MASTER OF COMPUTER APPLICATIONS-Syllabus

September'22 Session- Semester- I

Semester 1 (S3)

Program	Course Code	Course Name	Credit	Sem	Th /P /Pro
MCA	OMC001	Fundamentals of Computers and Programming	-	1	Th
MCA	OMC101	Object-Oriented Programming using C++	3	1	Th
MCA	OMC102	Data Structures	3	1	Th
MCA	OMC103	Discrete Mathematics	3	1	Th
MCA	OMC104	Operating Systems	3	1	Th
MCA	OMC105	Digital Logic and Computer Organization	3	1	Th
MCA	OMC121	C++ Laboratory	2	1	P
MCA	OMC122	Data Structures Laboratory	2	1	P

Fundamentals of Computers and Programming

Course Code: OMC 001	**Course Title: Fundamentals of Computers and Programming**

Course Objectives:

1. **To differentiate between the varying hardware technologies used over the generations of the computer**
2. **To explain how functional units contribute to a computer's successful operation**
3. **To identify the software and hardware components of the computer**
4. To describe the role of OS in running software programs
5. To understand the fundamentals of C programming and write re-usable code using C functions
6. To demonstrate operations such as sorting, searching using arrays
7. To identify an appropriate C programming construct to solve a given problem

Sr. No	**Units**	**Unit Objectives**
1	**Unit 1: Origin of Computer and its functioning** • Definition of a computer • Characteristics of a computer • Capabilities of a computer • Components of a computer • Functional units of a computer • Evolution • Generations of a computer • Types of computer based on o Usage o Capability o Working	Students will be able to understand the evolution, capabilities and limitations of the computer.

	• Applications in various fields • Limitations	
2	**Unit 2: I/O devices and Memory System** • Input devices such as ○ Keyboard ○ Mouse ○ Joystick ○ Scanner ○ Microphone ○ Webcam • Output devices such as ○ Monitor & its types ○ Printer & its types ○ Speaker ○ Plotter ○ Projector ○ Sound card • Types of memory ○ Primary (RAM, ROM) ○ Secondary (Hard disk) • Memory hierarchy • Storage devices such as ○ Magnetic disk tape ○ Hard disk ○ Floppy disk • Bit, Byte, KB, MB, GB and their relation	Students shall be able to discuss various input and output devices for data communication between the computer and user. Also learn different types of storage devices and their characteristics.
3	**Unit 3: Computer software concepts (Operating System (OS) & Programming language)** • Introduction to Software • Types of Software ○ System Software	Students shall be able to differentiate between the two types of computer software, list the types of OS and identify the two

	○ Application Software • Concept of free & proprietary software • Definition of OS • Need and functions of OS • Types of OS • Definition of a Programming language • Types of Programming languages • Low-level languages • High-level languages • Translators ○ Assembler ○ Compiler ○ Interpreter	main categories of programming languages.
4	**Unit 4: Algorithms** • Definition of Algorithm • Properties of an Algorithm • Examples of Algorithm	Students shall be able to design an algorithm for a given problem.
5	**Unit 5: Flowcharts** • Definition of Flowchart • Advantages of flowchart • Flowchart symbols • Examples of flowchart (conditional statements, loops etc..)	Students shall be able to familiarize the flowchart symbols and design a flowchart for a given problem.
6	**Unit 6: Introduction to C Programming language** • Characteristics of C • Structure of a C program • Life cycle of a C program • First C program – “Hello World” • Commands to execute a C program	Students shall be able to learn the basic structure and elements of a C programming language.

	• C Fundamentals ○ C Character set ○ Constants ○ Variables ○ Identifiers ○ Keywords ○ Escape sequences ○ Data types • Macros	
7	**Unit 7: C I/O Functions** • Types of I/O functions ○ Unformatted ○ Formatted • Unformatted I/O functions ○ getchar() ○ putchar() ○ gets() ○ puts() ○ putch() • Formatted I/O functions ○ Format specifiers ○ scanf() ○ printf()	Students shall be able to explain Read/Write operations on data using different I/O functions.
8	**Unit 8: Operators and Expressions in C** • Arithmetic operators • Relational operators • Logical operators • Assignment operators • Bitwise operators • Increment and Decrement operators • Conditional operator	Students shall be able to evaluate an expression comprising different types of operators.

