KANNUR UNIVERSITY

(Abstract)

Master of Computer Applications (MCA) Programme in the Department of Information Technology, Mangattuparamba Campus - Revised Scheme and Syllabus(I st Semester only)- Approved-Implemented w.e.f. 2023 admission--Orders issued

---

ACADEMIC C SECTION

ACAD C/ACAD C1/23379/2023 Dated: 13.11.2023

Read:-1. U. O. No. ACAD C/ACAD C3/22373/2019 dtd.12.09.2023

2. Circular No. dated ACAD C/ACAD C3/22373/2019 dated 12/09/2023

3. Email dated 08.11.2023 from the Head, Department of Information Technology, Mangattuparamba Campus.

4.Minutes of the meeting of the Department Council held on 04.10.2023

ORDER

1.The revised Regulations for Post Graduate Programmes under Choice Based Credit and Semester System in the University Teaching Departments/Schools were implemented w.e.f. 2023 admissions vide paper read(1) above.

2. As per paper read (2) above, Heads of all Teaching Departments were requested to submit the revised Syllabus in accordance with the approved regulations along with a copy of the Department Council Minutes

3. As per paper read (3) above, the Head, Department of Information Technology, Mangattuparamba Campus submitted the Scheme & Syllabus (Ist Semester only) of Master of Computer Applications Programme to be implemented in the University Teaching Department w. e. f. 2023 admissions.

4. Department Council vide the paper read (4) above approved the aforementioned Scheme & Syllabus of MCA Programme to be implemented in the Dept. of Information Technology of the University w. e. f. 2023 admission.

5.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 **Scheme & Syllabus (I st Semester only) of Master of Computer Applications (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.**

6.The Scheme & Syllabus (I st Semester Only) of Master of Computer Applications(M.C.A) 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)

7.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 * THAVAKKARA * CIVIL STATION P.O. KANNUR Pin-670 002

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

PROGRAMME SPECIFIC OUTCOMES

SL #	Outcome
PSO1	Familiar with the entrenched concepts of Computer Science and Applications
PSO2	Enhance the knowledge and skills about System Software and Application Software
PSO3	Attain skills to design Algorithms and Programs
PSO4	Acquire the knowledge in the building, designing and managing IT and IT enabled infrastructure
PSO5	Design, build, and test software systems to meet the given specifications by following the principles of Software Engineering
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 have 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 in 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 choose a MOOC in the third semester. The department council will prepare a list of MOOCs from the online courses offered by the NPTEL / any other MOOC course provider approved by the council from time to time at the beginning of the third semester. 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 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	% 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. BoE will 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.
4.

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's theorms - Simplifications using Boolean expressions and truth tables- Karnaugh map- SOP and POS minimizations- Simplification of Boolean expression using K-Map (up to four variables) – (8 Hours)

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. Numerical Integration:- Trapezium rule, Trapezoidal rule, Simpson's rule. Differential equations:

Preliminaries, Taylor series method, RungeKutta methods-Statistical description and modelling of data. (08 Hours)

Text Books:

1. Thomas L 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
TEC	Technology Enhancement Course

Course Structure

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

*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 C
1.5c	MCCSA01DSE03	Principles of Programming using CPP

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**			

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.	IDE/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 Course (offered for other Departments)

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 implementation 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 MethodVector 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 Programs that uses some reserved ports, 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, I, 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. File system configuration: i-nodes, superblocks, ext3, reiserFS, mounting and unmounting local disks, /etc/fstab file. 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, Host name Resolution Name lookup with DNS, Reverse Lookup, 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 (3 marks)
2. State what is meant by LILO boot process (3 marks)
3. Identify how to address a socket (3 marks)
4. State the need for DHCP (3 marks)
5. Differentiate between switch and hub (5 marks)
6. Prepare short notes on managing links in Linux (5 marks)
7. Differentiate: Half Duplex Unix Pipes and Named Pipes (5 marks)
8. Differentiate: FTP and TFTP (5 marks)
9. Explain OSI model (10 marks)
10. Illustrate IPC using UDP sockets (10 marks)
11. Explain interface configuration (10 marks)
12. Explain commands for files and directories (10 marks)

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 I I MCAIT01DSE (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)
MCAIT01DSE03 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.

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:
1
12
123
1234
12345
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