MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)
#14, Ramohalli Cross, Kumbalagodu, Mysore Road, Bengaluru–560074

Master of Computer Applications

Scheme and Syllabus of I Semester

(2024 Scheme)

VISION

To empower young minds through technology, research and innovation, to produce technically competent and socially responsible professionals in higher education.

MISSION

1. To deliver excellence in education through innovative teaching, impactful research, and continuous skill development, preparing students to meet global challenges with technical expertise and ethical responsibility.
2. To foster a transformative learning environment that integrates technology, research and practical experience, empowering students to become skilled professionals and socially conscious leaders.
3. To cultivate a culture of lifelong learning and professional excellence by encouraging creativity, research, and community engagement, equipping students with the skills to thrive in a dynamic world.
4. To provide a holistic educational experience that combines advanced technology, hands-on research, and community-focused learning, shaping students into competent, ethical professionals who contribute positively to society.

QUALITY POLICY

Rajarajeswari College of Engineering is committed to imparting quality technical education that nurtures competent, ethical professionals with global relevance. We ensure academic excellence through a dynamic, outcome-based curriculum, experienced faculty, and cutting-edge infrastructure. Continuous improvement is driven by innovation, research and strong industry collaboration. We foster holistic development and a progressive environment that supports lifelong learning, teamwork, and professional growth.

CORE VALUES

Academic Excellence, Integrity, Innovation, Global Competence, Continuous Improvement.

INDEX

MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Master of Computer Applications

Scheme of Teaching and Examinations – 2024

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

(Effective from the Academic Year 2025-26)

I Semester

S. No	Course Category and Course Code		Course Title	TD/PSB	Teaching Hours/ Week			Examination				
					Lecture	Tutorial/ SDA	Practical/ Seminar	SEE Duration Hours	CIE Marks	SEE Marks	Total Marks	Credits
					L	T/S	P					
1.	**BSC**	**P24MCA101**	Mathematical Foundation for Computer Applications	MCA	**3**	**2**	**0**	**3**	**50**	**50**	**100**	**4**
2.	**IPCC**	**P24MCA102**	Operating Systems with Linux Programming	MCA	**3**	**0**	**2**	**3**	**50**	**50**	**100**	**4**
3.	**PCC**	**P24MCA103**	Data Structures and Algorithms	MCA	**3**	**0**	**0**	**3**	**50**	**50**	**100**	**3**
4.	**PCC**	**P24MCA104**	Software Engineering	MCA	**3**	**0**	**0**	**3**	**50**	**50**	**100**	**3**
5.	**PCC**	**P24MCA105**	Computer Networks	MCA	**3**	**0**	**0**	**3**	**50**	**50**	**100**	**3**
6.	**PCCL**	**P24MCAL106**	Data Structures and Algorithms Lab	MCA	**0**	**0**	**2**	**3**	**50**	**50**	**100**	**1**
7.	**PCCL**	**P24MCAL107**	Computer Networks Lab	MCA	**0**	**0**	**2**	**3**	**50**	**50**	**100**	**1**
9.	**AEC**	**P24MCA108A**	C Programming	MCA	**2**	**0**	**0**					**PP**
		P24MCA108B	Mathematics - I									
							TOTAL		**350**	**350**	**700**	**19**

Note: BSC-Basic Science Courses, PCC: Professional core. IPCC-Integrated Professional Core Courses, PCC: Professional Core Courses, PCCL-Professional Core Course lab, NCMC- Non Credit Mandatory Course, L-Lecture, T/SDA-Tutorial / Skill Development Activities, P-Practical (Hours are for Interaction between faculty and students)

Skill development activities: Under Skill development activities in a concerning course, the students should

1. Interact with industry (small, medium and large).
2. Involve in research/testing/projects to understand their problems and help creative and innovative methods to solve the problem.
3. Involve in case studies and field visits/fieldwork.
4. Accustom to the use of standards/codes etc., to narrow the gap between academia and industry.
5. Handle advanced instruments to enhance technical talent.
6. Gain confidence in the modeling of systems and algorithms for transient and steady-state operations, thermal study etc.
7. Work on different software/s (tools) to simulate, analyze and authenticate the output to interpret and conclude. All activities should enhance student's abilities to employment and/or self-employment opportunities, management skills, Statistical analysis, fiscal expertise, etc. Students and the course instructor/s are to be involved either individually or in groups to interact together to enhance the learning and application skills of the study they have undertaken. The students with the help of the course teacher can take up relevant technical activities that will enhance their skills. The prepared report shall be evaluated for CIE marks.

