states and quantum operations - Quantum circuits and circuit model of Computation-Measurement and quantum measurement postulate

Module 3: Quantum Algorithms-Quantum parallelism and superposition - The Deutsch-Jozsa algorithm - Grover's algorithm for unstructured search -Shor's algorithm for factoring large numbers

Module 4: Quantum Simulators and Hardware- Overview of quantum simulators and their role in quantum computing research Introduction to quantum hardware (e.g., qubits, quantum gates) - Comparison of different quantum computing technologies (e.g., superconducting qubits, trapped ions) - **Quantum Programming Languages and Frameworks**- Introduction to quantum programming languages (e.g., Q#, Qiskit) - Basics of quantum program structure and execution - Quantum gates and operations in programming languages - Hands-on exercises using quantum programming frameworks

Module X (For Additional Reading and Comprehension by the Students):

Advanced Quantum Computing

- Quantum Fourier transform and its applications
- Shor's factorization algorithm and quantum cryptography
- Quantum simulation and quantum machine learning
- Introduction to quantum error correction

Quantum Computing Applications

- Quantum computing in cryptography and secure communication
- Quantum optimization and quantum annealing
- Quantum simulation of physical systems
- Potential impact of quantum computing on various industries

Programming Quantum Computers

- Introduction to quantum software development kits (SDKs)
- Quantum programming languages: Qiskit, Cirq, and PyQuil
- Building and running quantum circuits

Core Compulsory Readings

Quantum Computing for Everyone" Author: Chris Bernhardt Publisher: The MIT Press Year: 2019

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion and lab

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions to test Outcomes.

1. What is a qubit in quantum computing and how is it different from a classical bit?
2. What is superposition in quantum computing and why is it important for quantum algorithms?
3. Explain the concept of entanglement in quantum computing and its significance
   in quantum communication and computation.
4. Describe the fundamental difference between quantum gates and classical logic gates, and provide an example of a commonly used quantum gate.
5. Compare and contrast quantum computing with classical computing. Highlight
   at least three key differences between the two.
6. Explain the concept of quantum algorithms and provide an overview of an important quantum algorithm, such as Shor's algorithm or Grover's algorithm.

End of S3 DISCIPLINE SPECIFIC CORE COURSES

DISCIPLINE SPECIFIC ELECTIVES (POOL B)

SEMESTER III

DISCIPLINE SPECIFIC ELECTIVE

MCCSA03DSE04 ARTIFICIAL INTELLIGENCE

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

To introduce basic principles that drive complex real-world intelligence applications
To introduce and discuss the basic concepts of AI Techniques
To understand how the animal (especially human) intelligence is mimicked by the machines using AI algorithms.

Course Outcomes:

SL #	Course Outcomes
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

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1		✔	✔				✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Introduction - Overview of AI applications. Introduction to representation and search. The Propositional calculus, Predicate Calculus, Using Inference Rules to produce Predicate Calculus expressions, Application.

Module 2: Introduction to structure and Strategies for State Space search, Graph theory, Strategies for state space search, using the State Space to Represent Reasoning with the Predicate calculus (Sate space description of a logical system, AND/OR Graph). Heuristic Search: introduction, Hill-Climbing and Dynamic Programming, The Best-first Search Algorithm, Admissibility, Monotonicity and Informed Ness.

Module 3: Building Control Algorithm for State space search – Introduction, Production Systems, The blackboard architecture for Problem solving. Knowledge Representation – Issues, History of AI representational schemes, Conceptual Graphs, Alternatives to explicit Representation, Agent based and distributed problem solving. Strong Method Problem Solving –Overview of Expert System Technology, Rule Based Expert system, Model - Based, Case-Based and Hybrid Systems (Introduction to Model based reasoning, Introduction to Case Based Reasoning, Hybrid design), Introduction to Planning. Reasoning in Uncertain Situation (introduction), logic based Adductive Inference. Introduction to PROLOG.

Module 4: Machine Learning: Symbol Based – Introduction, Frame –work. The ID3 Decision tree Induction algorithm. Inductive bias and Learnability, Knowledge and Learning, Unsupervised learning, Reinforcement Learning, Machine Learning: Connectionist – Introduction, foundations, Perceptron learning. Machine learning: Social and emergent: Models, The Genetic Algorithm, Artificial Life and Social based Learning.

Core Compulsory Readings

1. S. Russel and p. Norvig, Artificial intelligence – A Modern Approach, 3rdEdn, Pearson
2. D W Patterson, introduction to Artificial Intelligence and Expert Systems, PHI, 1990

Core Suggested Readings

1. George F Luger, Artificial Intelligence – Structures and Strategies for Complex problem solving, 5thEdn, Pearson
2. George J Klir and Bo yuan, Fuzzy sets and fuzzy logic: Theory and Applications, Prentice Hall India 1995
3. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. When you think of AI, what sort of challenges come to your mind?
2. Criticize Turing's criteria for computer software being "intelligent".
3. What are the major issues in *knowledge representation*?
4. Assume that you are developing an expert system for troubleshooting a

complex electronic system. What is the source of uncertainty that can arise in this application?

5. Explain in detail about reinforcement learning with example.
6. In A* Algorithm, what do you mean by Estimated cost, h(n)?

Semester III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE05 BIOINFORMATICS

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3		3	3		3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course provides an insight into the fundamental aspects of Bioinformatics and various algorithms used. The course also provides a detailed account of sequence alignment. An overview about major resources available in the area of Bioinformatics and Biocomputing is also provided.

Course Objectives:

- To introduce the basic concepts of Bioinformatics
- To introduce various software process model
- To build an understanding on various algorithms for sequential alignment

Course Outcomes:

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

SL #	Course Outcomes
CO1	Identify the popular genomic and proteomic databases and analyze genomic data
CO2	Explain the basic algorithms used in Bioinformatics
CO3	Illustrate the concept of sequence alignment
CO4	Describe the features of major biological databases

CO-PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔			✔	✔		✔
CO2	✔		✔	✔	✔	✔	✔
CO3	✔		✔		✔		✔
CO4	✔	✔		✔	✔		✔

COURSE CONTENTS

Module 1: Bioinformatics - Introduction - 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.

Module 2: 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.

Module 3: Sequence alignment - pairwise 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 TCoffee. Phylogenetic algorithms - evaluation of phylogenetic trees, significance.

Module 4: 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.

Core Compulsory Readings

1. Mount D, Bioinformatics: Sequence & Genome Analysis, 2nd Edition, Cold spring Harbor Press, ISBN: 978-087969712.

Core Suggested Readings

1. Dan Gusfield, Algorithms on Strings Trees and Sequences, 1st Edition, Cambridge University Press, ISBN: 0521585198.
2. Pevzner P A, Computational Molecular Biology: An Algorithmic Approach, MIT Press, Cambridge, MA, ISBN: ISBN: 9780262161978.
3. Jeremy J. Ramsden, Bioinformatics: An Introduction, Springer, ISBN: 9789401570961.
4. Sushmita M and Tinku A, Data Mining: Multimedia, Soft Computing and Bioinformatics, Wiley Interscience, ISBN: 9780471460541.

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60%**
Continuous Evaluation (add marks as per our regulation)	**40%**
Tests	40%
Assignment	20%
Seminar…..	40%

Sample Questions to test Outcomes.

1. Write in brief about DNA double helix (3 marks)
2. Illustrate how a string matching algorithm can be used in Computational Biology (5 marks)
3. Explain the features of various genome databases (10 marks)

Semester III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE06 FUZZY SETS AND SYSTEMS

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
02		02	03		03	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: Fuzzy sets & Systems can take imprecise, distorted, noisy input information and produce acceptable but definite output in response to incomplete, ambiguous, distorted, or inaccurate (fuzzy) input. It is easy to construct and understand. Fuzzy logic is a solution to complex problems in all fields of life including medicine, astronomy, agriculture etc. It resembles human reasoning and decision making. Fuzzy systems imitate the way of decision making in humans that involves all intermediate possibilities.

Course Objectives:

- Familiarize the difference in concept of discrete sets and fuzzy sets representation
- Attain the knowledge of representing real world problems using approximation
- Familiarize the concept of fuzzy relations with approximation
- Familiarize the various operations on fuzzy relations
- Acquire the knowledge of membership functions and its properties for representing fuzzy relations
- Attain the knowledge of fuzzification of crisp sets
- Familiarize various defuzzification methods
- Acquire the knowledge of membership functions
- Attain the knowledge of developing membership functions for real world applications
- Familiarize the thoughts of representing knowledge-based systems for solving real life problems using fuzzy logic

- Acquire the knowledge of working of various fuzzy control systems
- Familiarize various clustering algorithms

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

SL #	Course Outcomes
CO1	Fundamental Understanding of Fuzzy Systems
CO2	Proficiency in Classical and Fuzzy Relations
CO3	Mastery of Membership Functions and Fuzzy Logic
CO4	Advanced Fuzzy System Development and Applications

CO-PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔				✔
CO2	✔		✔			✔	✔
CO3	✔	✔	✔	✔		✔	✔
CO4	✔	✔		✔	✔	✔	✔

COURSE CONTENTS

Module 1: 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).

Module 2: 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.

Module 3: 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.

Module 4: 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, Cmeans clustering (Hard and Fuzzy), Fuzzy c-means algorithm.

Core Compulsory Readings

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

Core Suggested Readings

1. Hajek P, Metamathematics of Fuzzy Logic. Kluwer, 1998.

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Digital Learning.

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Printed note

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40

Sample Questions to test Outcomes.

