SYLLABUS FOR COMPUTER SCIENCE (MAJOR)

Under Single Major Single Minor (FYUGP)

(To be implemented from Session 2024-25)

SEM. III & IV

Proposed Syllabus for four years B.Sc. Computer Science (Major) Programme							
Year	**Semester**	**Paper Code**	**Paper**	**Credits**	**Periods/Week**	**Exam. Marks**	**Total Marks**
2nd Year	III	COMSMAJ305	Programming with C++	3	3	60	80
		COMSMAJ305L	Programming with C++ (Lab)	1	4	20	
		COMSMAJ306	Computer Architecture and Organization	3	3	60	80
		COMSMAJ306L	Computer Architecture and Organization (Tutorial)	1	4	20	
		POOCSEC332 POOCSEC333	Fundamentals of Computer Networking MS Excel	2	2	40	60
		POO CSEC332L POO CSEC333L	Fundamentals of Computer Networking (Lab) MS Excel (Lab)	1	2	20	
		MIN3	Student will be provided the SAME discipline from the subjects selected previously as Minor.	3	3	60	80
		MIN3 (L/T)	Minor Practical/Tutorial	1	4	20	
		AEC1	Any one from MIL (Bengali/Nepali/Hindi/Urdu/Sanskrit/Alternative English)	3	3	60	80
		AEC1T	MIL (Tutorial)	1	2	20	
		IDC2	Student has to choose only ONE discipline from a pool of subjects that will be provided.	2	2	40	60
		IDC2T	Inter Disciplinary (Tutorial)	1	1	20	
	IV	COMSMAJ407	Operating System	3	3	60	80
		COMSMAJ407T	Operating System (Lab)	1	1	20	
		COMSMAJ408	Data Structure	3	3	60	80
		COMSMAJ408L	Data Structure (Lab)	1	2	20	
		MIN4	Student will be provided the SAME discipline from the subjects selected previously as Minor.	4	4	60	80
		MIN4 (L/T)	Minor Practical/Tutorial	1	4	20	
		VAC2	Understanding India	3	3	60	80
		VAC2T	Understanding India (Tutorial)	1	2	20	
		IDC3	Student has to choose only ONE discipline from a pool of subjects that will be provided.	2	2	40	60
		IDC3T	Inter Disciplinary (Tutorial)	1	1	20	
		IN1	Summer Internship The Colleges are expected to network with skill development centres, vocational training institutes for facilitating student internships. Online based internships programs are also permitted in case of Computer Science (Major) students. The students must submit a certificate of completion of the internship at the end of the semester.	2	-	-	

NOTE: Tutorials should involve problem solving session/activity related to the subject taught.

2nd Year
Semester-III

Course-MAJOR	**Paper Code- COMSMAJ305**	**Credits-3**	**Lectures/Week-3**
Paper:	**Programming with C++**		

Brief Course Description:
This course introduces the principles of object-oriented programming using C++.
It covers core concepts such as classes, objects, inheritance, and polymorphism.
Students will develop the ability to write efficient, modular, and reusable code using C++.

Prerequisite(s) and/or Note(s):

1. Basic knowledge of programming logic at the high school level.
2. Prior exposure to any programming language like C is helpful but not mandatory.
3. **Note(s):**
   - The syllabus is subject to minor modifications depending on institutional or academic requirements.
   - Students will be assessed only on the basis of topics covered in class during the semester.

Course Objectives

Knowledge Acquired:

1. Understand the fundamental principles of object-oriented programming (OOP).
2. Learn the syntax and semantics of the C++ programming language.
3. Develop a conceptual understanding of classes, objects, inheritance and polymorphism.
4. Learn how to implement modular, reusable, and maintainable code using OOP concepts.

Skills Gained:

1. Ability to write efficient, error-free C++ programs using object-oriented techniques.
2. Application of arrays, functions, classes, and inheritance to solve real-world problems.
3. Skills in operator overloading, polymorphism, and working with pointers and memory management.
4. Ability to use C++ constructs to design small-scale systems or components.