P24MCA305- Research Methodology and IPR- Non- Credit Mandatory Course (NCMC) if students have not studied this course in their undergraduate program then he /she has to take this course and complete the certification before the minimum duration of the program (Two years), however, this course will not be considered for vertical progression. This course will be evaluated in III semester.

Bridge Course: Non-Credit Mandatory Course **P24MCA108A-** C Programming: Students who have not taken Computer Subjects at the 10+2 or degree level are required to study and pass this course in the 1st semester. **P24MCA108B** – Mathematics-I: Students who have not taken Mathematics at the 10+2 or degree level are required to study and pass this course in the 1st semester. However, this course/subject will not be considered for vertical progression.

SEMESTER-I					
MATHEMATICAL FOUNDATION FOR COMPUTER APPLICATIONS **Category: BSC**					
Course Code	:	P24MCA101	CIE	:	50 Marks
Teaching Hours L : T : P	:	3:2:0	SEE	:	50 Marks
Total Hours	:	55	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3Hrs

Course Objectives	
1.	To introduce the concepts of sets, functions and perform the operations associated with sets and functions.
2.	Define and differentiate between discrete and continuous random variables, and understand their properties and examples.
3.	To introduce the concepts of mathematical logic.
4.	To introduce the concepts, relations and perform the operations.
5.	To introduce Graphs and use Graph Theory for solving problems.

Module - 1	**No. of Hours**
Basic Structures: Sets and subsets, Operations of Sets: Principle of Inclusion, Exclusion and Pigeonhole principle, Functions and Matrices: Eigenvalues and Eigenvectors.	11
Module - 2	**No. of Hours**
Random variable and Probability distribution: Concept of random variable, discrete probability distributions, continuous probability distributions, Mean, variance and Co-variance and co-variance of random variables. Binomial and normal distribution, Exponential and normal distribution with mean and variables and problems.	11
Module - 3	**No. of Hours**
Mathematical Logic: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences Predicates and Quantifiers, Nested Quantifiers, Rules of Inference Introduction to Proofs.	11
Module - 4	**No. of Hours**
Relations : Relations and Their Properties, n-ary Relations and Their Application, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.	11
Module - 5	**No. of Hours**
Graphs: Graphs and Graphs models, Graph Terminology and special types of graphs, Representing graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path problems, Planar Graphs, Graph Coloring.	11

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply the fundamentals of set theory and matrices for the given problem.
CO2	Apply the types of distribution; evaluate the mean and variance for the given case study/problem.
CO3	Solve the given problem by applying the Mathematical logic concepts.
CO4	Model the given problem by applying the concepts of graph theory.
CO5	Identify and list the different applications of discrete mathematical concepts in computer Applications

Text Books	
1.	Kenneth H Rosen, “Discrete Mathematics and its Applications”, McGraw Hill publications, 7th edition.
2.	Wolpole Myers Ye “Probability and Statistics for engineers and Scientist” Pearson Education, 8th edition.

Reference Text Books	
1.	Richard A Johnson and C.B Gupta “Probability and statistics for engineers” Pearson Education.
2.	J.K Sharma “Discrete Mathematics”, Mac Millian Publishers India, 3rd edition, 2011.

Web links and Video lectures (e-Resources)
• https://onlinecourses.nptel.ac.in/noc20_cs82/preview • https://www.nitt.edu/home/academics/departments/cse/programmes/mtech/curriculum/semester_2/mathematical_foundations_for_com/

ASSESSMENT DETAILS (BOTH CIE AND SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**30 (Average of Best Two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
Self learning	**Two Assignments**	**20**	**10**	
	Seminar Presentation	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

SEMESTER END EXAMINATION

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	-	1	-	1	-	1	-	-
CO2	3	3	1	-	2	-	1	-	1	-	-
CO3	3	2	1	-	1	-	1	-	-	2	-
CO4	3	3	1	-	1	-	1	-	1	-	-
CO5	3	2	1	-	1	-	-	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
OPERATING SYSTEMS WITH LINUX PROGRAMMING **Category: IPCC**					
Course Code	:	P24MCA102	CIE	:	50 Marks
Teaching Hours L : T : P	:	3:0:2	SEE	:	50 Marks
Total Hours	:	45 +30	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3Hrs

Course Objectives	
1.	It provides the basics and essentials of operating system.
2.	It provides the concepts of Linux to solve computing problems.
3.	It provides advance shell programming concepts.
4.	It adds new coverage of interactive scripts using regular expressions in simple and advanced filters.

