DEBRA THANA SAHID KSHUDIRAM SMRITI MAHAVIDYALAYA

Gangaram Chak, Chak Shyampur, Debra, West Bengal

PROPOSED SYLLABUS (DRAFT) OF

MAJOR COURSE UNDER CCFUP, 2023 FOR SEMESTER-III & IV

FOR COMPUTER SCIENCE(MAJOR) PROGRAMMES

(w.e.f. Academic Year 2025-2026)

Based on

Curriculum & Credit Framework for Undergraduate Programmes (CCFUP), 2023 & NEP, 2020

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit		L-T-P	Marks		
									CA	ESE	TOTAL
B.Sc. (Hons.)	2nd	III	SEMESTER-III								
			Major-3	UG/III/COMP/3/MJ-3T	T: Data Structure	4	3	3-0-1	15	40	75
				UG/III/COMP/3/MJ-3P	P: Data Structure Practical		1			20	
			Major-4	UG/III/COMP/3/MJ-4T	T: Computer Architecture	4	3	3-0-1	15	40	75
				UG/III/COMP/3/MJ-4P	P: Practical		1			20	
			SEC	UG/III/COMP/3/SE-3P	P: Android Programming	3		0-0-3	10	40	50
			AEC	AEC03 Communicative English -2 (common for all programmes)	AEC03 Communicative English -2 (common for all programmes)	2		2-0-0	10	40	50
			MDC	MDC03 Multidisciplinary Course -3 (to be chosen from the list)	MDC03 Multidisciplinary Course -3 (to be chosen from the list)	3		3-0-0	10	40	50
			Minor 3(Disc.-I)	UG/III/COMP/3/MI-3T	T: Digital Logic	4		4-0-0	15	60	75
					Semester-I Total	20					375

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit		L-T-P	Marks		
									CA	ESE	TOTAL
B.Sc. (Hons.)	2nd	IV	SEMESTER-IV								
			Major-5	UG/IV/COMP/4/MJ-5T	T: OOPs using C++	4	3	3-0-1	15	40	75
				UG/IVCOMP/4/MJ-5P	P: C++ Practical		1			20	
			Major-6	UG/IV/COMP/4/MJ-6T	T: Operating System	4	3	3-0-1	15	40	75
				UG/IV/COMP/4/MJ-6P	P: Operating System Practical		1			20	
			Major-7	UG/IV/COMP/4/MJ-7T	T : Computer Network	4	3	3-0-1	15	40	75
				UG/IV/COMP/4/MJ-7P	P : Computer Network		1			20	
			AEC	AEC04	MIL-2 (common for all programs)	2		2-0-0	10	40	50
			Minor 4(Disc.-II)	UG/IV/COMP/4/MI-4	T: Data Structure	4	3	3-0-1	15	60	75
					p : Practical		1			20	
			Summer Intern.	INT	Internship/ Apprenticeship - activities to be decided by the Colleges following the guidelines to be given later	4		0-0-4	-	-	50
					Semester-IV Total	24					400

MAJOR 3

UG/III/COMP/3/MJ-3: Data Structure Credit 04

OBJECTIVE OF THE COURSE

- Gain knowledge of different types of data structures and their applications in computer science.
- Evaluate time and space complexity using growth rates, order notation, and space analysis techniques.
- Implement and manipulate single and multi-dimensional arrays and understand the concept of pointers and their applications.
- Develop stack-based applications, perform infix, postfix, and prefix conversions, and understand recursion with its advantages and limitations.
- Implement and analyze singly, doubly, and circular linked lists along with their various applications in memory management and self organizing lists.
- Understand and implement different types of queues, including circular queues, de-queues, and priority queues
- Implement graph representations, traverse graphs using BFS and DFS, and apply shortest path algorithms.
- Implement different tree structures such as binary trees, threaded binary trees, and AVL trees, along with tree traversal techniques.
- Apply linear and binary search techniques, implement various sorting algorithms, and understand hashing mechanisms for efficient data retrieval.

UG/III/COMP/4/MJ-3T: Data Structure **Credits 03**

Course contents:

Module-I Introduction to Data Structures **04 Hrs.**

Introduction to Algorithm and Flowcharts, Analysis for Time and Space Requirements- Rate of growth, Basic Time Analysis of an Algorithm, Order notation, Space Analysis of an Algorithm. Various types of Data Structure, Static and Dynamic Memory Allocation, Function.