Competency Developed:

1. Ability to break down a problem into objects and define appropriate class structures.
2. Capability to handle complexity through encapsulation and modularity.
3. Confidence in handling object lifecycle (construction/destruction) and memory allocation.
4. Readiness for further studies in data structures, algorithms, and system-level programming using C++ or similar languages.

Syllabus Overview

Unit 1:	**Language Fundamental**	**10 Lectures**

Overview of OOP: The Object-Oriented paradigm, Basic concepts of OOP, Benefits of OOP, Object oriented languages, Application of OOP
Overview of C++: Evolution of C++ from C, Structure of a C++ program Data Types: Built-in data types, User-defined data types, Derived data types, Keywords, Constants and Variables: symbolic constants, Dynamic initialization of variable, Reference variable. Operators and expressions, Input and output using cin, cout, Control Structures: if-else, nested if-else, while, do-while, for, break, continue, switch, goto statement.

Unit 2: Arrays, Structure & Functions **10 Lectures**

Arrays: Single and multi-dimensional arrays, Character arrays (C-style strings), Introduction to C++ String class
Structures: A Simple structure, defining a structure variable, Accessing structure's member, Enumeration data type.
Function: Function Declaration, Calling Function, Function Definition, Passing Arguments to function: Passing Constant, Passing Value, Reference Argument, Structure as argument, Default Argument. Returning values from function: Return statement, Returning structure variable, Return by reference, Overloaded Function, Inline Function.

Unit 3: Object and Classes **10 Lectures**

Object and Class: Defining the class and its member, Making an outside function inline, nesting of member function, array as class member, Difference between classes and structures.
Memory allocation: Memory allocation for objects, new and delete operator, static data member, static member functions, object as function argument.
Constructor & Destructor: Constructors and types of constructors (Default constructor, Copy Constructor, Parameterized constructor), Destructors.

Unit 4: Pointers and Inheritance **7 Lectures**

Pointers: Introduction, & and * operator, pointer to object, this pointer, pointer to derived class.
Inheritance: Inheritance- definition, concept, Types of Inheritance, Visibility modes- Public, Private, Protected, Virtual Base Class, Benefits of Inheritance.

Unit 5: Polymorphism **8 Lectures**

Static Polymorphism: Operator overloading, Operator keyword, overloading unary operators (++ (pre increment and post increment), –) using operator function, overloading binary operators (+, -, =, >=, <=, +=, <, >, []). Friend function, Friend class, overloading binary operators using friend function and without using friend function.
Dynamic polymorphism: Virtual function, Pure Virtual Function, Abstract class.

Suggested Readings

1. 1. Object Oriented Programming with C++ : E. Balagurusamy, The McGraw-Hill
2. Let Us C++: Yashwant Kanetkar, BPB Publications
3. The C++ Programming Language: Bjarne Stroustrup, Addision Wasley.
4. Object Oriented Programming in C++ : Robert Lafore, Galgotia Publications.

Course-MAJOR Paper:	**Paper Code- COMSMAJ305L Programming with C++ (Lab)**	**Credits-1**	**Lab hours/Week-2**

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Write a program to display your name, age, and course using cout.
2. Write a program to calculate the area of a rectangle using user input.
3. Write a program to check whether a number is even or odd using if-else.
4. Write a program to find the largest of three numbers using nested if-else.
5. Write a program to print the multiplication table of a number using a for loop.
6. Write a program to find the sum of digits of a number using a while loop.
7. Write a program to find the maximum and minimum elements in an array.
8. Write a program to reverse an array.
9. Write a program to sort an array using bubble sort.