1. What is fuzzy set?
2. Differentiate between crisp and fuzzy sets.
3. Explain the differences between chances v/s fuzziness.
4. Explain the properties of fuzzy sets.
5. Discuss the various operations on fuzzy sets.
6. What is classical relation?
7. Explain fuzzy relation and its operations.
8. What is cartesian product?
9. Define the cardinality of fuzzy relation with examples.
10. What is λ-cut?
11. What is membership function? Explain its properties.
12. What do you mean by fuzzification?
13. Explain the different fuzzification methods.
15. What do you mean by defuzzification?
16. Explain the different defuzzification methods.
17. What id tolerance relation? Explain how will you convert a tolerance relation to equivalence relation.
18. What is meant linguistic hedge?
19. What is cluster analysis?
20. Explain about fuzzy c-mean clustering techniques.

Semester III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE07 GRAPH THEORY AND COMBINATORICS

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3		3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course provides an exposure to the basics of Graph Theory and its applications. The course also discusses principles of combinatorics, recurrence and nonhomogeneous recurrence relations. The course provides an insight into applying Graph Theory in the development of solutions in Computer Science.

Course Objectives:

- To introduce the fundamental concepts of Graph Theory
- To introduce the various applications of Graph Theory
- To comprehend the principles of combinatorics, recurrence and nonhomogeneous recurrence relations

Course Outcomes:

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

SL #	Course Outcomes
CO1	Identify the fundamental concepts of Graph Theory
CO2	Illustrate the applications of Graph Theory
CO3	Apply the principles of combinatorics
CO4	Explain the recurrence and nonhomogeneous recurrence relations

CO-PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔	✔			✔
CO2	✔		✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔		✔
CO4	✔	✔	✔	✔		✔	✔

COURSE CONTENTS

Module 1: 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 coloring and chromatic polynomials, map coloring.

Module 2: 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.

Module 3: Fundamental principles of counting, permutations and combinations, binomial theorem, combinations with repetition, combinatorial numbers, Principle of inclusion, derangements, arrangements with forbidden positions.

Module 4: 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.

Core Compulsory Readings

1. Grimaldi R. P. Discrete and Combinatorial Mathematics: an Applied Introduction, 3rd Edition, Addison Wesley, 1994.

Core Suggested Readings

1. Corman T. H., Leiserson C. E., Rivest R. L., Introduction to algorithms, Prentice Hall India, 1990
2. Mott J.L., Kandel A. and Baker T.P., Discrete Mathematics for Computer Scientists and Mathematicians‖, 2nd Edition, PHI
3. Rosen K.H., Discrete Mathematics and its Applications‖, 3rd Edition, McGraw Hill
4. Clark J. and Holton D. A., A first look at Graph theory‖, World Scientific.

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40

Sample Questions to test Outcomes.

1. State Chinese postman problem
2. Illustrate Disjkstra's shortest path algorithm
3. State the principle of inclusion and illustrate with a suitable example.

Semester III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE08 SOFTWARE ARCHITECTURE

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3		3	3		3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course focuses on providing the basics of Software Architecture.

Course Objectives:

- To understand the basics of Software Architecture
- To familiarize with various aspects of Service Oriented Architecture
- To explain about Interactive Systems
- To integrate the Software Engineering skills

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand the basics of Software Architecture
CO2	Describe various aspects of Service Oriented Architecture
CO3	Mastery of Architectural Patterns and Design Principles
CO4	Understanding Advanced Architectural Concepts

CO-PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔		✔		✔	✔
CO2	✔		✔		✔		✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔		✔	✔	

COURSE CONTENTS

Module 1: 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.

Module 2: 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 Behaviour - Building the Documentation Package.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

1. Hofmeister, Nord, Soni, Applied Software Architecture, Addison-Wesley

Core Suggested Readings

1. BPaul Clements et al., Documenting-software-architectures-views-and-beyond, 2nd edn, Pearson
2. Arlow & Neustadt, Enterprise Patterns And MDA-Building Better Software With Archetype Pattern An UML, Pearson, 2004
3. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Pattern-Oriented Software Architecture, Vol 1 - A System Of Patterns, Wiley.
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns, Pearson

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40

Sample Questions to test Outcomes.

1. Define software architecture and explain its role in the overall software life cycle.
2. Discuss key architectural principles and their significance in designing software systems.
3. Discuss the Module Architecture View and various styles associated with it.
4. Define the Component-and-Connector Viewtype and its styles in software architecture.
5. Explain the concept of Model Driven Architecture (MDA) with archetype patterns.
6. Discuss literate modeling and its role in software architecture.
7. Explore archetype patterns such as Customer Relationship Management (CRM), Product, Quantity, and Rule.
8. Discuss service variation patterns and service extension patterns in the context of SOA.
9. Explore object management patterns and adaptation patterns.
10. Explain communication patterns and their role in software architecture.

SEMESTER III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE09 FOUNDATIONS OF NATURAL LANGUAGE PROCESSING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

The course introduces the fundamentals of Natural Language Processing (NLP) from an algorithmic viewpoint. The course provides insight into how machines can deal with NLP. A gist about the various applications of NLP is also discussed.

Course Objectives:

- To introduce the fundamentals of NLP from an algorithmic viewpoint
- To introduce the use of CFG and PCFG in NLP
- To illustrate the process of syntax analysis in NLP
- To explain the fundamentals of speech processing in NLP

Course outcomes:

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

SL #	Course Outcomes
CO1	Acquire basic knowledge on natural language and automata.
CO2	Illustrate the process of syntax analysis in NLP
CO3	Apply grammars and parsing in NLP.
CO4	Discuss some applications of Natural Language Processing (NLP)

CO-PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔		✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4			✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Language: Linguistic Knowledge, Grammar, Language and Thought, computational linguistics vs NLP, why NLP is hard ?, why is NLP 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.

Module 2: N-grams: simple N-grams, smoothing, Applications, language modeling. 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

Daniel Jurafsky and James H Martin. Speech and Language Processing.

Core Suggested Readings

1. Hobson Lane, Cole Howard, Hannes Hapke. Natural Language Processing in Action
2. 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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions to test Outcomes.

1. List the reasons for choosing CFG to represent language in parsing (3 marks)

2. Distinguish between (i) semantics, pragmatics and discourse (5 marks)

3. (i) Write different types of inferences (5 marks)

(ii) Write FOPC for the following sentences: (5 marks)

All cats and dogs hate each other

I arrived in New York.

SEMESTER III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE10 COMPUTER ORGANIZATION AND ARCHITECTURE

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: This course is intended to familiarize with concepts in computer organization and computer architecture.

Course Objectives:

- To impart an understanding of the internal organization and operations of a computer
- To introduce the concepts of processor logic design and control logic design

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

SL #	Course Outcomes
CO1	Mastery of Computer Architecture Fundamentals.
CO2	Understand the working of DMA and cache memory
CO3	Describe the design of arithmetic unit and logic unit
CO4	Understand the basics of control logic design and micro-programmed control

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔	✔		✔	✔
CO2	✔		✔		✔		✔
CO3	✔		✔		✔	✔	✔
CO4	✔	✔	✔	✔		✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: Memory system: DMA, Memory hierarchy, Cache memory, performance requirements, virtual memory, memory management requirements, 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).

Module 3: Processor Logic Design: Register transfer logic – inter register transfer – arithmetic, logic and shift micro operations – conditional control statements.Processor organization: design of arithmetic unit, logic unit, arithmetic logic unit and shifter – status register – processor unit – design of accumulator.

Module 4: Control Logic Design: Control organization – design of hardwired control –control of processor unit –PLA control. Micro-programmed control: Microinstructions –horizontal and vertical micro instructions – micro-program sequencer –micro programmed CPU organization.

Module X (For Additional Reading and Comprehension by the Students):
Memory system: Basic concepts, Semiconductor RAMS, ROMs, secondary storage devices. Instructions and instruction sequencing – addressing modes – ARM Example (programs not required). Basic I/O operations – stacks subroutine calls. Algorithms for multiplication and division of binary and BCD numbers — array multiplier.

Core Compulsory Readings

1. Hamacher C., Z. Vranesic and S. Zaky, Computer Organization ,5/e, McGraw Hill, 2011

Core Suggested Readings

1. Patterson D.A. and J. L. Hennessey, Computer Organization and Design, 5/e, Morgan Kauffmann Publishers, 2013
2. William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson, 9/e, 2013
3. Chaudhuri P., Computer Organization and Design, 2/e, Prentice Hall, 2008
4. Rajaraman V. and T. Radhakrishnan, Computer Organization and Architecture, Prentice Hall, 2011

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Digital Learning

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40

Sample Questions to test Outcomes.

1. Explain instruction fetch and execution cycle.
2. Multiply (-10) and 4 using Booth algorithm.
3. Define cache memory.
4. How are multiple interrupts handled?
5. Explain the design of arithmetic unit.
6. Differentiate horizontal and vertical micro instructions.

SEMESTER III
DISCIPLINE SPECIFIC ELECTIVE
MCCSA03DSE11 FOUNDATIONS IN DATA SCIENCE

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
3	0	3	3	0	3	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Foundations of Data Science is an introductory course designed to provide students with a strong grounding in the fundamental concepts, techniques, and tools used in the field of data science. The course covers various aspects of data science, including data collection, data cleaning, data analysis, and data visualization. Through a combination of lectures, hands-on projects, and practical exercises, students will develop the essential skills required to work with data effectively and derive meaningful insights from it.

Course Objectives:

- To introduce students to the field of data science and its applications in diverse industries.
- To familiarize students with the data science workflow and the steps involved in processing and analyzing data.
- To equip students with essential programming skills using languages such as Python and R.
- To provide a comprehensive understanding of statistical concepts and their applications in data analysis.
- To introduce students to machine learning algorithms and their use in predictive modeling and clustering tasks.
- To emphasize the importance of ethical considerations in data science and responsible data handling practices.

Course Outcomes:

By the end of this course, students should be able to:

SL #	Course Outcomes
CO1	Proficiently apply programming languages (Python and R) for data processing, analysis, and machine learning.
CO2	Utilize statistical techniques to explore data, make data-driven decisions, and create meaningful visualizations.
CO3	Demonstrate an understanding of the data science workflow, including ethical considerations in data collection and privacy.
CO4	Collaborate effectively on data science projects and communicate results to diverse audiences.