	• Precedence and Associativity • Evaluation of expressions	
9	**Unit 9: Control statements in C** • Branching statements ○ If statement ○ If then else statement ○ Nested if-else ○ Else-if ladder ○ switch • Looping constructs ○ for ○ while ○ do while • Jump statements ○ break ○ continue ○ goto ○ return	Students shall be able to understand the concepts of branching, looping and how to use it in a programming language.
10	**Unit 10: Arrays and Strings** • Single dimensional array ○ Array declaration ○ Accessing elements of an array ○ Initialization ○ Array operations (insert, delete, sort and search) • Two-dimensional arrays ○ Declaration of a 2D array ○ Initialization ○ Operations on Matrices (addition, product, transpose) • Strings	Students shall be able to declare and define an array to store homogeneous data. Also demonstrate the use of string and string handling functions

	○ Declaration of strings ○ Initialization ○ Input and Output of strings ○ Formatting strings ○ Edit set conversion code ○ String handling functions	
11	**Unit 11: Functions and Storage classes in C** • Advantages of functions • Library functions • User-defined functions ○ Function declaration or prototype ○ Function definition ○ Function call • Return statement • Types of functions ○ Function with no arguments and no return value ○ Function with no arguments but with a return value ○ Function with arguments but no return value ○ Function with arguments and return value • Call by value and call by reference • Passing array to a function • Recursion ○ Advantages & disadvantages ○ Iteration v/s Recursion ○ Examples (Factorial, Fibonacci series...) • Storage classes ○ Automatic	Students shall be able to demonstrate different ways of passing parameters to a function and return a value from a function.

	○ Static ○ Register ○ External ○ Scope of variables with different storage classes	
12	**Unit 12: Structures and Unions** • Need of structures • Defining a structure • Declaration of structure variables • Initialization of structure variables • Accessing structure members • Assignment of structure variables • Size of a structure • Array of structures • Structure with arrays • Nested structure • Structures and functions • Self-referential structures • Typedef keyword • Bitfields • Defining a union • Declaration of union variables • Initialization and access of union variables • Size of a union • Nested unions ○ Union inside union ○ Structure inside union ○ Union inside structure • Difference between Structure and Union	Students shall be able to explain how to use structures and unions to store heterogeneous data in a program.

13	**Unit 13: C Pointers** • Need of Pointers • Pointer variables ○ Declaration ○ Assigning address to pointer variables ○ Dereferencing pointer variables • Pointer arithmetic • Pointer comparisons • Chain of Pointers • Pointers and Array • Pointers and String • Pointers and Function • Array of Pointers • Void pointers • Null pointers • Dangling pointer • Dynamic memory allocation ○ malloc() ○ calloc() ○ realloc() ○ free()	Students shall be able to discuss on how to declare pointers and use them for dynamic memory allocation. Also understand the pointer arithmetic with arrays.
14	**Unit 14: File management in C** • Types of files • File modes • Open a file • Close a file • End Of File (EOF) • Character I/O functions ○ fputc() ○ fgetc()	Students shall be able to discuss how to open, close a file. Read/write a character, string, integer and a block from/to a file using file handling functions.

	• Integer I/O functions ○ putw() ○ getw() • String I/O functions ○ fputs() ○ fgets() • Formatted I/O functions ○ fprintf() ○ fscanf() • Block Read/Write functions ○ fwrite() ○ fread() • Random access to a file ○ fseek() ○ ftell() ○ rewind() • Error handling in files • Command line arguments	

Text Book(s)

1. Norton & Peter, “Introduction to Computers”, 6th Edition, McGraw-Hill, 2009
2. Leon, Alexis & Leon, Mathews, “Introduction to Computers”, 1st Edition, Vikas Publishing, 2009
3. Yashavant Kanetkar, “Let Us C”, 14th Edition, BPB Publication, 2016
4. E. Balagurusamy, “Programming in ANSI C”, 6th Edition, McGraw-Hill, 2015

Reference Book(s)

1. P K.Sinha & Priti Sinha, “Computer Fundamentals”, 8th Edition, BPB, 2004
2. Brian W Kernighan & Dennis M Ritchie, “The C Programming Language”, 2nd Edition, Prentice Hall, 1988