Module- II: Arrays and Pointers **05 Hrs.**

Introduction to Arrays, Single and Multi-dimensional Arrays, Pointer, Pointer to Structure, Sparse Matrices (Array and Linked Representation).

Module- III: Stacks and Recursion **08 Hrs.**

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack. Developing Recursive Definition of Simple Problems

and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Module- IV: Linked Lists **10 Hrs.**

Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

Module- V: Queues **05 Hrs.**

Array and Linked representation of Queue, De-queue, Priority Queues

Module- VI: Graphs **05 Hrs.**

Introduction, Representation to Graphs, Graph Traversals Shortest Path Algorithms.

Module- VII: Trees **15 Hrs.**

Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height- Balanced Trees (Various operations on AVL Trees). Tree traversal techniques.

Module- VIII: Searching, Sorting and Hashing **08 Hrs.**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Comparison of Sorting Techniques. Hashing technique, Collision resolution, chaining Different types of hash function.

UG/III/COMP/4/MJ-3P: Data Structures Lab **Credits 01**

Module-I: Introduction to Data Structures

1. Write a program to analyze the time complexity of a given algorithm using order notation.
2. Implement a program to calculate the space complexity of an algorithm.

3. Demonstrate static and dynamic memory allocation for an integer array using C/C++.

Module-II: Arrays & Pointers

4. Implement a program to perform matrix addition, subtraction, and multiplication using arrays.
5. Write a program to implement sparse matrix representation using arrays and linked lists.
6. Implement a program to demonstrate pointer arithmetic and pointer-to-structure operations.
7. Write a program to copy and reverse an array using pointers instead of indexing.

Module-III: Stacks & Recursion

8. Implement a stack using arrays and perform push, pop, and peek operations.
9. Implement a program to check if a given expression has balanced parentheses using a stack.
10. Convert an infix expression to a postfix expression using a stack.
11. Evaluate a postfix expression using a stack.
12. Write a recursive function to compute factorial, Fibonacci sequence, and greatest common divisor (GCD).
13. Implement a recursive function to solve the Tower of Hanoi problem.
14. Implement a recursive function for binary search.

Module-IV: Linked Lists

15. Implement a singly linked list with insertion, deletion, and traversal operations.
16. Implement a doubly linked list with insertion, deletion, and traversal operations.
17. Implement a circular linked list and demonstrate its operations.
18. Implement a program to merge two sorted linked lists.
19. Implement a self-organizing linked list using the move-to-front heuristic.

Module-V: Queues

20. Implement a queue using an array and perform enqueue and dequeue operations.
21. Implement a queue using a linked list and perform enqueue and dequeue operations.
22. Implement a circular queue and demonstrate its advantages over a linear queue.
23. Implement a priority queue using a heap data structure.

Module-VI: Graphs

24. Implement a graph using adjacency matrix and adjacency list representations.
25. Implement Breadth-First Search (BFS) and Depth-First Search (DFS) traversal of a graph.
26. Implement Dijkstra's algorithm for the shortest path in a graph.
27. Implement Floyd-Warshall or Bellman-Ford algorithm for all-pairs shortest paths.

Module-VII: Trees

28. Implement a binary search tree (BST) with insertion, deletion, and traversal operations.
29. Implement inorder, preorder, and postorder traversal of a binary tree (recursive and iterative).
30. Implement an AVL tree with insertion and deletion operations.
31. Implement a threaded binary tree and perform its traversal.

Module-VIII: Searching, Sorting, and Hashing

32. Implement linear search and binary search and compare their performance.
33. Implement selection sort, bubble sort, insertion sort, and quick sort, and compare their performance.
34. Implement merge sort and heap sort and compare their time complexities.
35. Implement hashing with chaining to handle collisions.

36. Implement open addressing techniques (linear probing, quadratic probing, and double hashing).

Suggested Readings:

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++, Second edition, PHI, 2009.
4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson, 1999.
5. D.S Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
7. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java, 2003.
8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 editions, 2009
10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley, 2013
11. Herbert Schildt, "Java the Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
12. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.