CO- PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔	✔	✔	✔	✔
CO2	✔	✔		✔	✔	✔	✔
CO3				✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

Course Contents

Module 1: Introduction to Data Science .Overview of Data Science and its applications in various industries. Understanding the Data Science workflow: data collection, data cleaning, data exploration, data analysis, and data visualization. Introduction to data types, data formats, and data storage. Introduction to programming languages and libraries commonly used in data science (e.g., Python, R, NumPy, Pandas). Basics of data processing and cleaning techniques.

Module 2: Statistics and Data Analysis . Introduction to descriptive and inferential statistics. Measures of central tendency and variability. Hypothesis testing and confidence intervals. Correlation and regression analysis. Understanding probability distributions (e.g., normal, binomial, Poisson) and their applications in

data science. Data visualization techniques: scatter plots, histograms, box plots, etc.

Module 3: Data Processing and Machine Learning . Data preprocessing techniques: data scaling, encoding categorical variables, handling missing data, and feature engineering. Introduction to supervised, unsupervised, and semi-supervised machine learning algorithms. Linear and logistic regression for predictive modeling. Decision trees, random forests, and ensemble methods. Clustering algorithms (e.g., k-means, hierarchical clustering) for unsupervised learning. Model evaluation and performance metrics.

Module 4: Big Data and Data Ethics. Introduction to Big Data concepts and challenges in processing large datasets. Introduction to distributed computing and tools like Apache Hadoop and Spark. Ethical considerations in data science: privacy, bias, and fairness in data collection and analysis. Responsible data handling practices and data governance. Case studies and real-world applications in various domains.

Book of Study:

1. "Foundations of Data Science: An Introduction to Data Analysis and Machine Learning" Author: John Doe Publisher: XYZ Press Year: 2023
2. Reference Book: Title: "Python for Data Analysis" Author: Wes McKinney Publisher: O'Reilly Media Year: 2021
3. Reference Book: Title: "Introduction to Statistical Learning" Authors: Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani Publisher: Springer Year: 2017
4. Reference Book: Title: "Data Science for Business" Authors: Foster Provost and Tom Fawcett Publisher: O'Reilly Media Year: 2013

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

End of S3 Discipline Specific Elective Courses

IDC/MDC (Offered for other Departments)

SEMESTER III
IDC/MDC
MSCSC03MDC01 DESIGN AND ANALYSIS OF ALGORITHMS

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

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

Course Objectives:

- To introduce basic principles that drive various algorithm design strategies
- Discuss the complexity analysis techniques and overview of P, NP problems
- Discuss about the concept of design and analysis of parallel algorithms

Course Outcomes:

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

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

CO-PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔		✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔		✔		✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: 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).

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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.

Core Suggested Readings

1. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013
2. Upadhyay, N, Design and Analysis of Algorithms, Sk Kataria & Sons, 2008.
3. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley,
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India
5. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill
6. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd
7. Oded Goldreich, P,NP and NP- Completeness, Cambridge University Press,

2011.
8. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume - 1, Addison Wesley, 1997.
9. Sanjeev Arora and Boaz Borak, Computational Complexity- A Modern Approach, Cambridge University Press; 2009.
10. Daniel Hills W and Bruce M Boghosian, Parallel Scientific Computation, Science, Vol 261, Pp. 856-863

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions to test Outcomes.

1. Differentiate Direct Recursion and Indirect Recursion.

2. Solve the recurrence **$T(n) = 2T(\sqrt{n}) + \log n$ 2.**

3. Compute the time complexity of the following:

```
for i  100 to m-1
{ for j  10 to i
  {
    A  B + C[i][j]
```

```
    }
  }
```

4. Let $\mathbf{T(n) = 4\ T\ (n/2) + n^3}$, then show that $\mathbf{f(n) = \Omega(n^3)}$ and $\mathbf{T(n) = \theta(n^3).}$

5. Prove that Hamiltonian Cycle is NP Complete.

6. How can we solve Knapsack problem using Branch-and-Bound technique?

7. Given a set S = **{2, 4, 6}** and Weight = **6**. Find subset sum using backtracking approach.

8. Let **H(t)** be the number of multiplications in the following:

```
int Factorial( intt)
 {
   if (t = = 0)
      then return 1
   else
      return t* Factorial(t-1)
  }
```

Prove that **H(t) = t.**

9. 'The running time is directly proportional to the frequency count of the algorithm.' Explain the meaning of the statement in detail.

10. The recurrence $\mathbf{T(n) = 7T(n/2) + n^2}$describes the running time of an algorithm **A**. A competing algorithm $\mathbf{A^I}$ has a running time of $\mathbf{T^I(n)= kT^I(n/4)+n^2.}$ What is the largest integer value for **k** such that $\mathbf{A^I}$ is asymptotically faster than **A**?

11. What do mean by Parallel Prefix Computation?

12. Can the master theorem be applied to the recurrence $\mathbf{T(n) = 2T(n/2) + n \log n}$ **?** Why or Why not?

SEMESTER III
IDC/MDC
MSCSC03MDC02 PRINCIPLE OF PROGRAMMING AND NUMERICAL METHODS

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:
The course provides an insight into building programs using Python programming language, especially for numerical methods. Various numerical methods and concepts of differentiation and integration are also discussed.

Course Objectives:
- To understand various numerical methods
- To acquire knowledge about Errors and Approximations
- To develop programs using numerical methods
- To understand the basic concepts of Python programming

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

SL #	Course Outcomes
CO1	Acquainted with Numerical Methods.
CO2	Obtain the knowledge about Numerical Integration, Differentiation and its applications.
CO3	Understanding the basic concepts of Python programming.
CO4	Ripen skill in programming.

CO-PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔			✔		✔
CO2	✔	✔			✔		✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Programming: Basic concepts, algorithm, flow chart, programming languages – classification. Syntax and semantics of programming languages - Imperative and OO Languages. Functional Languages - Logic Programming Languages. Features of Python, Different Methods to Run Python, Basic Elements (Objects, Expressions, Numerical Types, Strings, Variables), Comments, Indentation in Python, Input and Output in Python, import function, Operators in Python, Branching (if, else, elif), Iteration (while, for), range and enumerate functions, Tuples, Lists, Sets, Dictionaries, Built - in methods of lists, sets and dictionaries, Mutable and Immutable Objects.

Module 2: Functions Definition, Function Calling, Function Arguments (Required, Keyword, Default), Recursion, Modules, Built-in Modules, Creating Modules, File Handling (Opening, Closing, Writing, Reading), Exceptions, Built-in Exceptions (Index Error, Overflow Error, Zero Division Error, Runtime Error), Exception Handling. Class Definition, Object Creation, Built-in Attribute Methods, Object Oriented Programming Features of Python. Arrays in Python, Numpy Module, ndarray, Creating Arrays (array, zeros, ones, empty, linspace, arrange, random), Two-Dimensional Array, Indexing, Slicing, Iterating, Copying, Splitting, Shape Manipulation (reshape, transpose, resize), Arithmetic Operations on Arrays. Data Visualization in Python matplotlib Module, pyplot, plot(), scatter, bar charts, Formatting, figure(), subplot(), text(), xlabel(), ylabel(), title(), Plotting Simple Mathematical Functions (sin x, x^2).

Module 3: Introduction to Numerical Methods: Nature of numerical problems; computer based solutions. Errors and Approximations. Nonlinear equations – Bisection Method, Regular - Falsie Method, Newton Raphson. System of Linear Equations - Gauss elimination, Gauss Jordan elimination, Triangulation method, Iterative method, Jacobi. Case study by writing algorithms.

Module 4: Numerical Integration and Differentiation: Concept of differentiation and Integration. Taylors series and Eulers methods-Simpson's Romberg, Gaussian, Runge - Kutta methods. Case study by writing algorithms.

Core Compulsory Readings

1. Jean-Paul Tremblay, R Manohar, Discrete Mathematical Structures with Application to Computer Science, McGraw Hill
2. Sastry S.S., Introductory Methods of Numerical Analysis, January 2012 PHI
3. Balagurusamy, E., Numerical Methods, Tata McGraw-Hill, New Delhi, 1999
4. R. G. Dromey, How to solve it by computer, Pearson education, fifth edition, 2007
5. Dr. Jeeva Jose, Taming Python By Programming, Khanna Publishing
6. John V. Guttag, Introduction to Computation and Programming Using Python with Application to Understanding Data, PHI (2016)
7. https://www.numpy.org/devdocs/user/quickstart.html
8. https://matplotlib.org/2.0.2/users/pyplot_tutorial.html

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions:

1. Explain the importance of algorithms in programming and how flowcharts aid in algorithm visualization.

2. Differentiate between imperative and object-oriented programming languages, providing examples of each.
3. Discuss the features of Python and various methods to run Python programs.
4. Explain the concepts of recursion and modular programming, providing examples.
5. Discuss file handling in Python, including opening, closing, writing, and reading files.
6. What are exceptions in Python, and how are they handled? Provide examples of built-in exceptions.
7. Explain the nature of numerical problems and discuss the importance of computer-based solutions.
8. Define errors and approximations in the context of numerical methods.
9. Describe the Bisection Method for solving nonlinear equations, providing an algorithmic overview.
10. Explain the application of Taylor series in numerical differentiation.
11. Discuss Euler's methods for numerical integration, highlighting their advantages and limitations.
12. Provide an overview of Simpson's Rule and Romberg integration methods.
13. Explain the principles behind Gaussian integration and its application in numerical methods.

SEMESTER III
IDC/MDC
MSCSC03MDC03 JAVA PROGRAMMING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course introduces JAVA programming language with object-oriented programming principles. Emphasis is placed on developing skills on implementing classes, inheritance, polymorphism and exception handling. This course covers GUI programming using swing.