Object-Oriented Programming Using C++

Course Code: OMC 101	**Course Title: Object-Oriented Programming Using C++**

Course Objectives:

1. **Understand the difference between the top-down and bottom-up approach**
2. **Describe the object-oriented programming approach in connection with C++**
3. **Apply the concepts of object-oriented programming**
4. **Illustrate the process of data file manipulations using C++**
5. **Apply virtual and pure virtual function & complex programming situations**

Sr. No	**Units**	**Unit Objectives**
1	**Unit 1: Object Oriented Programming** • Procedure Oriented vs. Object Oriented Programming, • Object Oriented Programming Concepts, • Benefits of Object oriented programming, • Object Oriented Languages.	Students will learn about the significance of Object Oriented Programming Languages.
2	**Unit 2: Introduction to C++** ▪ Structure of a C++ program ▪ Data Types ▪ Operators and Control Structures.	Students will be able to learn the representation of Data in C++ and basic programming structures.
3	**Unit 3: Objects and Classes** ▪ Defining Classes ▪ Encapsulation ▪ Instantiating Objects ▪ Member Functions	Students will be able to Define Classes, Hiding Data, and Binding Data with Functions.

	▪ Accessibility labels ▪ Static Members	
4	**Unit 4: Constructors and Destructors** ▪ Purpose of Constructors ▪ Default Constructor ▪ Parameterized Constructors ▪ Copy Constructor, Destructor ▪ Memory Management	Students will learn to initialize the objects and deallocate the memory whenever an object is destroyed.
5	**Unit 5: Inheritance** ▪ Concept of Reusability ▪ Types of Inheritance ▪ Single and Multiple Inheritance ▪ Multilevel Inheritance	Students will learn to reuse, extend or modify the attributes and behaviors which are defined in other class/classes.
6	**Unit 6: Operator Overloading** ▪ Function and Operator Overloading ▪ Overloading Unary and Binary Operators	Students will learn to use operators to work for user-defined classes.
7	**Unit 7: Polymorphism and Virtual Function** ▪ Abstract Class ▪ Function Overriding ▪ Dynamic Binding ▪ Pure Virtual Functions	Students will be able to write a function that can evaluate or be applied to values of different types and redefine a function of the base class in the derived class.
8	**Unit 8: Streams and Files Stream Classes** ▪ Types of I/O ▪ Formatting Outputs ▪ File Pointers ▪ Buffer	Students will learn to access data from files and format the data.
9	**Unit 9: Templates and STL**	Students will be able to create a single function or single

	▪ Function and Class Templates ▪ Use of Templates ▪ Standard Template Library	class to work with different data types using templates
10	**Unit 10: Exception Handling** ▪ Exceptions in C++ Programs ▪ Try and Catch Expressions ▪ Exceptions with argument	Students will learn to handle run-time anomalies or abnormal conditions that a program encounters during its execution.

Text Book(s)

1. E. Balagurusamy, “Object Oriented Programming with C++”, 4th edition, Tata McGraw Education Hill, 2011.
2. Herbert Schildt – The Complete Reference C++, Fourth Edition, Tata MgGraw Hill, 2003

Reference Book(s)

1. Robert Lafore - Object Oriented Programming with C++, Fourth Edition, Pearson Education India, 2008
2. Stephen Prata, “C++ Primer Plus: Teach Yourself Object Oriented Programming”, 2nd Edition, Waite Group, 1995

Data Structures

Course Code: OMC 102	Course Title: Data Structures

Course Objectives:

1. Describe the concept of Data Structures and assess how the choice of data structures impacts the performance of programs
2. Compare and contrast the merits and demerits of various data structures in terms of time and memory complexity.
3. Identify and propose appropriate data structures for providing the solution to real-world problems.
4. Implement operations like searching, insertion, deletion, traversing mechanism, etc. on various data structures
5. Be familiar with advanced data structures such as balanced search trees, hash tables, AVL trees, priority queues, ADT, etc.
6. To augment the merits of particular data structures on other data structures to develop innovation in the subject of study.