10. Write a program to check whether a string is a palindrome (use character array).
11. Write a program using a structure to store and display student details (name, roll, marks).
12. Write a function to find the factorial of a number using recursion.
13. Write a program to overload a function sum() to calculate sum of 2, 3, or 4 integers.
14. Write a program that passes a structure as a function argument and displays data.
15. Write a program to create a class Rectangle with data members length and breadth, and member functions to calculate area and perimeter.
16. Write a program to create a class Student with constructor and destructor that displays a message when invoked.
17. Write a program to demonstrate use of static data members and static functions in a class.
18. Write a program to pass an object as an argument to a function.
19. Write a program to demonstrate the use of pointers with variables and arrays.
20. Write a program to create a base class Person and a derived class Student. Use inheritance to input and display details.
21. Write a program to implement multilevel inheritance (e.g., Person → Employee → Manager).
22. Write a program to demonstrate use of this pointer.
23. Write a program to overload the + operator to add two complex numbers.
24. Write a program using virtual functions to demonstrate runtime polymorphism.
25. Write a program to create an abstract class Shape with a pure virtual function area(), and derive Circle and Rectangle from it.

Course-MAJOR	**Paper Code-COMSMAJ306**	**Credits-3**	**Lectures/Week-3**
Paper:	**Computer Architecture and Organization**		

Brief Course Description:

The course in Computer System Architecture aims to provide an in-depth understanding of various architectural designs in computing systems. It explores design decisions, architectural trade-offs, and their impact on system performance. It prepares students to analyze and compare computing architectures for enhanced efficiency and performance.

Prerequisite(s) and/or Note(s):

1. Familiarity with fundamental concepts in digital electronics and logic design.
2. **Students who have not taken Digital Systems** will need to do additional background reading on:
   - Combinational and sequential circuits
   - Assembly language basics
3. **Note(s):**
   - Course content may be revised during the term to align with academic priorities.
   - Students will be evaluated only on the topics covered in class.

Course Objectives:

Knowledge Acquired:

1. Understanding the structure and operation of computer systems at the architectural level.
2. Knowledge of CPU organization, memory hierarchy, instruction sets, and data path design.
3. Ability to comprehend the performance impact of architectural choices.
4. Understanding the design principles of input/output systems, control units, and memory systems.

Skills Gained:

1. Ability to analyse the working of different components in a computer system.
2. Proficiency in tracing instruction execution through data path and control logic.
3. Skills in comparing and evaluating architectural models based on performance metrics.

Competency Developed:

1. Capability to design and evaluate simple CPU architectures.
2. Ability to apply knowledge of memory organization, cache systems, and pipelining for system optimization.
3. Competency in hardware-software interaction, enhancing understanding of how software affects system performance.
4. Readiness for roles in system architecture design, embedded systems, performance analysis, and hardware-software co-design.

Syllabus Overview

Unit 1:	**Register transfer and microoperations**	**6 Lectures**

Register transfer, Arithmetic microoperations, logic microoperations, shift microoperations, Computer registers, Need of Registers, bus system, Interconnection Structures, Bus Interconnection.

Unit 2:	**Basic Computer Organization and Design**	**7 Lectures**

Instruction set, timing and control, instruction cycle, hardwired instruction format, memory reference, register reference and input-output instructions. Design of basic computer, Interrupt.

Unit 3:	**Central Processing Unit**	**10 Lectures**

Stack organization, microprogrammed control. Microprogrammed instruction formats, addressing modes, instruction codes, machine language, assembly language, design of CPU, RISC, CISC architectures.

Unit 4:	**Computer Arithmetic**	**8 Lectures**

Addition and Subtraction, Multiplication algorithm (Boots), Division algorithm, Floating point arithmetic operations.

Unit 5:	**Memory and Input-Output Organization**	**10 Lectures**

Memory hierarchy, Main memory, Cache memory, Virtual memory. Input/Output Operations, Serial and Parallel Data Transfers, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access.

Suggested Readings

1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology,2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8thEdition, Prentice Hall of India,2009
4. M. M. Mano, Digital Design, Pearson Education Asia,2013
5. Carl Hamacher, Computer Organization, Fifth edition, McGraw Hill, 2012.

Course-MAJOR	**Paper Code-COMSMAJ306L**	**Credits-1**	**Tutorial hours/Week-2**
Paper:	**Computer Architecture and Organization (Tutorial)**		

Computer System Architecture tutorials as assigned and advised by teacher(s).

