FACULTY OF COMPUTER APPLICATION

PO, PSO, CURRICULUM & SYLLABI

MASTER OF COMPUTER APPLICATION (MCA)

[Academic Session 2023-24 onwards]

PRAYAGRAJ

[Established under the U.P.Private Universities Act. No. 12 of 2019]

FACULTY OF COMPUTER APPLICATION

Master of Computer Application (MCA)

2 Year (4 Semester) Programme

Program Outcomes

PO1: Computational Knowledge: Apply a comprehensive understanding of computing fundamentals, specialized areas of computing, mathematics, and domain-specific knowledge to abstract and conceptualize computing models for solving defined problems and requirements.

PO2: Problem Analysis: Identify, formulate, and solve complex computing problems by conducting extensive research, analyzing relevant literature, and applying fundamental principles of mathematics, computing sciences, and related disciplines to reach substantiated conclusions.

PO3: Design and Development of Solutions: Design, evaluate, and develop innovative solutions for complex computing problems, systems, components, or processes that meet specific needs while considering aspects such as public health and safety, cultural, societal, and environmental factors.

PO4: Conduct Investigations of Complex Computing Problems: Utilize advanced research-based knowledge and research methods, including designing experiments, analyzing and interpreting data, and synthesizing information, to conduct in-depth investigations and provide valid conclusions for complex computing problems.

PO5: Modern Tool Usage: Proficiently utilize a wide range of contemporary computing tools, techniques, and resources to effectively create, select, adapt, and apply them in complex computing activities, while demonstrating an understanding of their capabilities and limitations.

PO6: Professional Ethics: Demonstrate a strong understanding of and commitment to professional ethics, cyber regulations, responsibilities, and the ethical norms and practices of the computing profession.

PO7: Lifelong Learning: Recognize the importance of continuous learning and possess the ability to engage in self-directed and independent learning for ongoing professional development as a computing practitioner.

PO8: Project Management and Finance: Apply knowledge and understanding of computing and management principles to effectively manage projects, both independently and as a member or leader in multidisciplinary teams, while considering project scope, timelines, resources, and financial aspects.

PO9: Communication Proficiency: Demonstrate effective communication skills within the computing community and broader society by comprehending and producing clear, concise, and well-structured reports, design documentation, presentations, and instructions.

PO10: Societal and Environmental Responsibility: Understand and evaluate the impact of computing solutions on society, the environment, and various stakeholders, while considering ethical, legal, societal, and cultural aspects, and adhering to responsible and sustainable computing practices.

PO11: Collaborative Work: Exhibit the ability to work effectively as an individual and as a member or leader in diverse teams and multidisciplinary environments, demonstrating strong interpersonal and teamwork skills, and contributing to the achievement of collective goals.

PO12: Innovation and Entrepreneurship: Identify emerging opportunities, leverage creativity, and apply innovative approaches to solve problems and create value, demonstrating an entrepreneurial mindset that fosters innovation, growth, and socio-economic development.

Program Specific Outcomes

PSO1: Fundamental Understanding of Computer Science: Develop a comprehensive understanding of the fundamentals of computer science, including theoretical concepts, algorithms, data structures, and programming languages. Graduates will be equipped with the knowledge and skills necessary to establish themselves as competent Computer and IT professionals in the dynamic IT and IT-enabled service industry.

PSO2: Technical Excellence in Computer Systems: Demonstrate proficiency in making valuable technical contributions to the design, development, production, and maintenance of computer systems. Graduates will possess strong problem-solving abilities and be capable of applying their technical expertise in real-world scenarios, both in software and hardware domains.

PSO3: Employability and Cross-Disciplinary Collaboration: Prepare graduates for successful careers in a wide spectrum of industrial and business environments. They will be equipped with a solid foundation in computer applications and possess the ability to effectively collaborate with professionals from diverse disciplines. Graduates will understand the cultural and technological nuances required to work seamlessly with people from other domains, facilitating interdisciplinary teamwork and promoting innovation.

FACULTY OF COMPUTER APPLICATION

Master of Computer Application (MCA)

2 Year (4 Semester) Programme

[Academic Session 2023-24 onwards]

CREDITS DISTRIBUTION

S.N.	Category	Number of Courses	Credit
1	Professional Core (PC)	21	55
2	Professional Elective (PE)	4	8
3	Open Elective (OE)	3	9
4	Basic Sciences (BS)	2	5
5	Humanities and Social Sciences (HS)	1	3
6	Project Work, Seminar, Internship (PWSI)	2	20
7	Mandatory Audit Courses (AU)	3	-
Total		**36**	**100**

COURSE STRUCTURE
Master of Computer Application (MCA)
2 Year (4 Semester) Programme
[Academic Session 2023-24 onwards]

Sr. No.	Course Code	Course Title	Category	Teaching			Credit
				L	T	P	
Semester- I							
1	**CAPCMC101T**	Programming & Problem Solving using 'C' Language	PC	3	1	-	4
2	**CAPCMC102T**	Computer Organization & Architecture	PC	3	1	-	4
3	**CAPCMC111T**	Data Structures	PC	3	1	-	4
4	**CAPCMC206T**	Theory of Automata & Formal Language	PC	3	1	-	4
5	**SCSMMCA10T**	Discrete Mathematics	BS	3	1	-	4
6	**CAPCMC101P**	Problem Solving Techniques Using 'C' Lab	PC	-	-	2	1
7	**CAPCMC102P**	Computer Organization & Architecture Lab	PC	-	-	2	1
8	**CAPCMC111P**	Data Structures Lab	PC	-	-	2	1
9	**SCSMMCA10P**	Discrete Mathematics Lab	BS	-	-	2	1
10	**PTSPMCA11T**	Professional Proficiency	AU	3	-	-	0
Total				**18**	**5**	**8**	**24**
Semester- II							
1	**CAPCMC211T**	Object Oriented Programming	PC	3	1	-	4
2	**CAPCMC202T**	Database Management Systems	PC	3	1	-	4
3	**CAPCMC207T**	Design and Analysis of Algorithms	PC	3	1	-	4
4	**CAPCMC205T**	Operating Systems	PC	3	1	-	4
5	**CMSFMCA10T**	Principles of Management & Communication	HS	3	-	-	3
6	**CAPCMC211P**	Object Oriented Programming Lab	PC	-	-	2	1
7	**CAPCMC202P**	Database Management Systems Lab	PC	-	-	2	1
8	**CAPCMC207P**	Design and Analysis of Algorithms Lab	PC	-	-	2	1
9	**CAPCMC208P**	Python Programming Lab	PC	-	-	4	2
10	**PTSPMCA21T**	Professional Proficiency	AU	3	-	-	0
Total				**18**	**4**	**10**	**24**

Note: Mini Project or Internship (3-4 weeks) shall be conducted during summer break after II semester and subsequently assessed in III semester.

Sr. No.	Course Code	Course Title	Category	Teaching			Credit
				L	T	P	
Semester- III							
1	**CAPCMC301T**	Computer Network	PC	3	1	-	4
2	**CAPCMC302T**	Software Engineering	PC	3	1	-	4
3	**CAPCMC303T**	Machine Learning	PC	3	1	-	4
4		***Open Elective – 1***	OE	3	-	-	3
5		***Professional Elective – 1***	PE	3	-	-	3
6	**CAPCMC301P**	Computer Network Lab	PC	-	-	2	1
7	**CAPCMC302P**	Software Engineering Lab	PC	-	-	2	1
8	**CAPCMC303P**	Machine Learning Lab	PC	-	-	2	1
9		***Professional Elective – 1 Lab***	PE	-	-	2	1
10	**CAPCMC314P**	Internship Assessment(M.C.A.)	PWSI	-	-	4	4
11	**PTSPMCA30T**	Professional Proficiency	AU	3	-	-	0
Total				**18**	**3**	**12**	**26**
Semester- IV							
1		***Open Elective – 2***	OE	3	-	-	3
2		***Open Elective – 3***	OE	3	-	-	3
3		***Professional Elective – 2***	PE	3	-	-	3
4		***Professional Elective – 2 Lab***	PE	-	-	2	1
5	**CAPCMC401P**	Major Project (M.C.A.)	PWSI	-	-	16	16
Total				**9**	**0**	**18**	**26**

Note: Students will identify the project problem from their chosen specialization or interdisciplinary in nature as Product Design Technology.