Sr. No	Units	Unit Objectives
1	**Unit 1: Introduction** • Basic Terminology • Pointer and dynamic memory allocation • Elementary Data Organization • Algorithm Complexity and Time-Space trade-offs	Students will learn Dynamic Memory Management, Data Organisation, and Efficiency.
2	**Unit 2: Arrays** ▪ Array Definition ▪ Representation and Analysis ▪ Single and Multidimensional Arrays ▪ Sparse Matrices	Students will be able to learn and use Single and Multidimensional Arrays.
3	**Unit 3: Recursion** ▪ Definition ▪ Tail recursion	Students will be able to develop programs using Recursion.

4	**Unit 4: Stacks** ▪ Array Representation of stack ▪ Linked Representation of Stack ▪ Infix, Prefix, and Postfix Expressions	Students will be able to develop programs using Stacks for various applications.
5	**Unit 5: Queues** ▪ Array and linked representation and of queues ▪ Circular queue ▪ D-queue ▪ Priority Queue	Students will be able to learn different types of Queues and use them to develop programs for various applications.
6	**Unit 6: Linked list** ▪ Representation of Singly Linked Lists ▪ Two-way Header List ▪ Doubly linked list ▪ Generalized linked list	Students will be able to represent data using Linked Lists and develop programs for various applications.
7	**Unit 7: Trees** ▪ Binary Trees ▪ Binary Search tree ▪ Algebraic Expressions ▪ Complete Binary Tree ▪ Extended Binary Trees ▪ Threaded Binary trees ▪ AVL Tree ▪ Huffman algorithm & Huffman tree	Students will be able to represent data using Trees, learn to balance the trees, and code the information using Huffman Trees.
8	**Unit 8: Searching and Hashing** ▪ Sequential search ▪ Binary search ▪ Hash Table ▪ Hash Functions ▪ Collision Resolution Strategies	Students will be able to use different searching techniques.

9	**Unit 9: Sorting** ▪ Insertion Sort ▪ Bubble Sorting ▪ Quick Sort ▪ Two Way Merge Sort ▪ Heap Sort	Students will be able to use different sorting methods to arrange the data.
10	**Unit 10: Graphs** ▪ BFS ▪ DFS ▪ Spanning tree ▪ Minimum Spanning tree, Kruskal’s Algorithm, Prim’s Algorithm ▪ Applications of graph	Students will be able to represent the data using Graphs and develop applications.
11	**Unit 11: File Structures** ▪ Physical Storage Media ▪ File Organization, Organization of records into Blocks ▪ Sequential Files ▪ Indexing ▪ Primary indices ▪ Secondary indices ▪ B+ Tree index Files ▪ B Tree index Files ▪ Indexing and Hashing Comparisons	Students will be able to learn different techniques of Data Organisation.

Text Book(s)

1. Horowitz and Sahani, “fundamental of data structures”, Galgotia Publishers, 1999
2. A.M Tenenbaum et al, “data structures and program design in C”, 2nd edition, Pearson education, 2006

Reference Book(s)

1. Lipschutz, “data structures”, 1st Edition, Tata McGraw Hill Education,2014
2. R. Kruse et al, “data structures and program design in C”, 2nd Edition, Pearson India,2006

3. K Loudon, “Mastering algorithm with C”, O'Reilly publishers, 1999

Discrete Mathematics

Course Code: OMC 103		**Course Title: Discrete Mathematics**
Course Objectives: 1. Apply the fundamentals of set theory for the given problem. 2. Solve the given problem by applying Logical Reasoning. 3. To prove the correctness of programs using Logical Reasoning. 4. Apply techniques of counting in the analysis of algorithms. 5. Apply Binary relations to establish relationships between different entities.		
Sr. No	**Units**	**Unit Objectives**
1	**Unit 1: Fundamentals** ▪ Sets and Subsets ▪ Operations on sets ○ Algebraic properties of set operations ○ The Addition principle ○ The Addition Principle for Disjoint sets ▪ Sequences ○ Finite and Infinite sequences ○ Characteristic Functions ○ Computer Representation of sets and subsets ○ Strings and Regular Expressions	Students will be able to understand the background to begin the exploration of Mathematical Structures.
2	**Unit 2: Finite Automata** ▪ Deterministic Finite Automata (DFA) ▪ Non Deterministic Finite Automata (NFA)	Students will be able to design the simplest abstract machines to recognize patterns
3	**Unit 3: Logic** • Propositions and Logical Operations ○ Logical Connectives and Compound Statements	Students learn the methods of reasoning and use them to prove theorems in mathematics and prove the