Course Objectives:

- Understand object oriented programming
- Use of conditional statements and looping statements to solve problems
associated with decision making and repetitions
- Write programs using more advanced JAVA features such as composition
of objects, operator overloading, dynamic memory allocation, inheritance and polymorphism, file I/O, and exception handling
- Improve the problem solving skills
- Get knowledge about the basic concept of writing a program

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand the basic concepts of java programming like control statements ,arrays and functions
CO2	Obtain knowledge about object oriented programming concepts
CO3	Acquire knowledge about file handling.
CO4	Apply Exception handling techniques and design GUI based applications using swing

CO-PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔			✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Review of Programming Concepts: program, identifiers, variables, constants, primitive data types, expressions, control statements, structured data types, arrays, functions.

Module 2: Object Oriented Concepts: Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, Abstract classes, Interfaces and Packages.

Module 3: File Handling: Byte Stream, Character Stream, File I/O Basics, File Operations, Serialization.

Module 4: Exception handling: Throw and Exception, Throw, try and catch Blocks, Multiple Catch Blocks, Finally Clause, Throwable Class, Types of Exceptions, java.lang Exceptions, Built-In Exceptions.GUI Design: GUI based I/O,

Input and Message Dialog boxes, Swing components, Displaying text and images in windows.

Core Compulsory Readings

1. James Gosling, Bill Joy, Guy L. Steele Jr, GiladBracha, Alex Buckley, The Java Language Specification, Java SE 7 Edition, Addison-Wesley, 2013
2. Cay S. Horstmann, Core Java - Vol. I – Fundamentals, 10th Edition, Pearson, 2017
3. Deitel & Deitel, Java-How to Program (9th ed.), Pearson Education, 2012
4. Richard Johnson, An Introduction to Java Programming and Object-Oriented Application Development, Thomson Learning, 2006
5. Herbert Schildt, Java: The Complete Reference, 10th Edition, McGraw-Hill Education, 2018

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions to test Outcomes.

1. Create a class named 'Student' with string variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as 2 and name as John by creating an object of Student class.
2. Create a class to perform method overloading by changing the

number of parameters

3. Create a class with a method to check if a number is less than 0 or not. If the number is less than 0, throw Arithmetic Exception, otherwise, print division is possible

4. Create a calculator using swing.

5. Create a simple program to read a text file MyFile.txt line by line using the Buffered Reader class.

SEMESTER III
IDC/MDC
MSCSC03MDC04 MACHINE LEARNING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

The course is designed to offer an introduction to machine learning approach to Artificial Intelligence. The course provides an overview of the concepts of supervised and unsupervised learning methods. A discussion about deep learning is also included.

Course Objectives:

- To understand machine learning approach to Artificial Intelligence
- To understand fundamental issues and challenges of supervised and unsupervised learning techniques
- To design and implement supervised and unsupervised machine learning algorithms for real-world applications
- To appreciate the underlying mathematical relationships within and across Machine Learning algorithms

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand basic concepts of machine learning
CO2	Acquire knowledge on supervised learning and artificial neural networks.
CO3	Obtain knowledge on unsupervised learning.
CO4	Understand the concepts of deep learning architectures

CO-PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔	✔	✔	✔	✔
CO2	✔	✔		✔	✔	✔	✔
CO3	✔	✔		✔	✔	✔	✔
CO4		✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Introduction: Learning theory. Basics: Introduction to Machine Learning - Different Forms of Learning, Basics of Probability Theory, Linear Algebra and Optimization. Regression Analysis: Linear Regression, Ridge Regression, Lasso, Bayesian Regression, Regression with Basis Functions.

Module 2: Supervised learning: Linear separability and decision regions, Linear discriminants, Bayes optimal classifier, Linear regression, Standard and stochastic gradient descent, Lasso and Ridge Regression, Logistic regression, Support Vector Machines, Artificial Neural Networks ,Perceptron, Back propagation, , Decision Tree Induction, Overfitting, Pruning of decision trees, Bagging and Boosting, Dimensionality reduction and Feature selection. Support Vector Machines: Structural and empirical risk, Learning nonlinear hypothesis using kernel functions.

Module 3: Unsupervised learning: Clustering, Mixture models, Expectation Maximization, Spectral Clustering, Non-parametric density estimation. Dimensionality Reduction: Principal Component Analysis, Independent

Component Analysis, Multidimensional Scaling, and Manifold Learning. Reinforcement Learning: Q-Learning, Temporal Difference Learning.

Module 4: Introduction to Deep Networks: Fundamentals of deep learning, Deep Feed forward 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.

Core Compulsory Readings

1. E. Alpaydin, Introduction to Machine Learning, 3rd Edition, Prentice Hall of India, 2014
2. T Hastie, R Tibshirani and J Friedman, The Elements of Statistical Learning Data Mining, Inference, and Prediction, 2nd Edition, Springer, 2009.
3. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2010
4. R. O. Duda, P. E. Hart, and D.G. Stork, Pattern Classification, John Wiley and Sons, 2012
5. Simon O. Haykin, Neural Networks and Learning Machines, Pearson Education, 2016
6. Tom Mitchell, Machine Learning, McGraw Hill Education, First Edition

Core Suggested Readings

1. R.O. Duda, P.E. Hart and D.G. Stork, Pattern Classification, Wiley, Second Edition
2. Jiawei Han and Michelline Kamber, Data Mining: Tools and Techniques, Morgan Kaufmann, 3rd Edition
3. Hastie, Tibshirani and Friedman, Elements of Statistical Learning, Springer.

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Question:

1. How does linear algebra play a role in machine learning, and what are its fundamental concepts?
2. Discuss the significance of optimization in the context of machine learning.
3. Explain the principles behind linear discriminants and the Bayes optimal classifier.
4. Compare and contrast standard and stochastic gradient descent.
5. Discuss the concepts of overfitting, pruning of decision trees, and the role of bagging and boosting.
6. Discuss the concepts of spectral clustering and non-parametric density estimation.
7. Explain the principles behind dimensionality reduction techniques such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA).
8. Discuss the various regularization techniques used in deep learning.
9. Explain the optimization methods employed in training deep models.
10. Provide an introduction to convolutional networks and their applications.

SEMESTER III
IDC/MDC
MSCSC03MDC05 FOUNDATIONS IN DATA SCIENCE

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:
This course is designed to offer the fundamental aspects of data science.

Course Objectives:

- To demonstrate proficiency with statistical analysis of data
- To develop the ability to build and assess data-based models
- To execute statistical analyses and interpret outcomes
- To apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively

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

SL #	Course Outcomes
CO1	Understand the fundamentals of data science
CO2	Illustrate various methods for statistical data modelling
CO3	Illustrate various methods for predictive modelling
CO4	Illustrate descriptive modeling, association rule mining data mining

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔		✔	✔	✔	✔
CO2	✔	✔		✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Introduction: Introduction data acquisition, data preprocessing techniques including data cleaning, selection, integration, transformation and reduction, data mining, interpretation.

Module 2: Statistical data modeling: Review of basic probability theory and distributions, correlation coefficient, linear regression, statistical inference, exploratory data analysis and visualization.

Module 3: Predictive modeling: Introduction to predictive modeling, decision tree, nearest neighbor classifier and naïve Baye's classifier, classification performance evaluation and model selection.

Module 4: Descriptive Modeling: Introduction to clustering, partitional, hierarchical, and density-based clustering (k-means, agglomerative, and DBSCAN), outlier detection, clustering performance evaluation. Association Rule Mining: Introduction to frequent pattern mining and association rule mining, Apriori algorithm, measures for evaluating the association patterns. Text Mining: Introduction of the vector space model for document representation, term frequency-inverse document frequency (tf-idf) approach for term weighting, proximity measures for document comparison, document clustering and text classification.

Core Compulsory Readings

1. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython, 2nd Ed., O'Reilly, 2017.

Core Suggested Readings

1. P. Tan, M. Steinbach, AKarpatne, and V. Kumar, Introduction to Data Mining, 2nd Ed., Pearson Education, 2018.
2. G James, D Witten, T Hastie and R Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer Texts in Statistics, Springer, 2013.
3. G. Grolemund, H. Wickham, R for Data Science, 1st Ed., O'Reilly, 2017.

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Question:

1. What is data acquisition, and why is it a crucial step in the data mining process?
2. Explain the different techniques involved in data preprocessing, including data cleaning, selection, integration, transformation, and reduction.
3. Explain the concept of correlation coefficient and its importance in analyzing relationships between variables.
4. Discuss linear regression and its role in predictive modeling.
5. Compare and contrast nearest neighbor classifiers and naïve Bayes classifiers.
6. Explain the importance of classification performance evaluation and model selection in predictive modeling.
7. Explain the principles of outlier detection and its significance in data analysis.
8. Introduce association rule mining and discuss the Apriori algorithm.
9. What is the vector space model for document representation, and how does it contribute to text mining?

SEMESTER III
IDC/MDC
MSCSC03MDC06 DIGITAL SIGNAL PROCESSING

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course offers the fundamentals of digital signal processing. Topics such as modern digital signal processing algorithms and applications, analysis of discrete time signals are included.

Course Objectives:

- To study the modern digital signal processing algorithms and applications
- To study the analysis of discrete time signals
- To achieve comprehensive knowledge to use of digital systems in real time applications
- To apply the algorithms for a wide area of recent applications.

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand the fundamentals of digital signal processing.
CO2	Understand various classes of digital signals.
CO3	Illustrate the time domain representations of signals and systems.
CO4	Explain the frequency analysis of signals.

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1		✔		✔	✔	✔	✔
CO2	✔	✔		✔	✔	✔	✔
CO3		✔		✔	✔	✔	✔
CO4		✔	✔	✔	✔	✔	✔

COURSE CONTENTS

Module 1: Signals and Signal Processing - Characterization and classification of Signals, Typical signal processing operations, Typical Signal Processing Applications, Advantage of Digital Signal Processing.