MAJOR 4

UG/III/COMP/3/MJ-4: Computer Architecture Credit 04

OBJECTIVE OF THE COURSE

- Understand the fundamental concepts of logic design, Boolean algebra, combinational and sequential circuits, and their applications in digital systems.
- Analyse data representation techniques, including number systems, fixed and floating-point arithmetic, and basic computer arithmetic operations.
- Explain the organization and operation of basic computer components, including registers, bus systems, memory, and instruction execution cycles.

- Differentiate between CPU architectures (RISC & CISC), instruction formats, addressing modes, and control unit design (hardwired & microprogrammed).
- Explore memory hierarchy, including cache memory, and understand various input-output mechanisms such as programmed I/O, interrupts, and DMA.
- Develop hands-on skills in HDL-based digital design by implementing arithmetic circuits, memory units, basic CPU architecture, and instruction cycle simulations.

UG/III/COMP/4/MJ-4T: Credit 03

Module I: Introduction 20 Lecture.
Logic gates, Boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.

Module II: Data Representation and Basic Computer Arithmetic 10 Lecture.
Number systems, complements, fixed and floating-point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers

Module III: Basic Computer Organization and Design 10 Lecture.
Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, Organization of a basic single-bus computer system.

Module IV: Central Processing Unit 10 Lecture.
Register organization, arithmetic and logical operations, Instruction formats, addressing modes, instruction codes, machine language, assembly language, RISC, CISC architectures, Hardwired and micro programmed control unit design.

Module V: Memory Organization 6 Lecture.
Memory interfacing and addressing, cache memory organization.

Module VI: Input-Output Organization 6 Lecture.
Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

Suggested Readings:

1. M. Mano, Computer System Architecture, Pearson Education 1992
2. W. Stallings, Computer Organization and Architecture Designing for Performance, 8 Edition, Prentice Hall of India,2009
3. M.M. Mano , Digital Design, Pearson Education Asia,2013
4. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

UG/III/COMP/4/MJ-4P: Computer Architecture **Credits 01**

1. Introduction to HDL – Basics of Hardware Description Languages (VHDL/Verilog).
2. Basic Digital Logic Design – Implement basic logic gates and simple combinational circuits.
3. Arithmetic Circuit Design – Implement 8-bit addition, multiplication, and division.
4. Register and Memory Unit Design – Design an 8-bit register and memory unit; perform basic read/write operations.
5. 8-bit ALU Design – Implement a simple ALU with basic arithmetic and logic operations.
6. Basic CPU Design – Design a simple 8-bit CPU and interface it with memory.
7. Instruction Cycle Implementation – Develop a basic instruction fetch routine and simulate execution.
8. Microoperations and Instruction Set Design – Create and simulate microoperations for a simple instruction set.

SEC-3

UG/III/COMP/3/SE-3P: Android Programming Credit 03

OBJECTIVE OF THE COURSE

- Learn the fundamentals of Android development.
- Build user-friendly mobile applications using Java/Kotlin.
- Understand UI/UX design, database management, and API integration.
- Develop, debug, and publish Android applications.

Android Programming:

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the emulator. Also display “Hello World” in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.
9. Build an application that plays an audio file when a button is clicked and stops when another button is clicked.
10. Develop a list of all courses in your college and display the teacher-in-charge at the bottom when a course is selected.
11. Create a simple to-do list application where users can add and remove tasks dynamically.
12. Create an application with three option buttons, where selecting a button changes the background color of the screen.

MAJOR 5

UG/IV/COMP/4/MJ-5T: OOPs using C++ Credit 04

OBJECTIVE OF THE COURSE

- Understand the principles of Object-Oriented Programming (OOP) and compare it with structured programming.
- Develop proficiency in C++ fundamentals, including data types, control structures, functions, and user-defined data types.
- Implement encapsulation, constructors, destructors, and object manipulation techniques in C++.
- Apply concepts of polymorphism and inheritance, including operator overloading, virtual functions, and abstract classes.
- Utilize pointers, dynamic memory allocation, and exception handling for efficient program development.
- Gain hands-on experience in solving real-world problems using C++ through practical implementation of OOP concepts.