Module - 1	**No. of Hours**
Introduction: operating system, history (1990s to 2000 and beyond), distributed computing, parallel computation. Process concepts: definition of process, process states-Life cycle of a process, process management- process state transitions, process control block(PCB), process operations , suspend and resume, context switching, Interrupts -Interrupt processing, interrupt classes, Inter process communication-signals, message passing.	9
Module - 2	**No. of Hours**
Asynchronous concurrent processes: mutual exclusion- critical section, mutual exclusion primitives, implementing mutual exclusion primitives, Peterson's algorithm, software solutions to the mutual Exclusion Problem-, n-thread mutual exclusion- Lamports Bakery Algorithm. Semaphores – Mutual exclusion with Semaphores, thread synchronization with semaphores, counting semaphores, implementing semaphores. Concurrent programming: monitors, message passing.	9
Module – 3	**No. of Hours**
Deadlock and indefinite postponement: Resource concepts, four necessary conditions for deadlock, deadlock prevention, deadlock avoidance and Dijkstra's Banker's algorithm, deadlock detection, deadlock recovery.	9
Module - 4	**No. of Hours**
Job and processor scheduling: scheduling levels, scheduling objectives, scheduling criteria, preemptive vs non-preemptive scheduling, interval timer or interrupting clock, priorities, scheduling algorithms-FIFO scheduling, RR scheduling, quantum size, SJF scheduling, SRT scheduling, HRN scheduling, multilevel feedback queues, Fair share scheduling.	9
Module - 5	**No. of Hours**
Real Memory organization and Management:: Memory organization, Memory management, Memory hierarchy, Memory management strategies, contiguous v/s non-contiguous memory allocation, single user contiguous memory allocation, fixed partition multiprogramming, variable partition multiprogramming, Memory swapping. Virtual Memory organization: virtual memory basic concepts, multilevel storage organization, block mapping, paging basic concepts, segmentation, Paging/segmentation systems. Virtual Memory Management: Demand Paging, Page replacement strategies.	9

Course Outcomes: At the end of the course, the students will be able to	
CO1	Understand the basics and essentials of operating system.
CO2	Apply the concepts of Linux to solve computing problems.
CO3	Implement advance shell programming concepts.
CO4	Develop interactive scripts using regular expressions in simple and advanced filters.
CO5	Analyze the different memory allocation strategies.

Sl. NO	List of Experiments
1.	Write a shell script that takes a valid directory name as an argument and recursively discard all the subdirectories, find the maximum length of any file in that hierarchy and write its maximum value to the standard output.
2.	Write a shell script that accepts a pathname and creates all the components in that pathname as directories. for example, if the script it named mpc, then the command mpc a/b/c/d should create directories a,a/b,a/b/c,a/b/c/d..
3.	Write a shell script that accepts two file names as arguments check if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file names followed by its permissions.
4.	Write a shell script which accepts valid login names as arguments and print their corresponding home directories. if no arguments are specified, print a suitable error message
5.	Create a script file called file properties that reads a filename entered and out its properties.
6.	Write a shell script to implement terminal locking (similar to the lock command).It should prompt the user for a password. After accepting the password enter by the user, it must be prompt again for the matching password as conformation. And it the match occurs, it must lock the keyword until a matching password is enter again by the user. Note that the script must be written to disregard break, Ctrl-D. No time limit need be implemented for the lock duration.
7.	Write a shell script that accepts one or more file name as arguments and convert all of them to uppercase provided they exists in current directory.
8.	Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If the second argument is not present, the search is to begin in current working directory. In either case the starting directory as well as all its sub- directories at all levels must be searched. The script need not include any error checking.
9.	Write a shell script that accepts as filename as argument and display its creation time if the file exists and if it does not send output error message
10.	Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one or two digit.
11.	Write a shell script to find a file/s that matches a pattern given as command line argument in the home directory, display the contents of the file and copy the file into the directory ~/mydir.
12.	Write a shell script to list all the files in a directory whose file name is atleast 10 characters (Use expr command to check the length).
13.	Write an awk script that accepts date argument in the form of DD- MM-YY and displays it in the form if month, day, year. The script should check the validity of the argument and in the case of error, display a suitable message.
14.	Write an awk script to delete duplicated line from a text file. The order of the original lines must be remain unchanged.