UNITED UNIVERSITY

MCA PROGRAMME

Guidelines on Continuous Assessment (CA)

Continuous Assessment (CA) of a course, with weightage 50%, comprising of two components viz**.**

CLASS TEST (CT) AND TEACHER ASSESSMENT (TA).

(a) Class Test (30 % Marks):

There should be 2 to 3 class tests of at least 1 hour in each paper. First test normally should cover 40 % of the course.

(b) Teacher Assessment (20 % Marks):

It is based on some kind of assignments such as Live Mini Projects with its presentations, and some Presentations based on topics of the paper which may also be related to some real life examples from Corporate & Industrial Houses and some Financial and Economic Institution etc.

*From the academic year 2023-24, 25% of the Teacher Assessment will be based on the attendance of the students in the classes.

Guidelines on Elective Subjects

- In Professional Elective Courses, a student must take at least 2 elective subjects from a basket to specialize in streams, which may be taken in different semesters.
- Subjects in a basket may have prerequisites, which will be explained while opting for any of the elective subjects.
- Allotment of open/professional elective subjects shall be made based on the previously attained CGPA and options to be given.
- Minimum number of students to be allowed in a subject is 40% (Minimum Student Count 10) of total number of students in the respective class/course. If any subject is being opted by less than 40% of the class strength, the subject may not be offered.
- Allocation of open elective subjects may be done based on number of student choices and availability of faculty members who have been identified as experts of the electives subjects.

PROFESSIOANL ELECTIVE COURSES [PE]

Basket/Group: Application Development

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	CSPEAD001T/P	Full Stack Developer using Python	3-0-2	4
2	CSPEAD002T/P	Web Technology	3-0-2	4
3	CSPEAD003T/P	Android App Development	3-0-2	4
4	CSPEAD004T/P	Modern Application Development	3-0-2	4
5	CSPEAD005T/P	Web Development using Design Pattern	3-0-2	4
6	CSPEAD006T/P	Web Development for Blockchain Application	3-0-2	4
7	CSPEAD007T/P	Semantic Web Technology	3-0-2	4
8	CSPEAD008T/P	Application Development using Kotlin	3-0-2	4

Basket/Group: Artificial Intelligence

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	CSPEAI001T/P	Artificial Intelligence	3-0-2	4
2	CSPEAI003T/P	Information Retrieval	3-0-2	4
3	CSPEAI004T/P	Soft Computing	3-0-2	4
4	CSPEAI005T/P	Artificial Neural Network	3-0-2	4
5	CSPEAI006T/P	DeepLearning	3-0-2	4
6	CSPEAI007T/P	Reinforcement Learning	3-0-2	4
7	CSPEAI008T/P	Computer Vision	3-0-2	4
8	CSPEAI009T/P	Natural Language Processing	3-0-2	4
9	CSPEAI010T/P	Cognitive Computing	3-0-2	4
10	CSPEAI011T/P	AI in Healthcare	3-0-2	4
11	CSPEAI012T/P	Machine Learning in Healthcare	3-0-2	4
12	CSPEAI013T/P	Deep Learning in Healthcare	3-0-2	4

Basket/Group: Cyber Security

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	CSPECS001T/P	Information Security	3-0-2	4
2	CSPECS002T/P	Communication Technology	3-0-2	4
3	CSPECS003T/P	Network and Wireless Security	3-0-2	4
4	CSPECS004T/P	Cyber Security with Block chain	3-0-2	4
5	CSPECS005T/P	Data & Cloud Security	3-0-2	4
6	CSPECS006T/P	Security in IoT Devices	3-0-2	4
7	CSPECS007T/P	Web Security	3-0-2	4
8	CSPECS008T/P	Mathematics & Logic behind Modern Cryptography	3-0-2	4

9	CSPECS009T/P	Advanced Cryptography and Network Security	3-0-2	4
10	CSPECS010T/P	Cyber ForecastingToolsandTechniques	3-0-2	4
11	CSPECS011T/P	Biometric Security Analysis	3-0-2	4
12	CSPECS012T/P	Privacy & Security in Big Data	3-0-2	4

Basket/Group: Data Science

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	CSPEDS001T/P	DataScience	3-0-2	4
2	CSPEDS002T/P	Data Mining and Warehousing	3-0-2	4
3	CSPEDS003T/P	Social Network Analysis	3-0-2	4
4	CSPEDS004T/P	Data Visualization and Dashboards	3-0-2	4
5	CSPEDS005T/P	Big Data Analytics	3-0-2	4
6	CSPEDS006T/P	Biological Data Analytics	3-0-2	4
7	CSPEDS007T/P	Structural Equation Modeling	3-0-2	4
8	CSPEDS008T/P	Data Science for BusinessIntelligence	3-0-2	4
9	CSPEDS009T/P	Satellite Data Analysis	3-0-2	4
10	CSPEAI014T/P	Advanced Machine Learning and Data Analytics	3-0-2	4

Basket/Group: Image Processing and Virtual Realty

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	CSPEIP001T/P	Image Processing	3-0-2	4
2	CSPEIP002T/P	Virtual Realty	3-0-2	4
3	CSPEIP003T/P	Computer Graphics and Animation	3-0-2	4
4	CSPEIP004T/P	Graphics and Visual Computing	3-0-2	4
5	CSPEIP005T/P	Augmented Reality	3-0-2	4
6	CSPEIP006T/P	Video Processing & Video Compression	3-0-2	4
7	CSPEIP007T/P	Satellite Image Processing	3-0-2	4
8	CSPEAI008T/P	Computer Vision	3-0-2	4

OPEN ELECTIVE COURSES [OE]

S. No	CourseCode	CourseTitle	L-T-P	Credit
1	**OPENCS007T**	Ecommerce and Social Media Analysis	3-0-0	3
2	**OPENCS011T**	Life Skill and Science of Happiness	3-0-0	3
3	**OPENCS012T**	AI, Public Policy and Law	3-0-0	3
4	**OPENCS014T**	Green ICT	3-0-0	3
5	**OPENCS015T**	Non-Conventional Energy Resources and Applications	3-0-0	3
6	**OPENCS016T**	Material Science	3-0-0	3
7	**OPENCS017T**	Laser Technology	3-0-0	3
8	**OPENCS018T**	Nano Science Technology	3-0-0	3
9	**OPENCS019T**	Quantum Computing	3-0-0	3
10	**OPENCS020T**	Applications of IT in Mass Communication	3-0-0	3
11	**OPENCS021T**	Applications of Image Processing in Agriculture	3-0-0	3
12	**OPENCS022T**	Applications of IoT in Agriculture	3-0-0	3
13	**OPENCS023T**	Empirical Analysis for Dietary and Nutrition	3-0-0	3
14	**OPENCS024T**	Image Processing Technology in Textile Industry	3-0-0	3
15	**OPENCS025T**	Human Values and Professional Ethics	3-0-0	3
16	**OPENCS026T**	Science of Yoga	3-0-0	3
17	**OPENCS031T**	Industrial Sociology	3-0-0	3
18	**OPENCS032T**	Industrial Psychology	3-0-0	3
19	**OPENCS033T**	Entrepreneurship Development	3-0-0	3
20	**OPENCS034T**	Environmental Science	3-0-0	3

<table>
<tr><td><b>Course: MCA</b></td><td colspan="4"><b>Year: 1</b></td></tr>
<tr><td><b>Subject Name:</b> Programming & Problem Solving using ‘C’ Language</td><td colspan="4"><b>Subject Code:</b> CAPCMC101T/P</td></tr>
<tr><td rowspan="2"><b>Semester: I</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>C</b></td></tr>
<tr><td><b>3</b></td><td><b>1</b></td><td><b>2</b></td><td><b>5</b></td></tr>
</table>

Course Objective:
The course is designed to provide complete knowledge of C language. Students will be able to develop logics which will help them to create programs, applications in C. Also by learning the basic programming constructs they can easily switch over to any other language in future.