UG/IV/COMP/4/MJ-5T **Credits 03**

Module-I: Introduction to OOPs and C++ Element **15 Lectures.**

Structured vs. Object Oriented Programming, Object Oriented Programming Concepts, Benefits of Object oriented programming, Object Oriented Languages, Structure of a C++ program, Data Types, Operators and Control Structures, Iteration / Loop Construct, Arrays, Functions (User defined Function, Inline Function, Function Overloading), User Defined Data Types (Structure, Union and Enumeration).

Module II: Class, Object, Constructor & Destructor: **15 Lectures.**

Defining Classes, Encapsulation, Instantiating Objects, Member Functions, Accessibility labels, Static Members, Friend Function, Purpose of Constructors, Default Constructor, Parameterized Constructors, Copy Constructor, Destructor.

Module III: Pointer, Polymorphism & Inheritance: **20 Lectures.**

Pointer (Pointer to Object, this Pointer, Pointer to Derive Class), Introduction to Polymorphism (Compile time Polymorphism, Run time Polymorphism), Operator Overloading, Overloading Unary and Binary Operators, Virtual Function, Pure Virtual Functions, Inheritance (Single Inheritance, Multiple Inheritance, Multilevel Inheritance, Hierarchical Inheritance, Hybrid Inheritance), Virtual Base Class, Abstract Class.

Module IV: Exception Handling: **10 Lectures.**

Exceptions in C++ Programs, Try and Catch Expressions, Exceptions with arguments

Suggested Readings:

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
2. BjarneStroustroup, "Programming -- Principles and Practice using C++", 2nd Edition, Addison- Wesley 2014.
3. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
4. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
5. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
6. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley , 2000.
7. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley,2005.
8. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by Addison-Wesley, 5th Edition, 2012
11. E Balagurusamy , Object Oriented Programming with C++, 5 th edition, Tata McGraw, 2011.
12. Deitel and Deitel , “C++: How to Program”, 9th Edition, Pearson, 2013.

UG/IV/COMP/4/MJ-5P **Credits 01**

1. Write a C++ program to find the sum of individual digits of a positive integer.
2. Write a C++ program to print the given number in reverse order.
3. Write a C++ program to print first 100 non-Fibonacci numbers.
4. Write a C++ program to convert a decimal number into a hexadecimal number.
5. Write a C++ program to search an element of an array using binary search technique.
6. Write a C++ program to calculate compound interest in a bank using default arguments.
7. Write a C++ program to display the student details using classes and object as array.
8. Write a C++ program to implement stack using array.

9. Write a C++ program for matrix multiplication using dynamic memory allocation, copy construction and overloading of assignment operator.
10. Write a C++ program to read a two-dimensional matrix and display its transpose.
11. Write a C++ program to implement inline function.
12. Write a C++ program to implement constructor and destructor.
13. Write a C++ program to implement the functionalities of a copy constructor.
14. Write a C++ program to display the account number and balance using constructor overloading.
15. Write a C++ program to find the volume of cube, rectangle and cylinder using function overloading. 16. Write a C++ program to overload operator ++ and operator - using friend functions.
16. Write a C++ program to add two complex numbers using binary operator overloading.
17. Write a C++ program to implement single inheritance and multilevel inheritance.
18. Write a C++ program to draw a rectangle, square and circle using multiple inheritance with virtual function.
19. Write a C++ program to implement hybrid inheritance.
20. Write a C++ program to display student details using virtual base class.
21. Write a C++ program to implement pure virtual function.

Reference Books:

1. E. Balaguruswami-Object Oriented programming with C++
2. Kris James-Success with C++
3. David Parsons-Object Oriented programming with C++
4. D. Ravichandran-Programming in C++
5. Dewhurst and Stark-Programming in C++

MAJOR 6

UG/IV/COMP/4/MJ-6: Operating System Credit 04

OBJECTIVE OF THE COURSE

- Provide a comprehensive understanding of fundamental concepts and functions of modern operating systems.
- Cover the architecture and components of operating systems, including process management,
- memory management, file systems, and input/output systems.
- Teach about concurrency, process synchronization, and inter-process communication for efficient task management.
- Emphasize the role of operating systems in resource allocation and system security.
- Offer hands-on experience in implementing and configuring operating system features through labs and projects.
- Explore various operating systems like Windows, Linux, and macOS to understand their differences and commonalities.
- Delve into virtualization and distributed systems to highlight current trends in operating system design.
- Equip students with the skills to analyze, design, and optimize operating systems, preparing them for advanced study and careers in systems programming and software engineering.