Text Books	
1.	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles – 10th Edition, John Wiley & Sons Inc., 2021.
2.	EviNerneth, "Unix and Linux Handbook", Pearson Education, 5th Edition -2019

Reference Text Books	
1.	Eric Foster –Johnson, John C Welch, Micah Anderson, "Beginning Shell Scripting", Wrox Publication, 2005
2.	Richard Peterson, "The Complete Reference- Linux", Wiley Publication, 6th Edition, 2017 .
3.	Dhananjay M. Dhamdhere, "Operating Systems – A Concept – Based Approach", Tata McGraw – Hill, 3rd Edition, 2017

Web links and Video lectures (e-Resources)
https://onlinecourses.nptel.ac.in/noc21_cs72/preview

ASSESSMENT DETAILS (BOTH CIE AND SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment-1	50	30 (Average of Best Two Assessments)	50/2 = 25
	Internal Assessment-2	50		
	Internal Assessment-3	50		
Self Learning	Two Assignments	10+10	20	
Laboratory	Record & Observation	Evaluating each expt. for 10 marks*12 expts.	10	25
	Lab Internal Test	50	15	
SEE	Semester End Examination	100	50	50
Grand Total				100

SEMESTER END EXAMINATION

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	-	-	-	-	-	-	1	-	-	-	-
CO2	-	-	-	-	-	-	1	-	-	-	1
CO3	3	3	3	3	3	1	1	-	-	-	2
CO4	3	3	3	1	3	1	1	-	-	-	-
CO5	3	3	-	1	3	1	1	-	-	-	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
DATA STRUCTURES AND ALGORITHMS **Category: PCC**					
Course Code	:	P24MCA103	CIE	:	50 Marks
Teaching Hours L : T : P	:	3:0:0	SEE	:	50 Marks
Total Hours	:	45	Total	:	100 Marks
Credits	:	3	SEE Duration	:	3Hrs

Course Objectives	
1.	Analyze step by step and develop algorithms to solve real world problems
2.	Evaluate the Expressions like postfix, prefix conversions.
3.	Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.
4.	Understanding various searching & sorting techniques.
5.	Be able to compare functions using asymptotic analysis and describe the relative merits of worst, average and best-case analysis.

Module - 1	**No. of Hours**
Basic Concepts: Introduction to Arrays, Pointers and Dynamic Memory Allocation, Dynamic Array representations, Structures and Unions. **Classification of Data Structures:** Primitive and Non- Primitive, Linear and Nonlinear; Data structure Operations, **Stack:** Definition, Representation, Operations and Applications: Polish and reverse polish expressions, Infix to post fix conversion, evaluation of postfix expression, infix to prefix, postfix to infix conversion.	9
Module - 2	**No. of Hours**
Queue: Definition, Representation, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues and Programming Examples **Linked List**: Limitations of array implementation, Types of Linked List, Operations of Linked List. **Memory Management:** Static (Stack) and Dynamic (Heap), Memory Allocation, Memory management functions. Definition, Representation, Operations: getnode() and Freenode() operations.	9
Module - 3	**No. of Hours**
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, **Binary Tree Traversals**: Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, **Binary Search Trees:** Definition, Insertion, Deletion, Traversal, Searching, Application of Trees- Evaluation of Expression, Programming Examples.	9
Module - 4	**No. of Hours**
Fundamentals of the Analysis of Algorithm Efficiency: Notion of Algorithm, Fundamentals of Algorithmic problem Solving, Important Problem Types, Fundamental data Structures, Analysis Framework, Asymptotic Notations and Basic efficiency classes, Brute-Force Algorithms - Bubble Sort, Insertion sort, Shell sort.	9
Module - 5	**No. of Hours**
Algorithms: Divide and Conquer – Merge sort, Quick sort, Binary Search, Greedy method – Dijkstra's algorithm, Prim's and Kruskal algorithm, Decrease and Conquer – DFS, BFS.	9

Course Outcomes: At the end of the course, the students will be able to	
CO1	Understand the fundamental Data Structures.
CO2	Apply the operational aspects of stacks, Queues and linked list to solve recursive applications.
CO3	Analyze various types of sorting and searching techniques and identify the optimal approach for a given scenario.
CO4	Summarize the paradigms and approaches used to design and analyze algorithms by categorizing problems based on the popular domains.
CO5	Discuss Divide & conquer algorithms, Greedy Methods and Decrease and Conquer and measure their performance.

Text Books	
1.	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2014.
2.	Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Edition, McGraw Hill, 2014

Reference Text Books	
1.	Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Edition, Cengage Learning, 2014.
2.	Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Edition, McGraw Hill, 2013
3.	Anany Levitin: Introduction to the Design and Analysis of Algorithms, 2nd Edition, 2009. Pearson.