Course- SEC	Paper Code-POOCSEC332	Credits-2	Lectures/Week-2
Paper:	Fundamentals of Computer Networking		

Prerequisite(s) and/or Note(s):

1. Basic Computer Literacy: Students should be familiar with fundamental computer operations such as using the keyboard, mouse, files, folders, and basic software applications.
2. No Prior Networking Knowledge Required: This is an introductory course, designed for beginners.
3. Recommended for: Undergraduate students from Computer Science, IT, Electronics, or any discipline with interest in Information Technology.

Course Objectives:

1. To introduce students to the basic concepts and purpose of computer networks.
2. To explain the different types of networks, devices, transmission media, and communication protocols.
3. To provide a foundational understanding of network architectures such as the OSI and TCP/IP models.
4. To familiarize students with basic IP addressing, configuration of small networks, and networking tools.
5. To develop practical skills for designing, setting up, and troubleshooting basic wired and wireless networks.

Knowledge Acquired:

By the end of this course, students will have gained theoretical and practical understanding of:

1. The role and components of computer networks in modern communication.
2. Types of networks (LAN, WAN, MAN), topologies, and data transmission media.
3. Hardware components like routers, switches, hubs, access points, NICs, and their uses.
4. Networking protocols such as TCP/IP, HTTP, FTP, SMTP, DNS, and DHCP.
5. Network security fundamentals including firewall basics, encryption, and safe internet practices.

Skills Gained:

1. Identify and differentiate between types of networks, hardware, and topologies.
2. Setup, configure, and troubleshoot a basic wired or wireless network.
3. Use basic networking tools and commands like ping, ipconfig, tracert, and netstat.
4. Share resources like printers and files over a local network, Configure Wi-Fi routers and manage basic internet sharing settings.
5. Implement safe browsing and basic network security practices.

Competency Developed:

1. Understanding the foundational structure and operation of computer networks.
2. Performing hands-on tasks involved in configuring small network environments.
3. Communicating effectively using networking terminology and concepts.
4. Applying practical problem-solving techniques in network setup and troubleshooting.
5. Enhancing employability for roles such as IT support technician, helpdesk executive, or junior network administrator.

Syllabus Overview

Unit 1: Introduction to Computer Networks 12 Lectures

Definition and objectives of computer networks; Advantages and applications of networking; Types of networks: LAN, MAN, WAN, PAN; Network topologies: Bus, Ring, Star, Mesh, Hybrid; Modes of transmission: Simplex, Half Duplex, Full Duplex; Categories of networks: Peer-to-Peer vs. Client-Server; Basic network architecture: OSI Model (overview) and TCP/IP model (layers and functions)

Unit 2: Networking Hardware and Transmission Media 6 Lectures

Networking devices: NIC (Network Interface Card), Switch, Hub, Router, Bridge, Gateway, Repeater, Access Point; Transmission media - Guided: Twisted Pair, Coaxial Cable, Fiber Optic, Unguided -Wi-Fi, Bluetooth, Infrared, Satellite, Concept of signal types: Analog and Digital signals; IP addressing: IPv4 and IPv6 (Basics only)

Unit 3: Network Protocols and Configuration 5 Lectures

Concept of protocol and protocol stacks; Important networking protocols: TCP/IP, HTTP/HTTPS, FTP, SMTP, POP3, DNS and DHCP; IP Addressing schemes, subnetting (basic concepts); Understanding firewalls and antivirus for network protection.