UG/IV/COMP/4/MJ-6T: Credit 03

Module I: Introduction 10 Hrs.

Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

Module II: Operating System Organization 6 Hrs.

Processor and user modes, kernels, system calls and system programs.

Module III: Process Management 16 Hrs.

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent processes, critical section, semaphores, methods for inter- process communication, Deadlocks: Detection, prevention, avoidance (Banker’s Algorithm).

Module IV: Memory Management **10 Hrs.**

Memory hierarchy and memory allocation strategies, Paging and Segmentation, Virtual Memory: Demand paging, Page replacement algorithm (FIFO, LRU, Optimal), Thrashing and working set model.

Module V: File Systems and Storage Management **8 Hrs.**

File concepts, access methods, and directory structures, File allocation methods: Contiguous, Linked, and Indexed, Disk scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, RAID levels and storage hierarchy.

Module VI: **I/O Systems and Device Management** **8 Hrs.**

I/O hardware and software structure, Interrupts, Direct Memory Access (DMA), Device drivers and disk management.

Module VII: Security and Protection **6 Hrs.**

Basics of OS security: Authentication and Authorization, Threats and vulnerabilities in OS.

Suggested Readings:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

UG/IV/COMP/4/MJ-6P: **Credit 01**

Use C/ C++/ Python:

1. Write program to implement FCFS scheduling algorithm.
2. Write program to implement Round Robin scheduling algorithm.
3. Write program to implement SJF scheduling algorithm.
4. Write program to implement Priority scheduling algorithm.
5. Implement a solution to the Producer-Consumer problem using semaphores.
6. Write a program to implement the Reader-Writer problem using mutex locks.
7. Implement Banker’s Algorithm to check for deadlock avoidance in a multi-process system.
8. Implement the First Fit, Best Fit, and Worst Fit memory allocation strategies and compare their efficiency.

MAJOR 7

UG/IV/COMP/4/MJ-7: Computer Network Credit 04

OBJECTIVE OF THE COURSE

- Learn about different network types, topologies, classifications, and protocols, along with layered network architectures such as the OSI and TCP/IP models.
- Understand the roles of network devices like hubs, switches, routers, repeaters, and gateways, as well as different transmission modes (simplex, half-duplex, full-duplex).
- Explore analog and digital signals, transmission media, data encoding schemes, multiplexing techniques, and error detection and correction mechanisms.
- Learn about circuit switching, packet switching (datagram and virtual circuit switching), and data transfer technologies like DSL and cable modems.
- Study framing techniques, flow control mechanisms, and error control protocols such as Stop-and-Wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ.
- Understand CSMA/CD, Ethernet LANs, and backbone networking devices such as repeaters, switches, bridges, and routers.
- Learn about IPv4/IPv6 addressing, subnetting, CIDR, and routing algorithms like Distance Vector and Link State, along with protocols such as RIP, OSPF, and BGP.
- Study TCP and UDP, their functionalities, connection establishment mechanisms (three-way handshake), flow control, and congestion control techniques.
- Gain knowledge about DNS, HTTP, SMTP, POP3, and IMAP, and understand how these protocols facilitate web and email communication.

UG/IV/COMP/4/MJ-7T: Credit 03

Module I: Introduction to Computer Networks 8 Hrs.

Basics of Computer Networks; network topologies (Bus, Star, Ring, Mesh, Hybrid); network classifications; network protocol; layered network architecture; overview of OSI reference model (Layers and Their Functions); overview of TCP/IP protocol suite, Comparison with OSI Model. Network Devices: Hub, Switch, Router, Repeater, Gateway. Transmission Modes: Simplex, Half-duplex, Full-duplex.

Module II: Data Communication Fundamentals and Techniques **10 Hrs.**

Analog and digital signal; Transmission Media: Wired (Twisted Pair, Coaxial, Fiber Optic) and Wireless (Radio Waves, Microwaves, Infrared), data-rate limits; Encoding schemes NRZ, Manchester, Differential Manchester; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques- FDM, TDM,WDM; Error Detection and Correction: Parity Check, Hamming Code, CRC.