Web links and Video lectures (e-Resources)
• NPTEL https://www.nitt.edu/home/academics/departments/cse/programmes
• Introduction to Data Structures - GeeksforGeeks

ASSESSMENT DETAILS (BOTH CIE AND SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment1	50	30 (Average of Best Two Assessments)	50
	Internal Assessment2	50		
	Internal Assessment3	50		
Self learning	Two Assignments	20	10	
	Seminar Presentation	20	10	
SEE	Semester End Examination	100	50	50
Grand Total				100

SEMESTER END EXAMINATION

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	-	3	2	-	-	-	1	-	1	-	-
CO2	3	-	-	-	-	-	1	-	1	-	-
CO3	3	1	1	-	-	-	1	-	-	2	-
CO4	3	-	-	-	3	-	1	3	1	-	-
CO5	-	2	3	-	1	-	-	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
SOFTWARE ENGINEERING **Category: PCC**					
Course Code	:	P24MCA104	CIE	:	50 Marks
Teaching Hours L : T : P	:	3:0:0	SEE	:	50 Marks
Total Hours	:	45	Total	:	100 Marks
Credits	:	3	SEE Duration	:	3Hrs

Course Objectives	
1.	Outline software engineering principles and activities involved in building large software programs.
2.	Identify ethical and professional issues and explain why they are of concern to software engineers.
3.	Explain the fundamentals of object oriented concepts.
4.	Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation.
5.	Differentiate system models, use UML diagrams and apply design patterns.
6.	Discuss the distinctions between validation testing and defect testing.

Module – 1	**No. of Hours**
Introduction: Professional Software Development Attributes of good software, software engineering diversity, IEEE/ACM code of software engineering ethics, case studies. Software Process and Agile Software Development Software Process models: waterfall, incremental development, reuses oriented, Process activities; coping with change, The Rational Unified Process.	9
Module – 2	**No. of Hours**
Agile Methods, Plan-Driven and Agile Development, Extreme Programming, Agile Project Management, scaling agile methods. Requirement Engineering: Functional and non-functional requirements, The Software requirements document, Requirements specification, Requirements engineering processes, Requirement elicitation and analysis, Requirement validation, Requirement management	9
Module – 3	**No. of Hours**
What is object orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history, modeling as design Technique: Modeling; abstraction; the three models. Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips. Advanced objects and class concepts; Associations ends; N-array association; Aggregation, Abstract class; Multiple inheritance; Metadata; Reification; Constraints; Derived data; packages; practical tips.	9
Module – 4	**No. of Hours**
System Models: Context models, Interaction models. Structural models. Behavioral models. Model-driven Engineering Design and Implementation: Introduction to RUP, Design Principles. Object-oriented design using the UML. Design patterns. Implementation issues. Open source development.	9
Module – 5	**No. of Hours**
Software Testing: Development testing, Test-driven development, Release testing, User testing. Test Automation. Software Evolution: Evolution processes. Program evolution dynamics. Software maintenance. Legacy system management	9

Course Outcomes: At the end of the course, the students will be able to	
CO1	Design a software system, component or process to meet desired needs within realistic constraints.
CO2	Apply programming constructs of C language to solve the real world problem.
CO3	Explore user-defined data structures like arrays in implementing solutions to problems like searching and sorting.
CO4	Explore user-defined data structures like structures, unions and pointers in implementing solutions.
CO5	Design and Develop Solutions to problems using modular programming constructs using functions.

Text Books	
1.	Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012.
2.	Michael Blaha, James Rumbaugh: Object Oriented Modelling and Design with UML, 2nd Edition, Pearson Education, 2005.

Web links and Video lectures (e-Resources)
• https://onlinecourses.nptel.ac.in/noc24_ce107/preview • http://www.digimat.in/nptel/courses/video/106106126/L01.html

ASSESSMENT DETAILS (BOTH CIE AND SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**30 (Average of Best Two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
Self learning	**Two Assignments**	**20**	**10**	
	Seminar Presentation	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

SEMESTER END EXAMINATION

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students shall answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	-	3	-	1	-	1	-	-
CO2	3	3	3	-	2	-	2	-	1	-	-
CO3	3	2	2	-	3	-	1	-	-	2	-
CO4	3	3	3	-	2	-	2	-	1	-	-
CO5	3	2	1	2	3	-	-	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
COMPUTER NETWORKS **Category: PCC**					
Course Code	:	P24MCA105	CIE	:	50 Marks
Teaching Hours L : T : P	:	3:0:0	SEE	:	50 Marks
Total Hours	:	45	Total	:	100 Marks
Credits	:	3	SEE Duration	:	3Hrs

Course Objectives	
1.	Recognize computer networks.
2.	List computer network topologies.
3.	List required hardware to constitute computer network.