Module 2: Classification of signals –Introduction to vector space - The concept of frequency in continuous and discrete time signals -Sampling of analog signals – Sampling theorem – Quantization and Coding – Digital to analog conversion.

Module 3: Time Domain Representation of signals and systems - Discrete time signals, Operations on sequences, Discrete time Systems, Linear Time invariant Discrete Time Systems - convolution sum – correlation of discrete time signals, Z-Transform.

Module 4: Frequency Analysis of Signals - Frequency Analysis of Continuous Time Signals, Frequency Analysis of Discrete Time Signals, Fourier Transform of discrete time signals –Discrete Fourier Transform (DFT). FFT (Qualitative idea only) - Wavelet Transform - FIR and IIR Filters.

Core Compulsory Readings

1. Proakis, John G. and Dimitris G. Manolakis. Digital signal processing: principles, algorithms and applications. Pearson Education India, 2001

Core Suggested Readings

1. Roberts, Michael J. Signals and systems: analysis using transform methods and MATLAB. McGraw-Hill Higher Education, 2011
2. Oppenheim, Alan V., and Ronald W. Schafer. Digital Signal Processing [by] Alan V. Oppenheim [and] Ronald W. Schafer.Prentice-Hall, 1975
3. Antoniou, Andreas. Digital signal processing. McGraw-Hill, 2016
4. Rainer, Lawrence R., Bernard Gold, and C. K. Yuen. Theory and application of digital signal processing. Prentice-Hall, 2007

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar…..	40
Total	**100**

Sample Questions:

1. Define and differentiate between continuous and discrete signals. Provide examples of each.
2. What are the typical operations performed in signal processing, and how do they contribute to signal characterization?
3. Explain the sampling theorem and its importance in signal processing.
4. What is the process of quantization and coding in the context of digital signals?
5. Describe the steps involved in digital-to-analog conversion.
6. What are Linear Time Invariant Discrete-Time Systems, and how does convolution sum play a role in their analysis?
7. What is the Fourier Transform, and how does it relate to the frequency analysis of signals?
8. Briefly explain the Discrete Fourier Transform (DFT) and its applications. Provide a qualitative idea of FFT.

SEMESTER III
IDC/MDC
MSCSC03MDC07 QUANTUM COMPUTING AND INFORMATION THEORY

Credit			Teaching Hours			Assessment Weightage (%)		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course offers the basics of quantum computing information theory.

Course Objectives:

- To study the basics of complex vector spaces
- To study the fundamental concepts of quantum mechanics as applied in quantum computing.
- To learn the architecture and algorithms in quantum computing
- To study the fundamentals of quantum computations

Course Outcomes:

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

SL #	Course Outcomes
CO1	Understand the fundamentals of quantum computing
CO2	Understand the fundamentals concepts of mathematics and physics required to learn quantum computing
CO3	Illustrate the principles of quantum circuits
CO4	Explain the various algorithms and applications in quantum computing

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔			✔	✔	✔	✔
CO2		✔		✔	✔	✔	✔
CO3		✔	✔	✔	✔	✔	✔
CO4		✔		✔	✔		✔

COURSE CONTENTS

Module 1: Introduction to Quantum Computation: Quantum bits, Bloch sphere representation of a cubit, multiple cubits.

Module 2: Background Mathematics and Physics: Hilbert space, Probabilities and measurements, entanglement, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis.

Module 3: Quantum Circuits: single qubit gates, multiple qubit gates, design of quantum circuits.

Module 4: Quantum Information and Cryptography: Comparison between classical and quantum information theory. Bell states. Quantum teleportation. Quantum Cryptography, no cloning theorem. Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor factorization, Grover search.

Core Compulsory Readings

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press.-2002.
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol II: Basic Tools and Special Topics, World Scientific.-2004.
3. Pittenger A. O., An Introduction to Quantum Computing Algorithms-2000.

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Notes

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation (add marks as per our regulation)	**40**
Tests	40
Assignment	20
Seminar	40
Total	**100**

Sample Questions:

1. Explain the concept of quantum bits (qubits) and how they differ from classical bits.
2. What is the significance of entanglement when working with multiple qubits in quantum computation?
3. Discuss the basics of quantum mechanics and its fundamental principles.
4. How do measurements in bases other than the computational basis impact quantum systems?
5. Explore the design principles behind quantum circuits for multiple qubits.
6. Explain the concept of Bell states and their significance in quantum entanglement.
7. Walk through the process of quantum teleportation and its applications.
8. Discuss the no-cloning theorem and its implications in quantum cryptography.
9. Provide an overview of quantum algorithms, mentioning Deutsch's algorithm, Deutsch-Jozsa algorithm, Shor's factorization, and Grover's search.

End of S3 Multi Disciplinary Elective Course

End of Semester III

Semester IV

Semester IV | Elective III I MCCSA04DSE12 - 19 | POOL C

SEMESTER IV
DSE COURSE
MCCSA04DSE12 Big Data Analytics

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CA	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

This course provides students with an overview of the fundamental concepts, technologies, and tools associated with big data. This course deals with the importance of Big Data, Stream Data Model in Big Data, Big Data Analytics and Hadoop Distributed File System.

Course Objectives:

- Understand the Big Data Platform and its Use cases.
- Understand the key issues in big data analytics and its associated applications in business analytics.
- Understand several key big data technologies used for storage, analysis and manipulation of data.
- Illustrate different types of big data technologies in the Hadoop parallel world.
- Demonstrate the concepts in Hadoop for application development.
- Provide HDFS Concepts and Interfacing with HDFS
- Understand Map Reduce Jobs
- Explore map reduce framework and optimize its jobs.
- Explain the basic methodologies of pig and hive.

Course Outcomes:

SL #	Course Outcomes
CO1	Understand Big Data, importance, challenges and various sources of data
CO2	Explain about the fundamental concepts of streams and real time analysis
CO3	Explain the components of Hadoop and Hadoop Eco-System and apply them in developing solutions for Big Data analysis
CO4	Explain the applications on Big Data using Pig and Hive

CO - PSO Mapping							
	PSO1	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**	**PSO7**
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction to Big Data Platform – Challenges of Conventional Systems - Intelligent data analysis –Nature of Data - Analytic Processes and Tools - Analysis vs Reporting - Modern Data Analytic Tools -Statistical Concepts: Sampling Distributions - Re-Sampling - Statistical Inference - Prediction Error.

Module 2: 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.

Module 3: 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 Reduce Job Run-Failures-Job Scheduling-Shuffle and Sort – Task execution - Map Reduce Types and Formats - 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- Hadoop in the cloud.

Module 4: Applications on Big Data Using Pig and Hive – Data processing operators in Pig – Hive services – HiveQL – Querying Data in Hive - fundamentals of HBase and Zookeeper - IBM InfoSphereBigInsights and Streams. Visualizations - Visual data analysis techniques, interaction techniques, Systems and applications.

Core Compulsory Readings

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
2. Tom White, Hadoop: The Definitive Guide, 3rdEdn, O'reily Media, 2012.

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Demonstration, Questioning and Answering

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Describe the nature of data in the context of big data platforms.
2. How does re-sampling contribute to statistical analysis?
3. Explain the concept of stream computing and its role in real-time data analysis.
4. Discuss the importance of filtering streams in stream data processing.
5. How can you count distinct elements in a stream? Describe any relevant algorithms or approaches.
6. What are moments in stream data analysis? How can they be estimated?
7. Explain the concept of counting oneness in a window in the context of stream processing.
8. Discuss case studies related to real-time sentiment analysis and stock market predictions using stream data analysis techniques.
9. How can data be analyzed using Hadoop? Discuss the key steps and tools involved in the process.
10. How does Hadoop Streaming enable the integration of non-Java programming languages with the Hadoop framework?
11. Describe the design of HDFS and its key characteristics.
12. How can you develop a MapReduce application? Explain the basic steps involved.
13. Provide an overview of how MapReduce works, including its key phases and the role of mappers and reducers.
14. How does Hadoop handle failures in a MapReduce job? Discuss the fault tolerance mechanisms in place.
15. Explain the concepts of job scheduling, shuffle and sort, and task execution in the context of Hadoop's MapReduce framework.
16. Discuss the benchmarks commonly used to evaluate the performance of Hadoop clusters.
17. How can Pig and Hive be used for applications on big data? Discuss their respective roles and functionalities.
18. Provide an overview of IBM Info Sphere BigIn sights and Streams. What are their key features and use cases?

SEMESTER IV
DSE COURSE
MCCSA04DSE13 Operations Research

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description:

Operations Research (OR) is a discipline that involves the development and application of advanced analytical methods to aid complex decisions. This course will provide students with the skills to be able to apply a variety of analytical methods to a diverse set of applications.

Course Objectives:

- To impart knowledge in concepts and tools of Operations Research
- To understand mathematical models used in Operations Research
- To apply these techniques constructively to make effective business decisions

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

SL #	Course Outcomes
CO1	Explain the applications of operations research and linear programming
CO2	Explain various problems such as transportation problem, assignment problem, traveling salesperson problem, integer Programming problem (IPP), and dynamic programming (DPP) problem
CO3	Explain various aspects of sequencing problem and project scheduling
CO4	Illustrate the fundamentals of simulation

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module X (For Additional Reading and Comprehension by the Students):
Managerial Applications and Limitations of OR. Linear and Non- Linear, Integer, Goal [Multi- Objective] and Dynamic Programming Problems (Emphasis is on Conceptual Frame Work-no Numerical Problems). Queuing Theory, Concepts of Queue/Waiting Line, General Structure of a Queuing System, Operating Characteristics of queues, Deterministic queuing models, Probabilistic Queuing Model, Cost Analysis. Single Channel Queuing Model, Poisson Arrival and Exponential Service Times with Infinite Population. Game Theory, Concepts, Saddle Point, Dominance, Zero-Sum Game, Two, Three and More Persons Games, Analytical Method of Solving Two Person Zero Sum Games, Graphical Solutions for (m x 2) and (2 x n) Games.