Module III: Networks Switching Techniques and Access mechanisms **10 Hrs.**

Circuit switching; packets witching- connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

Module IV: Data Link Layer Functions and Protocol **10 Hrs.**

Framing Techniques, Flow Control: Stop-and-Wait, Sliding Window Protocol Error detection, error correction and Error Control: ARQ (Automatic Repeat reQuest) - Stop-and-Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ; Point to Point Protocol on Internet.

Module V: Multiple Access Protocol and Networks **5 Hrs.**

CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways.

Module VI: Networks Layer Functions and Protocols **6 Hrs.**

IPv4 and IPv6 Addressing: Classful and Classless Addressing, Subnetting, CIDR, Routing; routing algorithms Distance Vector, Link State; network layer protocol RIP, OSPF, BGP, Internet control protocols. Network Address Translation (NAT).

Module VII: Transport Layer Functions and Protocols **6 Hrs.**

Transmission Control Protocol (TCP): Features, Connection Establishment and Termination, Flow Control, Congestion Control; Transport services- error and flow control, Connection establishment and release – three-way handshake; User Datagram Protocol (UDP): Features and Comparison with TCP.

Module VIII: Overview of Application layer protocol **5 Hrs.**

Domain Name System (DNS) and its Working overview of DNS protocol; Overview of WWW &HTTP protocol. Email Protocols: SMTP, POP3, IMAP.

Suggested Readings:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM, 2007.
2. A. S. Tanenbaum: Computer Networks**,** Fourth edition**,** PHI, 2002.

UG/IV/COMP/4/MJ-7P: **Credit 01**

Use C/ C++/ Python:

1. **Transmission Modes Simulation** – Write a program to demonstrate simplex, half-duplex, and full-duplex transmission using message passing.
2. **Bit Stuffing & Byte Stuffing** – Implement bit stuffing and byte stuffing algorithms for given input data.
3. **CRC (Cyclic Redundancy Check)** – Implement CRC error detection algorithm for a given data frame and polynomial.
4. **Hamming Code for Error Correction** – Write a program to implement Hamming Code for detecting and correcting a single bit error.
5. **LAN Topology Simulation** – Simulate Bus, Star, and Ring topologies and demonstrate basic data communication between nodes.
6. **Stop-and-Wait ARQ Protocol** – Implement Stop-and-Wait ARQ for reliable data transmission.
7. **Sliding Window Protocol** – Implement Go-Back-N and Selective Repeat ARQ protocols.
8. **CSMA/CD (Carrier Sense Multiple Access with Collision Detection)** – Simulate CSMA/CD mechanism for multiple devices communicating over a network.
9. **Implementation of Distance Vector Routing Algorithm** – Simulate how routers exchange information using Distance Vector

Routing (Bellman-Ford Algorithm).

10. **Subnetting Calculator** – Write a program to take an IPv4 address and subnet mask as input and calculate network ID, broadcast address, and valid host range.
11. **Socket Programming - Basic Chat Application** – Implement a client-server chat application using TCP sockets.
12. **Socket Programming - File Transfer** – Implement a simple file transfer protocol using TCP/UDP sockets.
13. **DNS Query Resolver** – Write a program that takes a domain name as input and resolves its IP address using a DNS query.
14. **TCP Congestion Control Simulation** – Implement Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery algorithms.
15. **Firewall Packet Filtering Simulation** – Develop a simple firewall that filters packets based on predefined rules (e.g., allow/block specific IPs and ports).

MINOR 3

UG/III/COMP/3/MI-3T: Digital Logic Credit 04

OBJECTIVE OF THE COURSE

- Learn the basics of binary, octal, decimal, and hexadecimal number systems and conversions between them.
- Master the principles of Boolean algebra, including logic operations, truth tables, and De Morgan's laws.
- Gain proficiency in simplifying Boolean expressions using techniques such as Karnaugh maps and the Quine-McCluskey method.
- Develop skills in designing basic combinational logic circuits, including adders, subtractors, multiplexers, and decoders.
- Understand the behaviour and design of sequential circuits, including flip-flops, latches, counters, and registers.
- Explore circuit minimization techniques to reduce complexity and cost in digital designs.

Number systems: **15 Hrs.**

Positional number systems; Binary, Octal, Hexadecimal, and Decimal number systems; conversion of a number in one system to the other; Representation of signed numbers-signed magnitude, one's complement, 2's complement representation techniques, Merits of 2's complement representation scheme; Various binary codes - BCD, excess -3, Gray code, binary addition and subtraction.