Unit	Content	Hours
1	**Basics of programming:** Introduction of programming principle, Benefits of learning programming in real world. Types of Programming language. Concept of Algorithm, flowchart, program development steps, overview of the compiler GCC, GDB, Assembler, Linker and Loader. Concept and role of structured programming. C Basics: Introduction of ‘C’ Programming, History, Structures of ‘C’ Program. Overview of compilation and execution process in an IDE. Character set, C Tokens, Keywords, Identifiers, Variables, Constant, Data Types, Comments. Console based I/O and related built in I/O function. Concept of header files, Pre-processor directives.	**9**
2	**Operators:** Types of operators, Precedence and Associativity, Expression, Statement and types of statements. **Control structures:** Decision making structures: If, If-else, Nested If-else, Switch; Loop Control structures: While, Do- while, for, Nested for loop; Other statements: break, continue, goto, exit.	**9**
3	**Functions :** Basic types of function, Declaration and definition, Function call, Types of function, Parameter passing, Call by value, Call by reference, Scope of variable, Storage classes : Introduction, Types- automatic, register, static and external. **Arrays :** Definition, declaration and initialization of one dimensional array, Accessing array elements, Displaying array elements, Sorting arrays, Arrays and function, Two-Dimensional array: Declaration and Initialization, Accessing and Displaying, Memory representation of array; Multi dimensional array. **Pointers:** Introduction, Characteristics, * and & operators, Pointer type declaration and assignment, Pointer arithmetic, Call by reference, Passing pointers tofunctions, arrayof pointers, Pointers to functions, Pointer to pointer, Array ofpointers.	**9**
4	**Dynamic memory allocation:** Introduction, Library functions – malloc, calloc, realloc and free. **Strings :**Strings: Introduction, Initializing strings, Accessing string elements, Array of strings, Passing strings to functions, String functions. **Structures:** Definition and declaration, Variables initialization, Accessing fields and structure operations, Nested structures. Union: Definition and declaration. Usage of unions Operations on union. Enumerated data types .Differentiate between Union and structure.	**9**
5	**Introduction C Preprocessor:** Definition of Preprocessor; Macro substitution directives; File inclusion directives; Conditional compilation. Bitwise Operators: Shift operators; Masks; Bit field. **File Handling;** Definition of Files, File types, File operations, Opening modes of files, File handling functions :fopen(), fclose(), feof(), fseek(), fewind();Using text files: fgetc(), fputc(), fscanf() File handling through command line argument, Record I/O in files.	**9**

List of Experiments:

1. Write programs to perform mathematical operations by taking input from users.
2. Write programs to use the concept of conditional operators.
3. Write programs to use the concept of looping.
4. Write programs to use various types of operators like logical operators.
5. Write programs using array to perform various operations on them.
6. Write programs to implement user defined functions.
7. Write programs to perform various operations on strings.
8. Write programs to using structures and unions to store heterogeneous data together.
9. Write programs for file handling in C Language.
10. Create a small project for “Library Management System’ using file handling, structures and menu based programs.

Learning Resources:

1. Programming in ANSI-C By E. Balaguruswami, TMH Publication
2. Let us C By Yashwant Kanetkar, BPB Publication.
3. HanlyJ. R. and Koffman E. B., “Problem Solving and Program Design in C”, Pearson Education
4. BehrouzA. Forouzan and Richard F. Gilberg, “Computer Science A Structured Programming Approach Using C”, PHI, 3rd Ed.,2007.
5. Jeri R. Hanly and Elliot B. Koffman, “Problem Solving and Programming in C”, Pearson, 5th Ed.2007.
6. B. Kernighan and D. Ritchie, “The ANSI C Programming Language”, PHI.,2000

Course: MCA	**Year: 1**			
Subject Name: Computer Organization & Architecture	**Subject Code:** CAPCMC102T/P			
Semester: I	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:

The course objective of Computer Organization & Architecture is to provide students with a comprehensive understanding of the fundamental principles that govern the design and functioning of computer systems. Students will learn about the organization of computer components, including the CPU, memory, and I/O devices, and how they interact to execute instructions. Additionally, the course aims to cover the architecture of different computer systems, including the concept of instruction set architecture, addressing modes, and memory hierarchies. Through this course, students will develop the necessary knowledge and skills to analyze and optimize the performance of computer systems, enabling them to design efficient and reliable computing solutions.

Unit	Content	Hours
1	**Introduction to Computers:** Basic of Computer, internal organization of CPU, Functional Units of digital system and interconnection, Software, Basic Operational Concepts, Data Representation, Fixed-Point Representation, Floating-Point Representation, bus architecture, types of buses and bus arbitration. Register, Processor organization, general registers organization, stack organization and addressing modes.	**9**
2	**ALU and Micro Operations:** Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and Logic operations. Floating point arithmetic operation, Basic Computer Organization: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt.	**9**
3	**Central Processing Unit:** Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes with numerical examples, Data Transfer and Manipulation, Program Control, Program Interrupt, Types of interrupts, CISC Characteristics, RISC Characteristics. Introduction to Parallel Processing, Pipelining, Polling and Handshaking Controls, Control Design: Hardwired & Micro Programmed (Control Unit), Control Memory, Address Sequencing, Conditional and Unconditional Branching, Micro program Example.	**9**
4	**Memory Organization:** Basic concept of memory and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory. **Input-Output Organization:** Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.	**9**
5	**Parallel Computing;** Multiprocessor and thread level parallelism- classification of parallel architecture-models of communication and memory architecture-Symmetric shared memory architecture-cache coherence protocols-distributed shared memory architecture **Data Parallelism and Microprocessor;** Data Level Parallelism-Vector processors-SIMD extensions, GPU, GPU and CUDA, Overview of CUDA C;	**9**

List of Experiments:

1. Implementing HALF ADDER, FULL ADDER using basic logic gates
2. Implementing Binary -to -Gray, Gray -to -Binary code conversions.
3. Implementing 3-8 line DECODER.
4. Implementing 4x1 and 8x1 MULTIPLEXERS.
5. Verify the excitation tables of various FLIP-FLOPS.
6. Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
7. Design of an 8-bit ARITHMETIC LOGIC UNIT.
8. Design the data path of a computer from its register transfer language description.
9. Design the control unit of a computer using either hardwiring or microprogramming based on its register transfer language description.
10. Implement a simple instruction set computer with a control unit and a data path.

Learning Resources:

1. D. A. Patterson and J. L. Hennessy,*Computer Organisation and Design: The Hardware/Software Interface*, 5/e, Morgan Kaufmann,2014.
2. J. L Hennessy and D. A. Patterson, A.*Computer Architecture: A Quantitative approach*, 6/e, Morgan Kaufmann,2017.
3. V.P.HeuringandH.F.Jordan,*ComputerSystemDesignandArchitecture*,PrenticeHall,2003.
4. D. A. Patterson and J. L. Hennessy,*Computer Organisation and Design: The Hardware/ Software Interface*, 5/e, Harcourt Asia Pte Ltd (Morgan Kaufman),2014.