Core Compulsory Readings

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

Core Suggested Readings

1. Hillier, Frederick S. & Lieberman, Introduction to Operations Research Concepts and Cases, 2010, 8th Ed. TMH
2. N.D. Vohra, Quantitative Techniques in Management, 2010, 4thEd.TMH.
3. J.K. Sharma, Operations Research Theory and Applications 2009, 4th Ed. McMillan.
4. Kasana, HS & Kumar, KD, Introductory Operations Research theory and Applications, 2008, Springer.
5. Chakravarty, P, Quantitative Methods for Management and Economics, 2009, 1st Ed. HPH.

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. What is linear programming?
2. Maximize profit: A company produces two products, A and B, with profit margins of $10 and $15 per unit, respectively. Product A requires 2 hours of labor and 3 units of raw material, while product B requires 4 hours of labor and 2 units of raw material. The company has 100 hours of labor and 60 units of raw material available. How many units of each product should the company produce to maximize profit?
3. A manufacturing company produces two products, A and B. Product A requires 4 hours of labor and 2 units of raw material, while product B requires 3 hours of labor and 5 units of raw material. The company has 200 hours of labor and 150 units of raw material available. Product A sells for $8 per unit, and product B sells for $10 per unit. Determine the production quantities of A and B that minimize the total production cost using the simplex method.
4. A salesperson needs to visit a set of cities and return to the starting city, covering the shortest possible distance. The distances between cities are given in the table below. Find the shortest tour that visits all cities exactly once and returns to the starting city.

	City 1	**City 2**	**City 3**	**City 4**
City 1		10	15	20
City 2	10	-	35	25
City 3	15	35	-	30
City 4	20	25	30	-

SEMESTER IV
DSE Course
MCCSA04DSE14 Algorithms in Computational Biology

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

Course Description: The course provides an account about the algorithms in Computational Biology. String matching algorithms, which play a remarkable role in Computational Biology, is discussed in detail. The course also introduces sequence alignment in detail.

Course Objectives:

- To understand the algorithms in Computational Biology
- To familiarize with the application of string matching algorithms
- To provide advance knowledge about sequence alignment
- To understand the significance of phylogenetic trees

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the basic algorithms in Computational Biology
CO2	Illustrate the application of string matching algorithms and data matrices
CO3	Illustrate the process of sequence alignment
CO4	Explain the basics of multiple sequence alignment

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Basic Algorithms in Computational Biology: Exhaustive search methods and their applications in Computational Biology - Motif finding - Tandem repeats.

Module 2: 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.

Module 3: Sequence Alignment: Pairwise 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.

Module 4: Multiple sequence alignment: n dimensional dynamic programming. Tools for MSA: Muscle and T - Coffee. Phylogenetic Algorithms: Clustering based methods - UPGMA and neighbour joining, Optimality based: Fitch Margoliash and minimum evolution algorithm; Character based methods - Maximum Parsimony and Maximum Likelihood methods; Evaluation of phylogenetic trees - significance.

Core Compulsory Readings

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

Core Suggested Readings

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
6. Sushmita M and Tinku A, Data Mining Multimedia, Soft Computing and Bioinformatics, John Wiley & Sons, Inc., 2003
7. 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

TEACHING LEARNING STRATEGIES
Lecturing

MODE OF TRANSACTION
Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE15 Object Oriented Analysis and Design

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course focuses on providing the fundamentals of object oriented analysis and design. A discussion about various design and architectural diagrams is also included.

Course Objectives:

- To understand the basic principles of Object Oriented Systems
- To familiarize with the Unified Modeling Language (UML) and various design and architectural diagrams
- To understand the concept of components
- To learn Impact of object orientation on testing

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

SL #	Course Outcomes
CO1	Describe the basic principles of Object Oriented Systems and various types of class diagrams
CO2	Illustrate various kinds of architecture diagrams and design diagrams
CO3	Illustrate the principles of encapsulation structure
CO4	Illustrate various topics such as abuses of inheritance, danger of polymorphism, mix-in classes, rings of operations, components and objects, software quality assurance and developing test cases

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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 operation. Class diagrams: generalization, association, 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.

Module 2: Architecture diagrams: packages, deployment diagrams for hardware artifacts and software constructs. Interface diagrams: window-layout and window-navigation diagrams.

Module 3: Encapsulation structure, connascence, domains of object classes, encumbrance, class cohesion, state-spaces and behaviour of classes and subclasses, class invariants, pre-conditions and post-conditions, class versus type, principle of type conformance, principle of closed behaviour.

Module 4: Abuses of inheritance, danger of polymorphism, mix-in classes, rings of operations, components and objects, design of a component, lightweight and heavyweight components, advantages and disadvantages of using components. Software Quality Assurance – Impact of object orientation on Testing – Develop Test Cases and Test Plans.

Core Compulsory Readings

1. Meilir Page-Jones, Fundamentals of object-oriented design in UML, Addison Wesley.
2. Booch. G, Rumbaugh J, and Jacobson. I, The Unified Modeling Language User Guide, Addison Wesely.

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. State about basic state diagram
2. Consider a House keeping system in a five star hotel. Draw a sequence and collaboration diagram for a scenario
3. Explicate in detail about components, their types and their pros and cons

SEMESTER IV
DSE Course
MCCSA04DSE16 Computer Vision

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

- To introduce and discuss the basic concepts of computer vision techniques
- Understandings on the applications of machine vision
- To accomplish knowledge about motion analysis

Course Outcomes:

SL #	Course Outcomes
CO1	Describe the importance and fundamentals of computer vision
CO2	Explain various aspects of object recognition
CO3	Illustrate various aspects of 3D vision
CO4	Illustrate the applications of motion analysis techniques

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: Object Recognition: Knowledge representation, Statistical pattern recognition, Neural nets, Syntactic pattern recognition, Recognition as graph matching, Optimization techniques in recognition, Fuzzy systems.

Module 3: 3D vision: 3D vision: Tasks - Basics of projective geometry - Scene construction from multiple views, Uses: Shape from X - Full 3D objects - 3D model based vision - 2D view based 3D representation.

Module 4: 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.

Core Compulsory Readings

1. Szeliski, Richard, Computer Vision: Algorithms and Applications- 2nd Edn, Springer Verlag, 2022.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis and Machine Vision, Cengage Learning, New Delhi, 2014.
3. Wesley E. Synder and Hairong Qi, Machine Vision, Cambridge University Press, USA, 2010.

Core Suggested Readings

1. Rafael C Gonzalez, Richard E Woods, Steven L Eddins, Digital Image Processing, Pearson Education, New Delhi, 2009.
2. Shapiro L and Stockman G, Computer Vision, Prentice-Hall, 2001.

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. How do humans recognize objects and how can we replicate this process in machines?
2. What are some applications of projective geometry in 3D vision, and how do they impact industries such as robotics, autonomous vehicles, and augmented reality?
3. What are some challenges faced when working with 2D view based 3D representations?
4. How can motion models be adapted to different tracking scenarios, such as in crowded environments, fast-moving objects, or occluded scenes?
5. What are the different types of video tracking algorithms, and how do they differ in terms of accuracy, speed, and robustness?
6. How can machine learning be used to improve the accuracy and efficiency of motion pattern detection in computer vision? Discuss the advantages and limitations of deep learning, supervised learning, and unsupervised learning methods in this context.
7. What are the main applications of motion pattern detection in computer vision, and how can they be applied in real-world scenarios? Discuss the challenges associated with deploying these methods in real-world applications.
8. What are the potential future directions of motion models in tracking, such as in 3D tracking or dynamic scene analysis? Discuss the need for novel algorithms and hardware advancements to realize these directions.
9. What is the role of camera calibration in multi-view stereo reconstruction, and how can it be performed accurately?
10. How can optical flow be used for motion analysis and object tracking, and what are the main challenges in these tasks?

SEMESTER IV
DSE Course
MCCSA04DSE17 Software Project Management

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course focuses on providing the fundamentals of object oriented analysis and design. A discussion about various design and architectural diagrams is also included.

Course Objectives:

- To understand the importance of Software Project Management (SPM)
- To familiarize with various aspects of planning software projects
- To appreciate the significance of Software Quality Assurance

Course Outcomes:

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

SL #	Course Outcomes
CO1	Describe the fundamentals of SPM and its need
CO2	Illustrate various aspects of planning software projects
CO3	Explain the dimensions of project monitoring & control and, earned value analysis
CO4	Explain the concepts of software quality assurance

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

1. M. Manish Kumar Jha, Software Project Management, Dhanpat Rai & Co.

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. How do humans recognize objects and how can we replicate this process in machines?
2. What are some applications of projective geometry in 3D vision, and how do they impact industries such as robotics, autonomous vehicles, and augmented reality?
3. What are some challenges faced when working with 2D view based 3D representations?
4. How can motion models be adapted to different tracking scenarios, such as in crowded environments, fast-moving objects, or occluded scenes?
5. What are the different types of video tracking algorithms, and how do they differ in terms of accuracy, speed, and robustness?
6. How can machine learning be used to improve the accuracy and efficiency of motion pattern detection in computer vision? Discuss the advantages and limitations of deep learning, supervised learning, and unsupervised learning methods in this context.

7. What are the main applications of motion pattern detection in computer vision, and how can they be applied in real-world scenarios? Discuss the challenges associated with deploying these methods in real-world applications.

8. What are the potential future directions of motion models in tracking, such as in 3D tracking or dynamic scene analysis? Discuss the need for novel algorithms and hardware advancements to realize these directions.