Boolean algebra: **15 Hrs.**

Fundamental of Boolean Expression: Definition of Boolean Algebra, Postulates, Basic Logic gates: (OR, AND, NOT); Universal Logic Gates: (NAND & NOR); Basic logic operations: logical sum (OR), logical product (AND), complementation (NOT), anti-coincidence (EX-OR) and coincidence (EX-NOR) operations: Truth tables of Basic gates; Boolean Variables and Expressions; De-Morgan’s theorem; Boolean expressions Simplification- Algebraic technique, Karnaugh map technique, 3 variable and 4 variable Karnaugh map.

Combinational Circuits: **15 Hrs.**

Half Adder, Full Adder (3-bit), Half Subtractor, Full Subtractor (3-bit), and construction using Basic Logic Gates (OR, AND, NOT) and Universal Logic Gates (NAND & NOR), Multiplexer, Encoders, Demultiplexer, and Decoder circuits.

Sequential Circuits: **15 Hrs.**

Latch, RS, D, JK, T Flip Flops; Race condition, Master Slave JK Flip Flop; Registers: Serial Input Serial Output (SISO), Serial Input Parallel Output (SIPO), Parallel input Serial Output (PISO), Parallel Input Parallel Output (PIPO), Universal Shift Registers; Counters:

Asynchronous Counter, Synchronous Counter.

Suggested Readings:

1. Morris Mano, Charles R. Kime, Logic and computer design fundamentals, Pearson Prentice Hall, 2004
2. Basavaraj,B., Digital fundamentals, New Delhi: Vikas Publishing House, 1999.
3. Kandel Langholz, Digital Logic Design, Prentice Hall, 1988.
4. Rafiquzzaman & Chandra, Modern Computer Architecture, West Pub. Comp., 1988.

MINOR 4

UG/IV/COMP/4/MI-4: Data Structure Credit 04

OBJECTIVE OF THE COURSE

- Gain knowledge of different types of data structures and their applications in computer science..
- Evaluate time and space complexity using growth rates, order notation, and space analysis techniques..
- Implement and manipulate single and multi-dimensional arrays and understand the concept of pointers and their applications.
- Develop stack-based applications, perform infix, postfix, and prefix conversions.
- Implement and analyze singly, doubly, and circular linked lists along with their various applications in memory management.
- Understand and implement different types of queues, including circular queues, de-queues, and priority queues.
- Implement different tree structures , along with tree traversal techniques.
- Apply linear and binary search techniques, implement various sorting algorithms.

UG/IV/COMP/4/MI-4T **Credit 03**

Module-I Introduction to Data Structures **04 Hrs.**

Introduction to Algorithm and Flowcharts, Analysis for Time and Space Requirements, Order notation, Space Analysis of an Algorithm. Static and Dynamic Memory Allocation.

Module- II Arrays and Pointers **05 Hrs.**

Introduction to Arrays, Single and Multi-dimensional Arrays, Pointer, Pointer to Structure, Sparse Matrices (Array and Linked Representation).

Module- III Stacks **08 Hrs.**

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack.

Module- III Linked Lists **10 Hrs.**

Singly, Doubly and Circular Lists (Array and Linked representation).

Module- IV Queues **05 Hrs.**

Array and Linked representation of Queue, De-queue, Priority Queues.

Module- VI Trees **15 Hrs.**

Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Search Trees); Definition of different types of trees with examples. Tree traversal techniques.

Module- VII Searching, Sorting **08 Hrs.**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort.

UG/IV/COMP/4/MI-4P **Credit 01**

1. Implement a program to perform matrix addition, subtraction, and multiplication using arrays.

2. Implement different type of operations on array

3. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.

4. Implement different type of operation on Singly Linked List.

5. Implement Singly, Doubly, Circular Linked List.

6. Perform Stack operations using Linked List and array implementation.

7. Implement queue operation using Linked List and array implementation.

8. Implement Linear Search and binary search.

9. Implement selection sort, insertion sort.

Suggested Readings:

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++, Second edition, PHI, 2009.
4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson, 1999.
5. D.S Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
7. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java, 2003.
8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 editions, 2009
10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley, 2013
11. Herbert Schildt, "Java the Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
12. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.