Module - 1	**No. of Hours**
Introduction – Network Hardware – Software – Reference Models – OSI and TCP/IP Models – Example Networks: Internet, ATM, Ethernet and Wireless LANs - Physical Layer – Theoretical Basis for Data Communication - Guided Transmission Media.	9
Module - 2	**No. of Hours**
Wireless Transmission - Communication Satellites – Telephone System: Structure, Local Loop, Trunks and Multiplexing and Switching. Data Link Layer: Design Issues – Error Detection and Correction.	9
Module - 3	**No. of Hours**
Elementary Data Link Protocols - Sliding Window Protocols – Data Link Layer in the Internet - Medium Access Layer – Channel Allocation Problem – Multiple Access Protocols – Bluetooth.	9
Module - 4	**No. of Hours**
Network Layer - Design Issues - Routing Algorithms - Congestion Control Algorithms – IP Protocol – IP Addresses – Internet Control Protocols. Introduction to NS2, Wired Script Components and Parameters.	9
Module - 5	**No. of Hours**
Transport Layer - Services - Connection Management - Addressing, Establishing and Releasing a Connection – Simple Transport Protocol – Internet Transport Protocols (ITP) - Network Security: Cryptography.	9

Course Outcomes: At the end of the course, the students will be able to	
CO1	To learn the basic concepts of Data communication and Computer network
CO2	To learn about wireless Transmission.
CO3	To learn about networking and data link layer.
CO4	To study about Network layer.
CO5	To learn the concept of Transport layer.

Text Books	
1.	A. S. Tanenbaum, “Computer Networks”, Prentice-Hall of India,
2.	“Computer Networks A Systems Approach, 6th Edition, 2021“ Larry L Peterson

Reference Text Books	
1.	B. A. Forouzan, “Data Communications and Networking”, Tata McGraw Hill, 4th Edition,2017
2.	F. Halsall, Data Communications, Computer Networks and Open Systems”, Pearson Education,
3.	D. Bertsekas and R. Gallagher, “Data Networks”, PHI,.
4.	Lamarca, “Communication Networks”, Tata McGraw- Hill,

Web links and Video lectures (e-Resources)
https://archive.nptel.ac.in/courses/106/105/106105183/

ASSESSMENT DETAILS (BOTH CIE AND SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**30 (Average of Best Two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
Self learning	**Two Assignments**	**20**	**10**	
	Seminar Presentation	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

SEMESTER END EXAMINATION

1. The question paper shall be set for 100 marks and duration of SEE is 3 hours.
2. Two questions of 20 marks (with minimum of 3 sub questions) from each module with internal choice.
3. Students should answer five full questions, selecting one full question from each module.
4. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	1	-	-	-	-	2	-	-	2	-
CO2	1	1	-	2	-	-	2	-	1	2	-
CO3	3	3	-	2	2	-	2	-	1	2	-
CO4	3	3	-	2	2	-	3	-	1	2	-
CO5	3	3	-	2	-	-	3	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
DATA STRUCTURES AND ALGORITHMS LAB **Category: PCCL**					
Course Code	:	P24MCAL106	CIE	:	50 Marks
Teaching Hours L : T : P	:	0:0:2	SEE	:	50 Marks
Total Hours	:	30	Total	:	100 Marks
Credits	:	1	SEE Duration	:	3Hrs

Course Objectives	
1.	Evaluate the Expressions like postfix, prefix conversions.
2.	Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.

S. No	**Experiments**
1.	Implement a Program in C for converting an Infix Expression to Postfix Expression
2.	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).
3.	Design, develop, and execute a program in C to simulate the working of a queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display
4.	Write a C program to simulate the working of a singly linked list providing the following operations: a. Display & Insert b. Delete from the beginning/end c. Delete a given element
5.	Write a C program to Implement the following searching techniques a. Linear Search b. Binary Search.
6.	Write a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
7.	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm (C programming)
8.	From a given vertex in a weighted connected graph, find shortest paths to other vertices Using Dijkstra's algorithm (C programming)
Demonstration Experiments (For CIE) if any	
9.	Using circular representation for a polynomial, design, develop, and execute a program in C to accept two polynomials, add them, and then print the resulting polynomial.
10.	Design, develop, and execute a program in C to evaluate a valid postfix expression using stack. Assume that the postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).