9. What is the role of camera calibration in multi-view stereo reconstruction, and how can it be performed accurately?

10. How can optical flow be used for motion analysis and object tracking, and what are the main challenges in these tasks?

SEMESTER IV
DSE Course
MCCSA04DSE18 Visual Cryptography

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description:

The Internet is the fastest growing communication medium and essential part of the infrastructure nowadays. To cope with the growth of the internet, it has become a constant struggle to keep the secrecy of information as and when profits are involved. It is termed as protecting the copyright of data. To provide secrecy and copyright of data, many of the steganographic techniques have been developed. Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image. It operates by splitting an image or text into multiple shares, such that when the shares are overlaid, the original image or text becomes visible.

Course Objectives:

- To learn about the internal representation of digital images
- Familiarize various image models and its usage
- Acquire the knowledge of digital image cryptography
- To learn the basic mathematics for secret sharing
- Attain the knowledge of encoding and decoding digital image for hiding useful information
- Familiarize the basic color image encryption and decryption algorithms

Course Outcomes:

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

SL #	Course Outcomes
CO1	Describe the fundamentals of image models, image representation and types of images
CO2	Illustrate the principles of steganography and digital watermarking and their applications
CO3	Explain various aspects of the real-world applications of Visual Cryptography
CO4	Explain various aspects of Color Visual Cryptography

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔		✔		✔	✔	✔
CO2	✔		✔		✔	✔	✔
CO3	✔		✔		✔	✔	✔
CO4	✔		✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module3: 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-ofn Visual Cryptography Schemes, Basis Matrices for k-out-of-n Visual Cryptography.

Module 4: 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.

Core Compulsory Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Visualization, Team Learning, Digital Learning.

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Demonstration, Questioning and Answering, Audio, Video, Printed note

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. What is digital image?
2. Explain the representation of digital images.
3. What is image sampling?
4. What is image quantization?
5. What are the different types of images?
6. What is binary image?
7. What is gray scale image?
8. What is intensity image? Explain its representation.
9. What is color image?
10. What is RGB?
11. What are the different color models?
12. What CMY model?
13. What is CMY-K model?
14. What is HIS model?
15. Explain about different image file formats.
16. What is steganography?
17. What is digital watermarking?
18. What is secret sharing?
19. Explain the principles of secret splitting?
20. What are different phases of secret sharing scheme? Explain.
21. What are basis matrices?
22. Explain 2-out-of-2 visual cryptography schemes.
23. Explain k-out-of-n visual cryptography schemes.
24. Explain the construction of 2-out-of-2 color visual cryptography schemes.
25. Explain the applications of visual cryptography schemes.

SEMESTER IV
DSE Course
MCCSA04DSE19 Biometric Image Processing

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description:

The course is designed to provide the fundamental concepts of Biometric Image Processing. Topics such as image enhancement, Hand and Iris Biometrics, and Morphological image processing are included in the discussion.

Course Objectives:

- To understand the fundamental concepts of Biometric Image Processing
- To acquire knowledge about Image enhancement
- To gain knowledge about Hand and Iris Biometrics
- To familiarize with Morphological image processing

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the fundamentals of Biometric Image Processing
CO2	Describe various techniques such as image enhancement and image segmentation
CO3	Illustrate the concepts of Morphological image processing and face recognition techniques
CO4	Explain various aspects of hand and Iris biometrics

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: Image enhancement: Spatial domain methods: Basic gray 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

1. Rafael C Gonzalez, Richard E Woods and Steven L Eddins, Digital Image Processing, Pearson Education, New Delhi, 2013.

Core Suggested Readings

1. Amine Nait Ali and Regis Fournier, Signal and Image Processing for Biometrics, John Wiley and Sons, UK, 2012
2. Arun A Ross, Karthik Nandakumar and Jain A K, Handbook of Multibiometrics, Springer, New Delhi 2011
3. Oge Marques, Practical Image and Video Processing using MATLAB, John Wiley and Sons, New Jersey, 2011

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Semester IV | Elective III I End of POOL C

SEMESTER IV
DSE Course
MCCSA04DSE20 Nature Inspired Computing

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course provides an insight into various inspirations provided by nature and how these inspirations can be used to solve real life problems.

Course Objectives:

- To understand natural inspirations in problem solving
- To acquire knowledge about Ant Colony Optimization
- To gain knowledge about Swarm Intelligence
- To learn genetic algorithms
- To familiarize with DNA Computing

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the inspirations provided by nature and illustrate how they can be used to solve real life problems
CO2	Illustrate Ant Colony Optimization and Swarm Intelligence
CO3	Describe various aspects of genetic algorithms
CO4	Explain DNA Computing

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Introduction: Nature Inspired Computing. Natural to Artificial Systems - Biological Inspirations in problem solving- Behaviour of Social Insects: Foraging - Division of labor - Task Allocation – Cemetery Organization and Brood Sorting – Nest Building – Cooperative Transport.

Module 2: Ant Colony Optimization : Ant Behaviour – 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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 Systems, 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

Core Suggested Readings

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, 200

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE21 Pattern Recognition

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course is designed to provide the basics and various techniques of pattern recognition. The course provides an insight to various clustering algorithms. A discussion about soft computing techniques for Pattern Recognition.

Course Objectives:

- To understand the basics of Pattern Recognition
- To familiarize with various techniques of Pattern Recognition
- To understand the various algorithms for clustering
- To create an awareness about soft computing techniques for Pattern Recognition

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

SL #	Course Outcomes
CO1	Explain the fundamentals of Pattern Recognition and various techniques belongs to it
CO2	Illustrate various algorithms for clustering and feature selection
CO3	Explain various aspects of feature selection
CO4	Explain the concepts of soft computing techniques for pattern recognition

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: Min - max and Neymann - Pearson classifiers, Discriminant functions, Decision surfaces. Maximum likelihood estimation and Bayesian parameter estimation. Overview of Nonparametric density estimation - Histogram based approach, classification using Parzen window. K - nearest neighbour estimation and classification. Classification of clustering algorithm - hierarchical clustering - agglomerative clustering. Partitional clustering Forgy's algorithm. K-means clustering.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

1. Duda R.O., and Har P.E., Pattern Classification and Scene Analysis, Wiley, New York, 1973.

Core Suggested Readings

1. Bishop C. M, Pattern recognition and machine learning, Springer, 2nd Edition, 2006
2. Theodoridis .S, Pikrakis .A, Koutroumbas .K, Cavouras .D, Introduction to Pattern Recognition: A Matlab approach, Academics Press 2010

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE22 Cyber Forensics

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: This course is to provide students with basic knowledge of computer forensics. This course deals with the significance of Computer forensics, Computer Forensics analysis and validation techniques.

Course Objectives:

- To provide basic knowledge of Computer forensics.
- To understand the principles, techniques, and methodologies involved in conducting investigations and forensic analysis of digital evidence in cybercrime cases.
- To know the Computer Forensics analysis and validation techniques.
- To explain Computer Forensic tools and mobile device forensics.

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the fundamentals of computer forensics
CO2	Illustrate principles of digital evidence collection and preservation
CO3	Explain various aspects of analyzing and validating forensics data
CO4	Explain the legal and ethical aspects of cyber forensics

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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, copyrights issues with graphics. Understanding data compression, identifying unknown file formats.

Module 3: 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.

Module 4: 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 email crimes and violations, understanding email 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.

Core Compulsory Readings

1. John 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.

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE23 Natural Language Processing with Python

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course is designed to provide the fundamental concepts of natural language processing using Python. Topics such as text processing of raw text, text categorization, information extraction from text and analysis of sentence structure are included in the discussion. Students are also exposed to developing solutions using Python for various operations related to NLP.

Course Objectives:

- To understand the fundamental concepts of natural language processing (NLP) with Python
- To acquire knowledge in text categorization using Python
- To gain knowledge and expertise in top NLP applications
- To create an awareness about information extraction from natural language text

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

SL #	Course Outcomes
CO1	Explain the fundamentals of NLP using Python
CO2	Illustrate various aspects of text categorization using Python
CO3	Explain various aspects of categorizing and tagging words
CO4	Explain the concepts of information extraction from natural language text.

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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/.

Core Suggested Readings

1. BDipanjan 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_RealWorld_Approach_to_Gaining_Actionable_Insights_from_Your_Data_Dipanjan_Sarkar

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE24 Grid and Cloud Computing

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: The course is designed to provide the fundamental concepts of natural language processing (using Python. Topics such as text processing of raw text, text categorization, information extraction from text and analysis of sentence structure are included in the discussion. Students are also exposed to developing solutions using Python for various operations related to NLP.

Course Objectives:

- To understand the fundamental concepts of Distributed Computing, Grid Computing and its large scale scientific applications
- To acquire knowledge on the concept of virtualization
- To gain knowledge about different cloud deployment models and cloud computing categories
- To learn implementations of open source grid middleware packages and Hadoop Framework
- To understand the security issues in the grid and the cloud environment

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

SL #	Course Outcomes
CO1	Explain the fundamentals of Distributed Computing, Grid Computing and its large scale scientific applications
CO2	Describe different cloud deployment models and cloud computing categories
CO3	Illustrate the implementations of open source grid middleware packages and Hadoop Framework
CO4	Identify the security issues in the grid and the cloud environment

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔		✔	✔	✔	✔
CO2	✔	✔		✔	✔	✔	✔
CO3	✔	✔		✔	✔	✔	✔
CO4	✔	✔		✔	✔	✔	✔

COURSE CONTENTS

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Demonstration

MODE OF TRANSACTION

- Lecture, Seminar, Discussion

ASSESSMENT RUBRICS

Components	Percentage
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE25 Information Security

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description:

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

Course Objectives:

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

Course Outcomes:

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

SL #	Course Outcomes
CO1	Explain the foundations of encryption and decryption using classical encryption algorithms
CO2	Describe various aspects of block cipher
CO3	Illustrate various aspects of public key cryptography
CO4	Explain various aspects of message authentication

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔	✔	✔	✔	✔
CO2	✔	✔	✔	✔	✔	✔	✔
CO3	✔	✔	✔	✔	✔	✔	✔
CO4	✔	✔		✔	✔	✔	✔

COURSE CONTENTS

Module 1: Foundations of Cryptography and security: Ciphers and secret messages, security attacks and services.Classical Encryption techniques - Symmetric cipher model, substitution techniques, transposition techniques, steganography. Basic Concepts in Number Theory and Finite Fields.