	○ Quantifiers • Conditional Statements • Methods of Proof • Mathematical Induction	correctness of algorithms in computer science.
4	**Unit 4: Counting** ▪ Permutations ▪ Combinations • Pigeonhole Principle	Students will be able to apply the techniques of counting in the analysis of algorithms.
5	**Unit 5: Relations and Digraphs** ▪ Product Sets and Partitions ▪ Relations and Digraphs ○ Relations ○ Sets Arising from Relations ○ The Matrix of a Relation ○ Digraphs ○ Paths in Digraphs ▪ Properties of Relations ○ Reflexive and Irreflexive Relations ○ Symmetric, Asymmetric, and Antisymmetric Relations ○ Transitive Relations ▪ Equivalence Relations ○ Equivalence Relations and Partitions	Students will be able to learn and develop binary relations among different entities.
6	**Unit 6: Functions** ▪ Functions ○ Special Types of Functions ○ Invertible Functions ▪ Functions for Computer Science ○ Hashing Functions	Students will be able to understand the concept of functions and use them as basic building blocks in real-life applications.

7	**Unit 7: Graph Theory** ▪ Graphs o Graphs o Subgraphs and Quotient Graphs ▪ Euler paths and Circuits ▪ Hamiltonian Paths and Circuits ▪ Coloring of Graphs o Chromatic Polynomials	Students will be able to develop real-time applications.
8	**Unit 8: Order Relations and Structures** ▪ Partially Ordered Sets o Hasse Diagrams o Topological Sorting o Isomorphism	Students will be able to construct the logical representation of computer circuits.
9	**Unit 9: Trees** ▪ Trees ▪ Labelled Trees o Computer Representation of Binary Positional Trees ▪ Tree Searching o Searching General Trees o Pseudocode Versions	Students will be able to develop applications using tree searching methods.
10	**Unit 10: Semigroups and Groups** ▪ Binary Operations o Tables o Properties of Binary Operations ▪ Semigroups o Isomorphism and Homomorphism ▪ Products and Quotients of Semigroups ▪ Groups.	Students will be able to apply the basics of group theory in coding theory.

Text Book(s):

1. Kollman, Busby and Ross, ”Discrete Mathematical Structures”, 3rd edition, Pearson Publishers, 2015
2. Gremaldi, Ramana, ”Discrete and Combinatorial Mathematics”, 5th edition, Pearson Publishers, 2006

Reference Book(s):

1. Susanna Epp, “Discrete Mathematics with Applications”, 4th Edition, Wadsworth Publishing Co Inc, 2010
2. Kenneth Rosen, “Discrete Mathematics and its Applications”,8th Edition, McGraw Hill, 2021

Operating Systems

Course Code: OMC 104		**Course Title: Operating Systems**
Course Objectives: Students will learn various services provided by operating systems and the functions performed by the different components of operating systems.		
Sr. No	**Units**	**Unit Objectives**
1	**Unit 1: Introduction** • Definition ○ What is an OS ○ Goals of OS • Components of Computer System • Types of operating systems ○ Batch Operating System ○ Multiprogramming Operating System ○ Multitasking / Time-Sharing Operating System ○ Real Time Operating System ○ Distributed Operating System ○ Parallel Operating System ○ Embedded Operating System • Operating system components ○ Process Management ○ Memory Management ○ File Management ○ Secondary Storage Management ○ Networking ○ Protection System • Services of Operating System ○ Program execution ○ I/O/ operation	Students will be able to define the operating system and identify different types of operating systems, various components, and services provided by operating systems.