Course Outcomes: At the end of the course, the students will be able to	
CO1	Implement the techniques for evaluating the given expression.
CO2	Implement sorting / searching techniques, and validate input/output for the given problem.
CO3	Implement data structures (namely Stacks, Queues, Circular Queues, Linked Lists, and Trees), its operations and algorithms.
CO4	Implement the algorithm to find whether the given graph is connected or not and conclude on the performance of the technique implemented.

ASSESSMENT STRUCTURE FOR LABORATORY:

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	**Lab Conduction & Record**	**Evaluating Each Expt. For 10marks*12 expts.**	**15**	**50**
	Laboratory Test 1: After 6 expts	**50**	**15**	
	Laboratory Test 1: After 12 expts	**50**	**20**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	1	-	-	-	-	2	-	-	2	-
CO2	2	1	-	-	-	-	2	-	1	2	-
CO3	2	3	-	-	-	-	2	-	1	2	-
CO4	3	3	-	-	-	-	3	-	1	2	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
COMPUTER NETWORKS LAB **Category: PCCL**					
Course Code	:	P24MCAL107	CIE	:	50 Marks
Teaching Hours L : T : P	:	0:0:2	SEE	:	50 Marks
Total Hours	:	30	Total	:	100 Marks
Credits	:	1	SEE Duration	:	3Hrs

Course Objectives	
1.	Recognize computer networks.
2.	List computer network topologies.
3.	List required hardware to constitute computer network.
4.	Explain each computer network topology physically or logically.

S. No	**List of Experiments**
1.	Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.
2.	Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
3.	Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
4.	Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
5.	Implement Dijsktra's algorithm to compute the shortest path through a network
6.	Implement data encryption and data decryption
7.	Simulate the network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.
8.	Simulate to study transmission of packets over Ethernet LAN and determine the number of packets drop

Course Outcomes: At the end of the course, the students will be able to	
CO1	Implement data link layer farming methods.
CO2	Analyze error detection and error correction codes.
CO3	Implement and analyze routing and congestion issues in network design.
CO4	Implement Encoding and Decoding techniques used in presentation layer.
CO5	To be able to work with different network tools.

ASSESSMENT STRUCTURE FOR LABORATORY:

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing marks for the CIE is 50% of the maximum marks and Minimum passing marks for the SEE is 40% of the maximum marks of SEE. The minimum passing marks is 50% i.e. sum of the CIE and SEE together.

CONTINUOUS INTERNAL EVALUATION (CIE):

Component	**Type of Assessment**	**Max. Marks**	**Max. Marks Scaling Down to**	**Total Marks**
Laboratory	**Lab Conduction & Record**	**Evaluating Each Expt. For 10marks*12 expts.**	**15**	**50**
	Laboratory Test 1: After 6 expts	**50**	**15**	
	Laboratory Test 1: After 12 expts	**50**	**20**	

SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	1	-	-	-	-	2	-	-	2	-
CO2	2	1	-	-	-	-	2	-	1	2	-
CO3	2	3	-	-	-	-	2	-	1	2	-
CO4	3	3	-	-	-	-	3	-	1	2	-
CO5	3	3	-	-	-	-	3	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
C PROGRAMMING **Category: NCMC**					
Course Code	:	P24MCA108A	CIE	:	--
Teaching Hours L : T : P	:	2:0:0	SEE	:	--
Total Hours	:	30	Total	:	--
Credits	:	--	SEE Duration	:	--

Course Objectives	
1.	Elucidate the basic architecture and functionalities of a computer.
2.	Apply programming constructs of C language to solve real-world problems.
3.	Explore user-defined data structures like arrays, structures and pointers in implementing solutions to problems.
4.	Design and Develop Solutions to problems using modular programming constructs such as functions and procedures.