Course: MCA	**Year: 1**			
Subject Name: Data Structures	**Subject Code:** CAPCMC111T/P			
Semester: I	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:
The objective of this course is to enable students construct and analyze various data structures and abstract data types including lists, stacks, queues trees, and graphs. The course deals with various design constructs and analysis of algorithms and measure performance. Students will implement various sorting, searching, and hashing algorithms. Students will build a substantial, complex data structure.

Unit	Content	Hours
1	**Introduction:** Basic Terminology, Elementary Data Organization, Abstract Data Types (ADT), Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations.	**9**
2	**Linked lists:** Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List.**Stacks:** Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack in C, Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi Problem, Simulating Recursion, Principles of recursion, Tail recursion, Removal of recursion Queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.	**9**
3	**Trees:** Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: Inorder, Preorder and Postorder, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm. .	**9**
4	**Graphs:** Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, **Graph Traversal:** Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transistive Closure and ShortestPath algorithm: Warshal Algorithm and Dijikstra Algorithm.	**9**
5	**Searching:** Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting. Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees. Hashing: Hash Function, Collision Resolution Strategies. Storage Management: Garbage Collection and Compaction.	**9**

List of Experiments:

1. Write C Programs to illustrate the concept of the following:
2. Sorting Algorithms-Non-Recursive.
3. Sorting Algorithms-Recursive.
4. Searching Algorithm.
5. Implementation of Stack using Array.
6. Implementation of Queue using Array.
7. Implementation of Circular Queue using Array.
8. Implementation of Stack using Linked List.
9. Implementation of Queue using Linked List.
10. Implementation of Circular Queue using Linked List.
11. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
12. Graph Implementation, BFS, DFS, Minimum cost spanning tree, shortest path algorithm.

Learning Resources:

1. Aaron M. Tenenbaum, Yedidyah Langsam and Moshe J. Augenstein, “Data Structures Using C and C++”, PHI Learning Private Limited, Delhi India
2. Lipschutz, “Data Structures” Schaum’s Outline Series, Tata McGraw-hill Education (India) Pvt. Ltd.
3. Thareja, “Data Structure Using C” Oxford Higher Education.

Course: MCA	**Year: 1**			
Subject Name: Theory of Automata & Formal Language	**Subject Code:** CAPCMC206T			
Semester: I	**L**	**T**	**P**	**C**
	3	**1**	**0**	**4**

COURSE OBJECTIVES:-

This course is extensive and theoretical treatment of issues in Computability and Complexity; Topics include Automata and Language Theory, Computability Theory, and Complexity Theory.

Unit	Content	Hours
1	**Basic Concepts and Automata Theory:** Introduction to Theory of Computation- Automata, Computability and Complexity, Alphabet, Symbol, String, Formal Languages, Chomsky Classification, Deterministic Finite Automaton (DFA)- Definition, Representation, Acceptability of a String and Language, Non Deterministic Finite Automaton (NFA), Equivalence of DFA and NFA, NFA with ε-Transition, Equivalence of NFA's with and without ε-Transition, Finite Automata with output- Moore machine, Mealy Machine, Equivalence of Moore and Mealy Machine, Minimization of Finite Automata, Myhill - Nerode Theorem, Simulation of DFA and NFA.	**9**
2	**Regular Expression and Language:** Regular Expression, Transition graph, statement of Kleen's Theorem, Regular Expressions, Equivalence of DFAs, NFAs and Regular Expressions Arden's theorem, Algebraic Method Using Arden's Theorem,, Closure properties of Regular Language, Non-Regular Languages, Pumping Lemma, Application of Pumping Lemma . Myhill Nerode Theorem.	**9**
3	**Push down Automata and Context Free Languages**: Context Free Grammar (CFG), designing context free grammar , ambiguity in CFG and its removal Parse Trees, Conversion of FA into CFG and Regular grammar into FA, Simplification of CFG, Normal Forms- Chomsky Normal Form(CNF), Greibach Normal Form (GNF),Chomsky Hierarchy, Push Down Automata(deterministic and nondeterministic) (PDA), graphical notations, Language accepted by PDA ,Equivalence of CFGs and PDAs, Closure properties of CFLs, , Decision Problems of CFL Pumping Lemma for CFL Programming problems based on the properties of CFLs., Parsing(including LL(1) , SLR and LR(1) Parsing Method)	**9**
4	**Turing Machines and Computability Theory:** Definition of Turing Machine, Extensions of Turing machines, Non – deterministic Turing machines, Equivalence of various Turing Machine Formalisms, Church – Turing Thesis, Decidability, Halting Problem, Reducibility, Recursion Theorem.	**9**
5	**Complexity Theory:** Time and Space measures, Hierarchy theorems, Complexity classes P, NP, space complexity, Savich theorem, L, NL, PSPACE complexity, Post correspondence problem, Probabilistic computation.	**9**

Learning Resources:

1. J. C. Martin, "Introduction to Languages and the Theory of Computation", TMH
2. J. Hopcroft, R. Motwani, and J. Ullman, "Introduction to Automata Theory, Language and Computation", Pearson.
3. J. Martin, "Introduction to languages and the theory of computation", McGraw Hill
4. Y.N.Singh," Mathematical Foundation of Computer Science", New Age International.
5. M. Sipser, "Introduction to the Theory of Computation", Cengage Publication.
6. H. R. Lewis and C. H. Papadimi Triou, "Elements of the Theory of Computation", Pearson.
7. K. L. Mishra and N. Chandrasekharan, "Theory of Computer Science Automata Language Computation", PHI.
8. Peter Linz, "Introduction to Formal Languages and Automata", Narosa.
9. Sudkamp, "Languages and Machines", Pearson Education.
10. Bernard Moret, "Theory of Computation", Pearson Education.

Course: MCA	**Year: 1**			
Subject Name: Discrete Mathematics	**Subject Code:** SCSMMCA10T/P			
Semester: I	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

COURSE OBJECTIVES:

The course objective of Discrete Mathematics is to introduce students to mathematical structures and concepts that are fundamental to computer science and related fields. It covers topics such as logic, sets, relations, functions, combinatorics, and graph theory. The course aims to enhance students' problem-solving abilities and logical reasoning skills, enabling them to apply discrete mathematics principles to solve real-world problems in various computing and engineering disciplines.

Unit	Content	Hours
1	**Sets:** Definition of sets, Types of Sets, Operations on Sets, Cartesian Product of Sets. **Relation:** Definition, types of relation, composition of relations, equivalence relation, partial ordering relation, Closure of Relations. **Functions:** Definition, type of functions, composition of functions, recursively defined functions.	**9**
2	**Algebraic Structures:** Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, examples and standard results.	**9**
3	**Hasse Diagram and Lattices:** Introduction, ordered set, Posets, Hasse diagram of partially ordered set, isomorphic ordered set, well ordered set, properties of Lattices, and complemented lattices.	**9**
4	**Mathematical Logic:** Proposition, First order logic, Basic logical operations , truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, Theory of Inference, predicates, Universal and existential quantifiers	**9**
5	**Graphs:** Simple graph, multi graph, graph terminology, representation of graphs, Adjacency and Incidence Matrices, Spanning, Euler graphs, Hamiltonian path and circuits, Shortest Path, Bipartite, Regular, Planar and connected graphs, connected components in a graph, Graph coloring , chromatic number, isomorphism and Homomorphism of graphs. **Combinatorics:** Basic Counting Technique, Pigeon-hole Principle, Recurrence Relation, Generating function, Polya's Counting Theorem	**9**

List of Experiments:

Programming Language/Tool Used: C and Maple

1. Write a program in C to create two sets and perform the Union operation on sets.
2. Write a program in C to create two sets and perform the Intersectison operation on sets.
3. Write a program in C to create two sets and perform the Difference operation on sets.
4. Write a program in C to create two sets and perform the Symmetric Difference operation.
5. Write a program in C to perform the Power Set operation on a set.
6. Write a program in C to Display the Boolean Truth Table for AND, OR , NOT .
7. Write a C Program to find Cartesian Product of two sets
8. Write a program in C for minimum cost spanning tree.
9. Write a program in C for finding shortest path in a GraphNote: Understanding of mathematical computation software Mapple to experiment the followings (exp. 10 to 25):
10. Working of Computation software
11. Discover a closed formula for a given recursive sequence vice-versa
12. Recursion and Induction
13. Practice of various set operations
14. Counting
15. Combinatorial equivalence
16. Permutations and combinations
17. Difference between structures, permutations and sets
18. Implementation of a recursive counting technique

Learning Resources:

1. Discrete Mathematics and Its Applications, By Kenneth H Rosen, McGraw Hill
2. B. Kolman, R.C Busby and S.C Ross, "Discrete Mathematics Structures", Prentice Hall
3. Discrete Mathematical Structures with Applications to Computer Science, By J. P. Tremblay, R.Manohar, McGraw Hill.
4. Graph Theory With Applications to Engineering and Computer Science, By Prentice Hall, Englewood Cliffs, N. J
5. Combinatorics: Theory and Applications, By V. Krishnamurthy, East-West Press Pvt. Ltd., New Delhi

Course: MCA	**Year: 1**			
Subject Name: Professional Proficiency	**Subject Code:** PTSPMCA11T			
Semester: I	**L**	**T**	**P**	**C**
	3	**0**	**0**	**0**

Course Objectives:

Students should be able to read and write correct English, attain reasonable fluency in the Language and should also be exposed to introductory lessons of Aptitude Building

Unit	Content	Hours
1	**HARD SKILLS** Hard skill includes Basic Grammar, Vocabulary, Articles, Tenses, Construction of Sentences and Reading Comprehension etc.	**10**
2	**COMMUNICATION SKILL** Efforts should be made to overcome the initial hesitation of speaking English of students and hence improve their fluency in English. Suggested methods include: • Follow only English language in the class. • Class should be interactive and students should always be engaged in some kind of conversation. • Each student should speak 10 minutes, 2-5 times in 2nd semester on topics of his choice selected from Social, Environmental, Sports, Business and Economics, Medicines and Health Care, Science and Technology ,Politics, World Affairs and Religion etc. • In the above process students should be regulated towards better Vocabulary and Pronunciation.	**25**
3	**APPTITUDE BUILDING** **QUANTITATVIE APPTITUDE** 1. Ratio and proportion. 2. Partnership. 3. Problem on Ages. **LOGICAL REASONING** 1. Inequalities. 2. Direction Test. 3. Syllogism (Basics).	**10**

Course: MCA	**Year: 1**			
Subject Name: Object Oriented Programming	**Subject Code:** CAPCMC211T/P			
Semester: II	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objective:
The course objective of Object-Oriented Programming (OOP) is to teach students the principles and concepts of object-oriented programming. It focuses on understanding classes, objects, inheritance, polymorphism, and encapsulation. The course aims to develop students' skills in designing, implementing, and maintaining object-oriented software, fostering the ability to create efficient, modular, and reusable code in various programming languages.

Unit	Content	Hours
1	OOPS concept & Java Language Basics: Object oriented approach. Basic terms and ideas Abstraction, Encapsulation. Inheritance& Polymorphism. Structured vs. Object Oriented Programming. Benefits of Object oriented programming. Introduction To Java, basic features, Java Virtual Machine Concepts, java environment. Primitive data types, tokens, variables constants & Java keywords. Java Operators. Java program structure. A simple Java program. Expressions, Statements, Control Statements, Selection Statements, Iterative Statements, Jump Statements. Arrays, Vector. String handling & wrapper classes.	**9**
2	OOP, Package, Exception Handling & Multithreading: Classes and Methods- Implementing & designing classes, constructors, polymorphism & inheritance. Interfaces. Interface: defining Interface, Extending Interface, implementing Interface & Accessing Interface Variable Package: System packages, using system package, import. Adding a class to a package, Hiding classes. Exception Handling: Concepts of Exceptions, types of exceptions, try; catch & finally keywords, throwing exceptions & nested try and catch. Multithreaded Programming: Life cycle of a Thread, creating Threads, extending Threads class, Stopping& blocking a thread, , using thread methods, thread exceptions, thread priority, and synchronization.	**9**
3	Java Applets & GUI: Java applets-Life cycle of an applet, adding images & sound to an applet. Passing parameters to an applet. Graphics & GUI: Working with Windows Graphics and Text. Using AWT Controls, Layout Managers, Event Handling & Menus. Swing based GUI	**9**
4	JDBC & Networking: JDBC- Overview, JDBC implementation, Connection class & Statements. Catching Database Results, handling database Queries. Networking– InetAddress class, URL class. TCP/IP & UDP sockets. RMI	**9**
5	Advance Java: Web programing- Web page Designing using HTML, Introduction to Java script features. Java Servlets- life cycle of a servlet. The Servlet API, Get and Post Methods, using Cookies & Session Tracking. JSP-JSP life cycle & JSP tags. Java Beans- types of beans, Stateless & stateful beans,	**9**

List of Experiments:

1. To write programs to illustrate the uses of decision control structures: if, nested if, switch case etc.
2. To write programs to illustrate the uses of loop control structures: do, while, for etc.
3. To write programs to illustrate the uses of array, Vector & String.
4. To write programs to illustrate the uses of creating and working with class and object.
5. To write programs to illustrate the uses of OOPs concepts: data abstraction, data hiding, encapsulation, inheritance & polymorphism (method overloading and overriding).
6. To write programs to illustrate the uses of Interfaces and packages.
7. To write programs using Multithreading & exceptions handling mechanism.
8. To write Java applets using Graphics.
9. To write GUI programs using AWT controls.
10. To write GUI programs to implement various layouts
11. To write GUI programs to handle mouse & key events.
12. To write network programs using TCP/IP & UDP sockets.

13. To write programs to retrieve data from data base using JDBCtype-1 and type-4 drivers.
14. To write servlet program using Generic and HTTP servlets.
15. To write servlet program that handles the user request by using doGet () and doPost () methods.
16. To write servlet program to implement Session Tracking.
17. To write programs to create a web page using JSP.
18. To write programs using RMI &Java Beans.

Learning Resources:

1. E. Balagunisamy. “Programming in Java”, TMH Publications.
2. Java The Complete Reference, Herbert Schildt 7th Edition. Tata McGraw- Hill Edition.
3. S. Horstmann, Gary Cornell – “Core Java 2 Volume II – Advanced Features” Addison Wesley.
4. Date C J, “An Introduction To Database System”, Addision Wesley
5. Korth, Silbertz, Sudarshan, “Database Concepts”, Tata Mcgraw-hill Education (India) Pvt. Ltd.
6. Elmasri, Navathe, “Fundamentals Of Database Systems”, Pearson Education New Delhi India.
7. Bipin C. Desai, “An introduction to Database Systems”, Galgotia Publication Pvt. Ltd. New Delhi.