Module 2: Block cipher principles – The data encryption standard (DES) – strength of DES – Differential and linear cryptanalysis – Block cipher design principles. Advanced encryption standard – AES structure – AES transformation function – key expansion – implementation. Block cipher operations –Multiple encryption – ECB – CBC – CFM – OFM – Counter mode. Pseudo Random Number generators - design of stream cipher, RC4.

Module 3: Public Key cryptography: Prime numbers and testing for primality, factoring large numbers, discrete logarithms. Principles of public-key crypto systems – RSA algorithm. Diffi-Helman Key exchange, Elgammal Cryptographic systems - Hash functions – examples – application – requirements and security – Hash function based on Cipher block chaining – Secure Hash algorithm.

Module 4: 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.

Core Compulsory Readings

1. William Stallings, Cryptography and Network Security, Pearson 2004

Core Suggested Readings

1. Foorouzan and Mukhopadhyay, Cryptography and Network security, 2ndedn
2. BuceSchneier., Applied cryptography – protocols and algorithms, Springer

Verlag 2003
3. William Stallings, Network Security Essentials, , 4thedn, Pearson
4. Pfleeger and Pfleeger, Security in Computing, 4thEdn, Pearson

TEACHING LEARNING STRATEGIES

- Lecturing, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

SEMESTER IV
DSE Course
MCCSA04DSE26 Data and Information Visualization

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

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

Course Outcomes:

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

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔			✔	✔	✔
CO2	✔	✔			✔	✔	✔
CO3	✔	✔			✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: Data and types of data, Data variability, uncertainty and context. Basics of Data Visualization: Definition of Data Visualization, Need for Visualization, How to visualize data, General types of Data Visualization, pros and Cons of Data Visualization. Visualization Components: Visual cues, Coordinate systems, Scales, and Context. Diagrams used for data visualization: Bar chart, Histogram, Scatter plot, Scatter plot, Network, Streamgraph, Tree map, Gantt chart, Stripe graphic, Animated spiral graphic. Visualization based on types of data: Visualizing Categorical data, Visualizing Time series data, Visualizing Spatial data.

Module 2: Information Visualization: Definition, Objectives of Information Visualization. Visual representation of large-scale collection of non - numerical information. Design Principles of Information Visualization: Principle of Simplicity, Principle of Proximity, Principle of Similarity, Principle of Closure, Principle of Connectedness, Principle of Good Continuation, Principle of Common fate, Principle of Familiarity, Principle of Symmetry.

Module 3: Methods for Information Visualization: Cartogram, Cladogram (phylogeny), Concept Mapping, Dendrogram (classification). Graph drawing, Heat map, Hyperbolic Tree, Tree mapping Multidimensional scaling. Information visualization reference model. Case study with real world problems.

Module 4: 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.

Core Compulsory Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Design a graph and describe the elements of data visualization with the help of it.
2. How do you explain positive and negative correlation in scatter plots.
3. Summarize design principles behind information visualization.
4. Design a cladogram and mark its parts.
5. Show the conversion from tree diagram to treemap with the help of an example diagram
6. Explain different scientific visualization techniques.
7. Discuss Any one surface rendering method.
8. Write a python program to plot $y=x$ and $y=x^2$ as subplots.

SEMESTER IV
DSE Course
MCCSA04DSE27 Information Retrieval System

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

Course Description: This course is to introduce fundamental concepts of information retrieval and information retrieval strategies. Emphasis is given to Distributed Information Retrieval and Information Retrieval algorithms. Students will learn about k-gram indexes for spelling correction.

Course Objectives:

- To understand the fundamentals of information retrieval
- To apply machine learning techniques for text classification and clustering
- To use information retrieval strategies
- To learn different Information retrieval algorithms

Course Outcomes:

SL #	Course Outcomes
CO1	Explain the objectives and foundations of information retrieval systems
CO2	Describe various aspects of retrieval utilities
CO3	Illustrate natural language systems to build semantic networks for text and the measures to evaluate the performance of cross language information
CO4	Explain various aspects of integrating structured data and text and distributed information retrieval

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔			✔	✔	✔
CO2	✔	✔			✔	✔	✔
CO3	✔	✔			✔	✔	✔
CO4	✔	✔			✔	✔	✔

COURSE CONTENTS

Module 1: Introduction: Retrieval strategies: vector space model, Probabilistic retrieval strategies: Simple term weights, Non binary independence model, Language models.

Module 2: Retrieval Utilities: Relevance feedback, clustering, N-grams, Regression analysis, Thesauri.

Module 3: Semantic networks, parsing Cross –Language: Information Retrieval: Introduction, Crossing the Language barrier.

Module 4: 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.

Core Compulsory Readings

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

Core Suggested Readings

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

TEACHING LEARNING STRATEGIES

- Lecturing

MODE OF TRANSACTION

- Lecture, Seminar, Discussion, Questioning and Answering

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Write a note on Taxonomy of Information retrieval Models.
2. Discuss in detail about User Relevance Feedback.
3. Explain in detail about components of the Information Retrieval System.
4. What are n-grams in the context of natural language processing?
5. What is cross-language information retrieval (CLIR) and why is it important?
6. What are the different types of clustering algorithms used in information retrieval?
7. What are some common regression models used in information retrieval?
8. What are the advantages and limitations of using thesauri in information retrieval systems?

SEMESTER IV
DSE Course
MCCSA04DSE28 Design and Analysis of Algorithms

Credit			Teaching Hours			Assessment		
L/T	P/I	Total	L/T	P/I	Total	CE	ESE	Total
4	0	4	4	0	4	40	60	100

Lecture/Tutorials, P/I=Practical/Internship, CE =Continuous Evaluation, ESE = End Semester Evaluation

COURSE CONTENTS

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

Course Objectives:

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

Course Outcomes:

SL #	Course Outcomes
CO1	Explain the objectives and foundations of design and analysis of algorithms
CO2	Illustrate various aspects of analyzing algorithms
CO3	Illustrate various complexity classes
CO4	Explain various aspects of the design and analysis of parallel algorithms

CO - PSO Mapping							
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7
CO1	✔	✔	✔		✔	✔	✔
CO2	✔	✔	✔		✔	✔	✔
CO3	✔	✔	✔		✔	✔	✔
CO4	✔	✔	✔		✔	✔	✔

COURSE CONTENTS

Module 1: 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).

Module 2: 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.

Module 3: 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.

Module 4: 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.

Core Compulsory Readings

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.

Core Suggested Readings

1. Pandey H M, Design and Analysis of Algorithms, University Science Press, 2013
2. Upadhyay, N, Design and Analysis of Algorithms, Sk Kataria & Sons, 2008.
3. U. Manber, Introduction to Algorithms: A Creative Approach, Addison Wesley
4. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice-Hall of India
5. Goodman S E and Hedetniemi, Introduction to the Design and Analysis of Algorithms, Mcgraw Hill
6. Horowitz E and Sahni S, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd
7. Oded Goldreich, P,NP and NP- Completeness, Cambridge University Press, 2011.
8. Donald Knuth, The Art of Computer Programming, Fundamental Algorithms, Volume - 1, Addison Wesley, 1997.
9. Sanjeev Arora and Boaz Borak, Computational Complexity- A Modern Approach, Cambridge University Press; 2009.
10. Daniel Hills W and Bruce M Boghosian, Parallel Scientific Computation, Science, Vol 261, pp. 856-863

TEACHING LEARNING STRATEGIES

- Lecturing, Team Learning, Digital Learning

MODE OF TRANSACTION

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

ASSESSMENT RUBRICS

Components	**Percentage**
End Semester Evaluation	**60**
Continuous Evaluation	**40**
Tests	40
Assignment	20
Seminar	40

Sample Questions to test Outcomes.

1. Differentiate Direct Recursion and Indirect Recursion.
2. Solve the recurrence **$T(n) = 2T(\sqrt{n}) + \log n$ 2.**
3. Compute the time complexity of the following:

1.

```
for i  100 to m-1
{ for j  10 to i
  {
    A  B + C[i][j]
  }
     }
```

4. Let **$T(n) = 4\ T\ (n/2) + n^3$**, then show that **$f(n) = \Omega(n^3)$** and **$T(n) = \theta(n^3)$.**
5. Prove that Hamiltonian Cycle is NP Complete.
6. How can we solve Knapsack problem using Branch-and-Bound technique?
7. Given a set S = **{2, 4, 6}** and Weight = **6**. Find subset sum using backtracking approach.
8. Let **H(t)** be the number of multiplications in the following:

```
int Factorial( int t)
 {
 if (t = = 0)
```

then return 1
else
return **t*** Factorial(**t**-1)

2. }
3. Prove that **H(t) = t.**
4. 9. 'The running time is directly proportional to the frequency count of the algorithm.' Explain the meaning of the statement in detail.
5. 10. The recurrence **$T(n) = 7T(n/2) + n^2$**describes the running time of an algorithm **A**. A competing algorithm **A^I** has a running time of **$T^I(n)= kT^I(n/4)+n^2$.** What is the largest integer value for **k** such that **A^I** is asymptotically faster than **A**?
6. 11. What do mean by Parallel Prefix Computation?
7. 12. Can the master theorem be applied to the recurrence **$T(n) = 2T(n/2) + n \log n$?** Why or Why not?

End of Semester IV