Module - 1	**No. of Hours**
Introduction to C: Introduction to computers, input and output devices, designing efficient programs. Introduction to C, Structure of C program, Files used in a C program, Compilers, Compiling and executing C programs, variables, constants, Input/output statements in C.	6
Module - 2	**No. of Hours**
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.	6
Module - 3	**No. of Hours**
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,	6
Module - 4	**No. of Hours**
Two dimensional arrays, operations on two-dimensional arrays, two-dimensional arrays to functions, multidimensional arrays. Applications of arrays and introduction to strings: Applications of arrays, case study with sorting techniques. Introduction to strings: Reading strings, writing strings, summary of functions used to read and write characters. Suppressing input using a Scanset.	6
Module - 5	**No. of Hours**
Strings: String taxonomy, operations on strings, Miscellaneous string and character functions, arrays of strings. Pointers: Understanding the Computer's Memory, Introduction to Pointers, Declaring Pointer Variables Structures: Introduction to structures, Union , Files	6

Course Outcomes: At the end of the course, the students will be able to	
CO1	Elucidate the basic architecture and functionalities of a computer and also recognize the hardware parts.
CO2	Apply programming constructs of C language to solve the real world problem
CO3	Explore user-defined data structures like arrays in implementing solutions to problems like searching and sorting
CO4	Explore user-defined data structures like structures, unions and pointers in implementing solutions
CO5	Design and Develop Solutions to problems using modular programming constructs using functions

Text Books	
1.	"Reema Thareja", Computer fundamentals and programming in c, Oxford University, 2nd edition, 2017.

Reference Text Books	
1.	E. Balaguruswamy, Programming in ANSI C, 7th Edition, Tata McGraw-Hill.
2.	Brian W. Kernighan and Dennis M. Ritchie, The 'C' Programming Language, Prentice Hall of India.

Web links and Video lectures (e-Resources)
• https://onlinecourses.nptel.ac.in/noc24_cs02/

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3	2	-	3	-	1	-	1	-	-
CO2	3	3	3	-	2	-	2	-	1	-	-
CO3	3	2	2	-	3	-	1	-	-	2	-
CO4	3	3	3	-	2	-	2	-	1	-	-
CO5	3	2	1	2	3	-	-	-	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low

SEMESTER-I					
MATHEMATICS-I **Category: NCMC**					
Course Code	:	P24MCA108B	CIE	:	--
Teaching Hours L : T : P	:	2:0:0	SEE	:	--
Total Hours	:	30	Total	:	--
Credits	:	--	SEE Duration	:	--

Course Objectives	
1.	The Curriculum supports the prerequisites to enhance their Mathematical knowledge towards understanding mathematical Concepts in the concerned fields.
2.	Enhance problem-solving skills using mathematical models and algorithms.
3.	Develop logical reasoning and analytical thinking
4.	Understand the mathematical foundations of computer science, including discrete mathematics and graph theory
5.	Apply mathematical concepts to computer science problems, such as algorithm design and analysis.

Module – 1	**No. of Hours**
Introduction to Number System: Overview of number systems: Binary numbers, Number based conversion, Octal and hexadecimal numbers, Complements.	6
Module – 2	**No. of Hours**
Propositional Logics: Mathematical logic introduction-statements Connectives-negation, conjunction, disjunction- statement formulas and truth tables- conditional and bi Conditional statements- tautology contradiction.	6
Module – 3	**No. of Hours**
Set Theory: Operations on sets, power set, Venn diagram, Cartesian product, relations, functions-types of functions - composition of functions.	6
Module – 4	**No. of Hours**
Matrix algebra: Introduction, Types of matrices-matrix operations, transpose of a matrix, determinant of matrix, inverse of a matrix, Cramer's rule.	6
Module – 5	**No. of Hours**
Differential calculus: Functions and limits - Simple Differentiation of Algebraic Functions – Evaluation of First and Second Order Derivatives – Maxima and Minima.	6

Course Outcomes: At the end of the course, the students will be able to	
CO1	Elucidate the basic architecture and functionalities of a computer and also recognize the hardware parts.
CO2	Apply programming constructs of C language to solve the real world problem
CO3	Explore user-defined data structures like arrays in implementing solutions to problems like searching and sorting
CO4	Explore user-defined data structures like structures, unions and pointers in implementing solutions
CO5	Design and Develop Solutions to problems using modular programming constructs using functions

Text Books	
1.	M. MORRIS MANO Professor of Engineering California State University, Los Angeles, Digital Logic and Computer Design.
2.	Kenneth H. Rosen, Monmouth University, Discrete mathematics and its applications (And formerly AT&T Laboratories).

CO-PO Mapping

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	1	1	2	1	2	1	1	1	-	-
CO2	2	2	2	1	1	3	1	1	1	-	-
CO3	1	1	1	1	2	3	1	2	-	-	-
CO4	1	2	2	2	1	3	-	-	-	-	-
CO5	2	2	2	1	2	1	-	1	1	1	-

Level 3 - High, Level 2 - Moderate, Level 1 - Low