<table>
<tr><td><b>Course: MCA</b></td><td colspan="4"><b>Year: 1</b></td></tr>
<tr><td><b>Subject Name:</b> Database Management Systems</td><td colspan="4"><b>Subject Code:</b> CAPCMC202T/P</td></tr>
<tr><td rowspan="2"><b>Semester: II</b></td><td><b>L</b></td><td><b>T</b></td><td><b>P</b></td><td><b>C</b></td></tr>
<tr><td><b>3</b></td><td><b>1</b></td><td><b>2</b></td><td><b>5</b></td></tr>
</table>

Course Objective:
Understand the basic concepts and the applications of database systems. The course emphasizes the understanding of the fundamentals of relational systems including data models, database architectures, and database manipulations.• Familiar with database storage structures and access technique understand the relational database design principles. Familiar with the basic issues of transaction processing and concurrency control and recovery

Unit	Content	Hours
1	**Introduction**: An overview of database management system, Database System Vs File System, Database system concepts and architecture, data models schema and instances, data independence and data base language and interfaces, Data definitions language, DML, Overall Database Structure. **Data Modelling using the Entity Relationship Model**: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.	**9**
2	**Relational data Model and Language**: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, tuple and domain calculus. **Introduction to SQL**: Characteristics of SQL, Advantages of SQL, SQL data types and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes, Queries and sub queries, Aggregate functions, Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors in SQL.PL/SQL, Triggers and clusters.	**9**
3	**Data Base Design & Normalization**: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependencies, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.	**9**
4	**Transaction Processing Concepts**: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling	**9**
5	**Concurrency Control Techniques:** Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi-version schemes, Recovery with concurrent transaction. Transaction Processing in Distributed system, data fragmentation. Replication and allocation techniques for distributed system, overview of concurrency control and recovery in distrusted database.	**9**

List of Experiments:

1. Student should decide on a case study and formulate the problem statement.
2. Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)
   Note: Student is required to submit a document by drawing ER Diagram to the Lab teacher.
3. Converting ER Model to Relational Model (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys)
   Note: Student is required to submit a document showing the database tables created from ER Model.
4. **Normalization -**To remove the redundancies and anomalies in the above relational tables, Normalizeup to Third Normal Form
5. **Creation of Tables using SQL-** Overview of using SQL tool, Data types in SQL, Creating Tables(along with Primary and Foreign keys), Altering Tables and Dropping Tables
6. **Practicing DML commands-** Insert, Select, Update, Delete
7. Practicing Queries using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSECT,
8. CONSTRAINTS etc.

9. Practicing Sub queries (Nested, Correlated) and Joins (Inner, Outer and Equi).
10. Practice Queries using COUNT, SUM, AVG, MAX, MIN, GROUP BY, HAVING, VIEWS Creation and Dropping.
11. **Practicing on Triggers -** creation of trigger, Insertion using trigger, Deletion using trigger, Updating using trigger
12. **Procedures-** Creation of Stored Procedures, Execution of Procedure, and Modification of Procedure.

Learning Resources:

1. Date C J, "An Introduction To Database System", Addision Wesley
2. Korth, Silbertz, Sudarshan, "Database Concepts", Tata Mcgraw-hill Education (India) Pvt. Ltd.
3. Elmasri, Navathe, "Fundamentals Of Database Systems", Pearson Education New Delhi India.
4. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication Pvt. Ltd. New Delhi.
5. Majumdar& Bhattacharya, "Database Management System", Tata Mcgraw-hill Education (India) Pvt. Ltd.
6. G.K. Gupta, "Database Management System", Tata Mcgraw-hill Education (India) Pvt. Ltd.
7. Ramakrishnan, Gehrke, "Database Management System", McGraw Hill (India) Pvt Ltd. New Delhi.
8. Chakravarti, "Advanced Database Mnagement System" Wiley Dreamtech Publications.

Course: MCA	**Year: 1**			
Subject Name: Design and Analysis of Algorithms	**Subject Code:** CAPCMC207T/P			
Semester: II	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:
Course objectives for Design and Analysis of Algorithms: Analyze and develop algorithms using mathematical and algorithmic techniques to solve computational problems. Understand the complexity of algorithms and evaluate their efficiency. Apply divide-and-conquer, dynamic programming, and greedy paradigms to design efficient algorithms. Develop problem-solving and analytical skills in algorithmic design and performance analysis.

Unit	Content	Hours
1	**Introduction:** Algorithms, Analysing algorithms, Complexity of algorithms, Growth of functions, Performance measurements, Analysis of sorting and order Statistics algorithms - Shell sort, Quick sort, Merge sort, Heap sort, Comparison of sorting algorithms, Sorting in linear time.	**9**
2	**Advanced Data Structures:** AVL Tree, Red-Black Tree, Trie, B – trees, Comparison of Tree structures, Binomial Heaps, Fibonacci Heaps, Augmenting Data Structures- Optimal Binary Search Trees, Amortized Algorithm and Analysis	**9**
3	**Divide and Conquer** with examples such as Sorting, Matrix Multiplication, Convex hull and Searching. **Greedy methods** with examples such as Optimal Reliability Allocation, Knapsack, Minimum Spanning trees – Prim's and Kruskal's algorithms, Single source shortest paths **-** Dijkstra's and Bellman Ford algorithms.	**9**
4	**Dynamic programming** with examples such as Knapsack, All pair shortest paths – Warshal's and Floyd's algorithms, Resource allocation problem. **Backtracking, Branch and Bound** with examples such as Travelling Salesman Problem, Graph Coloring, n-Queen Problem, Hamiltonian Cycles and Sum of subsets.	**9**
5	**Advanced Topics:** String Matching: Naïve, KMP, RK etc., Theory of NP-completeness, Approximation algorithms and Randomized algorithms	**9**

List of Experiments:

1. Implementing sorting algorithms.
2. Implementing at least 1 Example/Application of Greedy Algorithm.
3. Implementing at least 1 Example/Application of Divide and Conquer Algorithm.
4. Implementing at least 1 Example/Application of DP Algorithm
5. Implementing at least 1 Example/Application of Backtracking Algorithm.
6. Implementing at least 1 Example/Application of Branch & Bound Algorithm.
7. Implementing at least 1 String Matching Algorithm
8. Implementing AVL tree with operations: Insertion, deletion, search, traversal.
9. Implementing RB tree with operations: Insertion, deletion, search, traversal.
10. Implementing TRIE with operations: Insertion, deletion.
11. Implementing B-Tree tree with operations: Insertion, deletion.

Learning Resources:

1. Thomas H. Coreman, Charles E. Leiserson and Ronald L. Rivest, "Introduction to Algorithms", Printice Hall of India.
2. Thomas H. Coreman, "Algorithms Unlocked", MIT Press, 2013
3. E. Horowitz & S Sahni, "Fundamentals of Computer Algorithms"
4. Aho, Hopcraft, Ullman, "The Design and Analysis of Computer Algorithms" Pearson Education, 2008.
5. Jon Kleinberg and Éva Tardos, Algorithm Design, Pearson, 2005.
6. Michael T Goodrich and Roberto Tamassia, Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition, Wiley, 2006.
7. Harry R. Lewis and Larry Denenberg, Data Structures and Their Algorithms, Harper Collins, 1997
8. Robert Sedgewick and Kevin Wayne, Algorithms, fourth edition, Addison Wesley, 2011.
9. Harsh Bhasin,"Algorithm Design and Analysis", First Edition, Oxford University Press.
10. Gilles Brassard and Paul Bratley, Algorithmics: Theory and Practice, Prentice Hall,1995.

Course: MCA	**Year: 1**			
Subject Name: Operating Systems	**Subject Code:** CAPCMC205T			
Semester: II	**L**	**T**	**P**	**C**
	3	**1**	**0**	**4**

Course Objective:
Course objectives of Operating Systems: Understand the fundamental concepts and components of operating systems, including process management, memory management, file systems, and device management. Analyze and evaluate different scheduling algorithms for efficient resource allocation. Learn about synchronization mechanisms and inter-process communication. Develop skills to design and implement basic operating system functionalities.