	○ File system manipulation ○ Communication ○ Error detection ○ Resource allocation ○ Accounting ○ Protection ○ User interface • System Calls ○ Definition explanation in brief ○ Introduction to Threads	
2	**Unit 2: Process Concept** • Process definition • Process Control Block • Process States • Scheduling queues • Schedulers ○ Long term schedulers ○ CPU schedulers • Context switch • CPU scheduling criteria • Scheduling algorithms ○ FCFS ○ SJF ○ RR ○ Priority Scheduling	Students will be able to demonstrate different states of a process and evaluate CPU scheduling algorithm performance.
3	**Unit 3: Inter-process communication** • Communication ○ Shared memory ○ Message passing • Process Synchronization • Background	Students will be able to explain communication among processes and examine process synchronization

	• The Critical-Section problem ○ Two process solutions (general) • Semaphores ○ Types • Classical problems of synchronization (explanation without algorithm) ○ Readers' and writers' problem	
4	**Unit 4: Deadlock** • System model • Characterization ○ Necessary conditions ○ Resources allocation graph ○ Methods for handling deadlock • Deadlock prevention • Avoidance and Detection ○ Safestate • Bankers algorithm • Recovery from deadlock ○ Process termination ○ Resource preemption	Students will be able to compare the deadlock problems and their solutions.
5	**Unit 5: Memory Management** • Memory management ○ Overlays • Logical and Physical Address Space • Swapping • Contiguous allocation ○ Single partition ○ Multiple partitions ▪ First fit ▪ Best fit ▪ Worst fit	Students will be able to select a memory management scheme for the specific system.

	• Paging ○ Hardware support ○ Paging hardware with TLB • Segmentation ○ Definition ○ Difference between paging and segmentation	
6	**Unit 6: Virtual Memory** • Virtual Memory Background • Demand paging ○ Page fault • Page replacement algorithms ○ FIFO ○ LRU ○ Optimal • Thrashing ○ Definition	Students will be able to evaluate the cost and complexity of demand paging.
7	**Unit 7: File Systems** • File concept ○ Attributes ○ File operations • Access methods ○ Direct access ○ Sequential access • Directory structure ○ introduction	Students will be able to distinguish various aspects of files and directory structure.
8	**Unit 8: Disk Management, Protection and Security** • Disk structure • Disk scheduling methods. ○ FCFS (with Example) ○ SSTF (introduction)	Students will be able to review the disk scheduling algorithms and interpret the protection mechanism

	○ SCAN ○ C-SCAN ○ LOOK • Protection and Security ○ Goals of protection ○ Domain of protection ○ Access matrix	provided by the operating system.

Text Book(s)

1. Silberschatz, Galvin" Operating Systems Concept, 7th edition, John Wiley and sons

Reference Book(s)

1. William Stalling: Operating Systems: Internal and design principles, 7th edition PHI
2. D M Dhamdhere, "Operating Systems: A Concept-Based Approach" TMH

Digital Logic and Computer Organization

Programme	Master of Computer Applications
Semester	I
Course Title	Digital Logic and Computer Organization
Course Code	OMC105
Course Credits	3
Course Type	Core Theory Course

1. Course Summary

This course aims to provide the learners with a foundation of Digital logic and computer organization. The initial part of this course deals with data representation, basic components of digital systems, and their implementation. Students learn Boolean algebra, logic gates, implementation of combinational and sequential digital circuits using logic gates, and use of k-map for logic expression minimization. The later part discusses computer generations, functional components, their characteristics, performance, and interactions. Students learn addressing modes, input-output, and memory organization. Students gain an understanding of fundamental architectural techniques used to build today's high-performance processors and systems.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. List types of digital circuits and functional units of a computer [L-1]
CO-2. Explain the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division [L-2]
CO-3. Demonstrate the hierarchical memory system [L-3]
CO-4. Analyze the different ways of I/O operations and their effect on performance [L-4]
CO-5. Design simple logic circuits using gates, multiplexers, demultiplexers, encoders, and decoders[L-6]

3. Course Contents

Sr. No	Units	Unit Outcomes After the successful completion of the unit, the learner should be able to:
1	**Unit 1: Data Representation** • Number Systems-Binary, Octal, Decimal, Hexadecimal, Interconversion • Binary Addition and Subtraction • Representation of Negative Numbers-1's Complement, 2's Complement	1. List the different formats to represent numerical data and convert from one to another. 2. Demonstrate 1's- complement and 2's complement representation of negative integers. 3. Identify different character codes

2	**Unit 2: Logic Circuits** • Logic gates • Types of Logic Gates-Basic Gates, Universal gates, Special gates • Boolean Algebra- Use of Boolean Algebra, Identities of Boolean Algebra, DeMorgan’s Theorems, Forms of Boolean Expressions, Minterms, and Maxterms • Simplification of Logic Expressions-Introduction to K-maps, Minimization Using K-maps	1. Represent logic gates using logic symbols, truth tables, and logic expressions. 2. Explain De Morgan’s theorems. 3. Simplify logic expressions using laws and identities of Boolean Algebra. 4. Apply Karnaugh maps to obtain minimal expressions
3	**Unit 3: Combinational and Sequential Circuits** • Combinational Circuits-Adders and Subtractors, Decoder, Encoder, Multiplexer, Demultiplexer • Sequential Circuits • Flip-Flops-Types of Flip Flops • Registers • Counters	1. Implement the following circuits: Half adder and Full adder Multiplexer and Demultiplexer Decoder and encoder 2. Represent the different flip-flops using truth tables.
4	**Unit 4: The Computer and Generations** • The von Neumann Computer • Development of Computer Hardware-The First Generation, The Second Generation, The Third Generation, The Fourth Generation, Beyond the Fourth Generation	1. Explain the functional units of the Von Neumann machine 2. Explain computer evolution through various generations
5	**Unit 5: Basic Structure of Computers** • Computer Types • Functional Units • Basic Operational Concepts • Bus structure • Performance • Multiprocessing and Multi computers	1. Explain basic functional units and organization of computer. 2. Explain the use of a bus for data transfer. 3. Identify the various factors affecting performance.
6	**Unit 6: Machine Instructions and Programs**: • Memory Locations and Addresses • Memory Operations • Instructions and Instruction sequencing • Addressing Modes.	1. Outline the different concepts related to the Instructions of a computer. 2. Explain the instructions sequencing. 3. Identify the various addressing modes
7	**Unit 7: Input/Output Organization**: • Accessing I/O Devices • Interrupts • Enabling and Disabling Interrupts • Handling Multiple Devices • Controlling Device Requests • Direct Memory Access	1. Demonstrate the different ways of accessing I/O devices. 2. Explain the interrupt mechanism 3. Explain DMA operation
8	**Unit 8: Memory System**: • Basic Concepts, • Semiconductor RAM Memories, • Read Only Memories, • Speed, Size, and Cost, • Cache Memories • Virtual Memory	1. Identify the different types of memory technology. 2. Explain memory and how the memory hierarchy reduces effective memory latency. 3. Explain the role of cache memory.
9	**Unit 9: Basic Processing Unit:**	1. Explain processor organization

	• Some Fundamental Concepts, • Execution of Complete Instruction, Multiple Bus Organization, • Hard-wired Control, • Micro programmed Control	2. Explain multiple bus organizations. 3. Differentiate hard-wired control and microprogrammed control
10	**Unit 10: Fundamentals of computer architecture and pipelining** • VonNeumann vs. Harvard • RISC vs. CISC • Importance of RISC • Pipelining: Basic concepts of pipelining	1. Explain the various architectures 2. Highlight the importance of RISC architecture 3. Explain the importance of pipelining.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	**3**										**2**		**3**	**2**	**2**	**2**
CO-2	**3**										**2**		**3**	**2**	**2**	**2**
CO-3	**3**	**3**	**1**								**2**		**3**	**2**	**2**	**2**
CO-4	**3**	**3**	**2**	**2**							**2**		**3**	**2**	**2**	**2**
CO-5	**3**	**3**	**3**	**3**					**1**		**3**		**3**	**3**	**2**	**2**
3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution																

5. Course Resources

a. Essential Reading

1. Course Self-Learning Material
2. M Morris Mano: Digital Logic and Computer Design, 4th Edition, Pearson Education, 2014.
3. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, 5th Edition, Tata McGraw Hill, 2017.

b. Recommended Reading

1. William Stallings, (2016) *Computer Organization & Architecture*, 10th ed., PHI.
2. Charles H. Roth Jr and Larry L Kinney, (2014) *Fundamentals of Logic Design*. 7th ed., Cengage Learning.

c. Magazines and Journals

1. IEEE transaction on Very Large-Scale Integration (VLSI) Systems
2. IEEE transactions on Parallel and Distributed Systems

d. Websites

1. https://www.coursera.org/

2. http://nptel.ac.in/

e. Other Electronic Resources

1. https://ocw.mit.edu/index.htm
2. Course Video Lectures on ILearn