Unit	Content	Hours
1	**Introduction:** Definition and types of operating system, Evolution, Batch Processing System, Multiprogramming, Time Sharing, Parallel System, Real Time System, Distributed System, Network System, Operating System Structure, Components of Operating System, services, functions, System Calls, System programs, Kernel and its types, Virtual Machines	**9**
2	**Process Management:** Concept of Process, process queues, process Scheduling, Cooperating Process, Threads, Inter-Process Communication, CPU scheduling Criteria, Scheduling algorithms, Multiple Processor Scheduling, Real Time Scheduling, Algorithm Evaluation.	**9**
3	**Process Synchronization and Deadlock:** The Critical Section Problem, Synchronization Hardware, Semaphores, Monitors, Classical Problems of Synchronization, Critical Region, Deadlock System Model, Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock, Combined approach to handle Deadlock, Banker's Algorithm	**9**
4	**Memory Management:** Logical and Physical Address Space, Swapping, Contiguous Allocation, Dynamic Memory Allocation, Fragmentation, Memory Freeing, Virtual Memory Concepts and Its Implementation, Demand Paging and its Performance, Page Replacement Algorithms, Allocation of Frames, Thrashing, Page size and other consideration, Demand Segmentation.	**9**
5	**File Management and Security:** File System, Secondary Storage structure, Concept of File, Access Methods, Directory Implementation, Efficiency and Performance, Recovery**Security:** Safeguards, Penetration, Access and Information Flow control, Protection Problems, Formal Model of Protection	**9**

Learning Resources:

1. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Principles , 9/e, John Wiley,2013.
2. S. Tanenbaum, Modern Operating Systems, 4/e, Pearson Education, 2017.
3. G. J. Nutt, Operating Systems - A Modern Perspective, 3/e, Pearson Education, 2009.
4. W. Stallings, Operating Systems:Internals and design Principles , 7/e, Pearson Education, 2012.

Course: MCA	**Year: 1**			
Subject Name: Principles of Management & Communication	**Subject Code:** CMSFMCA10T			
Semester: II	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

COURSE OBJECTIVES:-

The objectives of the Management course are to provide students with the knowledge of the historical evolution of management and the understanding of management functions and principles. The course aims to equip students with the ability to apply these principles effectively in real-world organizational settings. Additionally, it focuses on developing strong communication skills to enable effective communication within the organization and overcome communication barriers**.**

Unit	Content	Hours
1	**Nature of Management:**Meaning, Definition, nature & purpose, importance & Functions, Management as Art, Science & Profession, Management as social System, Concepts of management-Administration-Organization, Management Skills, Levels of Management.	**7**
2	**Evolution of Management Thought:**Contribution of F.W.Taylor, Henri Fayol, Elton Mayo, Chester Barhard & Peter Drucker to the management thought.Business Ethics &Social Responsibility: Concept , Shift to Ethics, Tools of Ethics.	**7**
3	**Planning and Organizing:**Planning – Meaning- Need & Importance, types, Process of Planning, Barriers to Effective Planning, levels –advantages & limitations, Forecasting- Need & TechniquesDecision making-Types – Process of rational decision making & techniques of decision makingOrganizing–Elements of organizing & processes: Types of organizations, Delegation of authority – Need, difficulties , Delegation – Decentralization ,Staffing- Meaning & Importance	**7**
4	**Directing and Controlling:** Direction – Nature – Principles ,Motivation – Importance – theories, Leadership – Meaning –styles, qualities & function of leader Controlling – Need, Nature, importance, Process & Techniques, Total Quality Management. Coordination-Need-Importance	**7**
5	**Communication :**Communication - Nature ,Process, Networks and Barriers, Effective Communication	**7**

Learning Resources:

1. Essential of Management – Horold Koontz and Iteinz Weibrich- McGraw-Hill’s International
2. Management Theory & Practice – J.N.Chandan
3. Essential of Business Administration – K.Aswathapa, Himalaya Publishing House
4. Principles of Management, By Tripathi, Reddy Tata McGraw Hill
5. Principles & practice of management – Dr. L.M.Prasad,, Sultan Chand & Sons – New Delhi
6. Business Organization & Management – Dr. Y.K.Bhushan

Course: MCA	**Year: 1**			
Subject Name: Professional Proficiency	**Subject Code:** PTSPMCA21T			
Semester: II	**L**	**T**	**P**	**C**
	3	**0**	**0**	**0**

Course Objectives:

- To put in use the basic mechanics of Grammar.
- To provide an outline to effective Organizational Communication.
- Understand the role of communication in personal & professional success.
- Prepare and present messages with a specific intent.

Unit	Content	Hours
1	**Components of Technical Writing and Functional Grammar** Words and Phrases: Word formation; Root words from foreign languages & their use in English; Prefixes & Suffixes: Derivatives; Synonyms; Antonyms; Correct Usage: sub-verb agreement; Parts of Speech ; Modals; Concord; Articles; Infinitives; vocabulary development: technical vocabulary, vocabulary used in formal letters/emails and reports.	**7**
2	**Fundamentals of Technical Communication** Introduction to Communication; Process of Communication; Technical Communication: features: Distinction between General And Technical Communication; The flow of communication: Downward, Upward, Lateral/Horizontal (Peer group) ; Barriers to Communication ; Dimensions of Communication: Reading, Listening & Comprehension: skills, types & methods.	**7**
3	**Technical Style & Written Communication:** Technical Style: Features; types ;Requisites of Sentence Construction; Types of Sentences; Paragraph Development: Techniques and Methods: Inductive, Deductive, Spatial , Linear, Chronological etc. Devices; Coherence; Unity ; Emphasis in Writing; Use of Writing methods in Documents; Techniques of writing.	**7**
4	**Written Business Communication** Letter writing : Principles, Type : Sales ; Credit letters; Claim; Adjustment Letters; Job Application & official letter; Reports: Types; Significance; Structure, & drafting of Reports. Technical Proposal; Types; Writing of Proposal; Significance; Seminar & Conference paper writing; Expert Technical Lecture: Theme clarity; Analysis & Finding; Notices; Agenda; Minutes of Meeting.	**7**
5	**Presentation Strategies & Oral Communication** Analysis of Audience and Locale; Nuances and Modes of Delivery; Kinesics ; Proxemics; Dimensions of Speech: Syllable; Accent; Pitch; Rhythm; Intonation; Paralinguistic features of voice ; Methods of Presentation: Interpersonal; Impersonal; Audience Participation: Quizzes & Interjections; Flow in Speaking; Public Speaking: method; Techniques: Clarity of substance; emotion; Humour ;	**7**

Learning Resources:

1. Improve your Writing ed. V.N. Arora and Laxmi Chandra, Oxford Univ. Press, 2001, New Delhi.
2. Technical Communication- Principles and Practices by Meenakshi Raman & Sangeeta Sharma, Oxford Univ. Press, 2007, New Delhi.
3. Functional skills in Language and Literature, by R.P. Singh, Oxford Univ. Press, 2005, New Delhi.
4. Ashraf Rizvi, "Effective Technical Communication", 2ndEdition, McGraw Hill Education, 2017.
5. Communication Skills for Engineers and Scientists, Sangeeta Sharma et.al. PHI Learning Pvt. Ltd, 2011, New Delhi.
6. Business Correspondence and Report Writing by Prof. R.C.,Sharma& Krishna Mohan, Tata McGraw Hill & Co. Ltd. , 2001, New Delhi.
7. Word Power Made Easy by Norman Lewis, W.R. Goyal Pub. & Distributors, 2009, Delhi.
8. Developing Communication Skills by Krishna Mohan, MecraBannerji- Macmillan India Ltd. 1990, Delhi

Course: MCA	**Year: 2**			
Subject Name: Computer Networks	**Subject Code:** CAPCMC301T/P			
Semester: III	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:

The course objectives of Computer Networks are to provide students with a comprehensive understanding of communication interfaces between users and computer hardware, the structure, functions, services, and components of operating systems. It also aims to familiarize students with process representation, handling, scheduling, synchronization, and the concepts of memory management, file, and security issues in computer networks.

Unit	Content	Hours
1	**Introduction Concepts:** Goals and Applications of Networks, Network structure and architecture, The OSI reference model, services, Network Topology Design, Physical Layer Transmission Media, Line coding scheme, switching methods (circuit switching, Packet switching), TDM.	**9**
2	**Data link layer:** Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs.	**9**
3	**Network Layer:** Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.	**9**
4	**Transport Layer:** Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP.	**9**
5	**Application Layer:** Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. APPLICATION LAYER PROTOCOLS: Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol, Telnet.	**9**

List of Experiments:

1. Study of different types of Network cables and practically implement the cross-wired cable and straight through cable using clamping tool.
2. Study of Network Devices in Detail.
3. Study of network IPv4 and IPv6.
4. Connect the computers in Local Area Network.
5. Study of basic network command and Network configuration commands.
6. Configure a Network topology using packet tracer software.
7. Configure a Network topology using packet tracer software.
8. Configure a Network using Distance Vector Routing protocol.
9. Configure Network using Link State Vector Routing protocol.

Learning Resources:

1. A. S. Tanenbaum (2003), Computer Networks, 4th edition, Pearson Education/ PHI, New Delhi, India.
2. Behrouz A. Forouzan (2006), Data communication and Networking, 4th Edition, Mc Graw-Hill, India.
3. Kurose, Ross (2010), Computer Networking: A top down approach, Pearson Education, India.

Course: MCA	**Year: 2**			
Subject Name: Software Engineering	**Subject Code:** CAPCMC302T/P			
Semester: III	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:
The course objectives of Software Engineering are to enable students to apply engineering principles in software development, demonstrate software project management activities, model software project requirements, design and test software project requirements, implement software development processes, and evaluate standards in process and product to foster a deeper understanding of software engineering principles and practices.

Unit	Content	Hours
1	**Introduction:** Introduction and overview of Software Engineering, Software Crisis, Scope and necessity of software engineering, Software Engineering Processes, Software Development Life Cycle (SDLC) model: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.	9
2	**Software Requirement Analysis & Specification:** Requirement Engineering, Problem Analysis: Data Flow Diagram, Data Dictionaries, ER Diagram, Approaches to Problem Analysis, SRS Document. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.	9
3	**System Design:** Conceptual and Technical Design, Objectives of Design, Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function versus Object Oriented Design, Top-Down and Bottom-Up Design. **Software Measurement and Metrics:** Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.	9
4	**Software Testing:** Software verification & validation, Strategic Approach to Software Testing, Testing Fundamentals Test Plan, Test Design, Test Execution, Reviews, Inspection Auditing, Alpha and Beta Testing of Products, Recent Trends in Software Design/Specialized Software Testing, Related Tools and Standards.	9
5	**Software Maintenance and Software Project Management:** Software Maintenance, Types of Maintenance, Overview of RE-engineering Reverse Engineering, Software Configuration Management, Cost Estimation-Constructive Cost Models (COCOMO), Project Scheduling, Resource Allocation Models, Software Risk Analysis and Management.	9

List of Experiments:

1. To define problem statement of a problem taken and identify the requirements.
2. To draw and design various software development models using starUML (DFD/Class Diagram/Sequence Diagram/ Use Case Diagram/ State Chart Diagram).
3. To estimate various project metrics (calculate Function Point and Complexity of modules etc).
4. Prepare an SRS of the problem undertaken.
5. Prepare various test cases to ensure bug-free execution of all the modules developed in projects.

Learning Resources:

1. Roger Pressman, Software Engineering: A Practitioner‟s Approach, 7th Edition, McGraw Hill
2. Ian Sommerville, Software Engineering, 9th Edition, Addision-Wesley, 2016
3. Pankaj Jalote, A Concise Introduction to Software Engineering, Springer, 2008
4. William E. Lewis , Software Testing and Continuous Quality Improvement, Third Edition, Auerbach Publications, 2008

Course: MCA	**Year: 2**			
Subject Name: Machine Learning	**Subject Code:** CAPCMC303T			
Semester: III	**L**	**T**	**P**	**C**
	3	**1**	**2**	**5**

Course Objectives:
The course objectives of Machine Learning are to help students appreciate the importance of data visualization in data analytics solutions, apply structured thinking to unstructured problems, understand a broad collection of machine learning algorithms and problems, develop a mathematical understanding of machine learning algorithms, and gain an appreciation for the process of learning from data.

Unit	Content	Hours
1	**Introduction to Machine Learning:** Fundamentals of ML, supervised, unsupervised, reinforcement learning; **Supervised Learning:** **Classification**: kNN, Centroid Method, Perceptron, Support Vector Machines, Multi-level Perceptron, Decision tree **Regression:** Linear Regression	**9**
2	**Unsupervised Learning:** **Clustering:** Centroid-based Clustering, Density-based Clustering, Distribution-based Clustering, Hierarchical Clustering **Dimensionality Reduction:** PCA, MDS, ISOMAP, LE, LLE	**9**
3	**Bayesian and Computational Learning** Bayes Theorem, Concept Learning, Maximum Likelihood, Minimum Description Length Principle, Bayes Optimal Classifier, Gibbs Algorithm, Naïve Bayes Classifier, Bayesian Belief Network, EM Algorithm, Probability Learning, Sample Complexity, Finite and Infinite Hypothesis Spaces, Mistake Bound Model.	**9**
4	**Neural Network:** Introduction to neural networks, Fundamental concepts- neuron models and basic learning rules; Single layer neural Networks, input layer, output layer, hidden layers, Multilayer Neural Networks, Backpropagation, Associative Memory	**9**
5	**Deep Learning Techniques:** Gradient Descent; Convolutional Neural Network (CNN)- Convolution, activation, pooling; Receptive Fields and Dilated Convolution, Transposed Convolution; Residual Networks, Dense Networks, Transfer Learning; Self-Organizing Neural Networks, Self-Organizing Feature Maps (SOMs), Best Matching units **Advanced Deep Learning Techniques:** Recurrent Neural Networks (RNN) **-** Long-short term memory (LSTM), Gated RNN**;** Generative Adversarial Networks (GAN)- Generator, Discriminator; Autoencoders **-** sparse, denoising, contractive, stacked Autoencodes	**9**

List of Experiments:

1. To implementing at-least 2 classification algorithms using python/Matlab/R. (KNN/Decision Tree/Perceptron/Centroid Algorithm/SVM etc.)
2. To implementing at-least 2 clustering algorithms using python/Matlab/R. (K-means/BIRCH/DBSCAN etc.)
3. To implementing neural network algorithm using python/Matlab/R.
4. To implementing at-least 2 dimensionality reduction algorithms using python/Matlab/R. (PCA/MDS/ISOMAP/Kernel-PCA etc.)
5. To implementing Naïve-Bayes and Bayesian-Belief Network using python/Matlab/R.

Learning Resources:

1. Kevin Murphy , Machine Learning: a Probabilistic Perspective, 2012.
2. Chris Bishop, Pattern Recognition and Machine Learning, 2006.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, 2016
4. Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.
5. Seth Weidman, Deep Learning from Scratch: Building with Python from First Principles, 2019
6. Rishal Hurbans, Grokking Artificial Intelligence Algorithms, 2020
7. Web Links for Video Lectures: https://www.coursera.org/learn/machine-learning#syllabus