Unit 4: Network Setup, Troubleshooting, and Security Basics 7 Lectures

Steps to create a small network (wired/wireless LAN setup); Configuring a Wi-Fi router and sharing Internet connection; File and printer sharing over LAN; Common network troubleshooting tools and commands: ping, ipconfig, tracert, nslookup, netstat; Introduction to cybersecurity basics in networking Safe browsing practices, securing Wi-Fi, network encryption (WPA/WPA2); Introduction to Virtual Private Networks (VPN)

Suggested Readings

1. Data Communications and Networking, Author: Behrouz A. Forouzan, Publisher: McGraw-Hill Education.
2. Computer Networks, Author: Andrew S. Tanenbaum & David J. Wetherall, Publisher: Pearson Education
3. Computer Networking: A Top-Down Approach, Author: James F. Kurose & Keith W. Ross Publisher: Pearson
4. Networking All-in-One For Dummies (8th Edition), Author: Doug Lowe, Publisher: Wiley
5. Introduction to Networking, Author: Bruce Hallberg, Publisher: McGraw-Hill

Supplementary Materials and Online Resources:

1. Cisco Networking Essentials – Craig Berg Hands-on book with basic configuration and troubleshooting for beginners.
2. CompTIA Network+ Guide to Networks – Jill West, Tamara Dean, Jean Andrews Good for learners aiming at industry certification and real-world examples.
3. Cisco Networking Academy – https://www.netacad.com Free structured courses by Cisco for beginner to intermediate networking students.
4. GCF LearnFree – Computer Networking – https://edu.gcfglobal.org/en/computers Interactive tutorials and beginner-level explanations of networking concepts.
5. Microsoft Learn – Networking Fundamentals – https://learn.microsoft.com

Course-SEC	Paper Code-POOCSEC332L	Credits-1	Lab hours/Week-2
Paper:	Fundamentals of Computer Networking (Lab)		

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Create a diagram illustrating LAN, MAN, and WAN with examples.
2. Identify and compare various network topologies: Star, Ring, Bus, Mesh.
3. Simulate peer-to-peer and client-server communication using two computers.
4. Categorize different types of networks based on geographic coverage and purpose.
5. Label and explain the seven layers of the OSI model using a real-world analogy.
6. Create a table comparing OSI vs TCP/IP model with examples for each layer.
7. Identify different networking devices (hub, switch, router, NIC, modem) and state their purpose.
8. Connect two computers using Ethernet crossover cable and test connectivity.
9. Demonstrate the use of a Wi-Fi dongle and access point to connect a PC wirelessly.
10. Create a visual layout of different network cables (coaxial, twisted pair, fibre optic) and describe their use.
11. Test signal strength in various areas using a Wi-Fi analyser tool.
12. Create a presentation showing the differences between analog and digital transmission signals.
13. Use ping to test the connectivity between two devices on a network.
14. Demonstrate tracert or traceroute to trace the route to a public domain (e.g., www.google.com).
15. Demonstrate the use of nslookup command to find IP addresses of domains.
16. Demonstrate the use of email protocols (SMTP/POP/IMAP) using a mail client.
17. Set up a small wired LAN for file sharing between 3 computers.
18. Install and configure a Wi-Fi router with password.
19. Enable and disable Windows firewall and observe its effect on ICMP (ping) traffic.
20. Use netstat to observe active connections on a system and interpret the results.

Course- SEC	Paper Code- POOCSEC333	Credits-2	Lectures/Week-2
Paper:	MS Excel		

Prerequisite(s) and/or Note(s):

1. Basic knowledge of using a computer and operating systems (Windows/macOS)
2. Familiarity with basic mathematical operations
3. No prior experience with MS Excel is required Course Objectives:
1. To provide comprehensive knowledge of Microsoft Excel from basic to advanced features
2. To enable students to efficiently manage, analyze, and visualize data
3. To develop problem-solving skills using formulas, functions, and tools in Excel
4. To prepare students for professional and academic tasks involving spreadsheet usage

Knowledge Acquired:

1. Understanding of the Excel interface and navigation
2. Concepts of workbook, worksheets, cell references, ranges, and data types
3. Knowledge of formatting, sorting, filtering, and conditional formatting
4. Deep understanding of formulas and functions (logical, statistical, lookup, text, date/time)
5. Knowledge of data visualization tools such as charts and graphs.

Skills Gained:

1. Creating, formatting, and managing worksheets and workbooks
2. Applying and combining functions for calculations and data analysis
3. Using charts and graphs to represent data visually and effectively
4. Collaborating and protecting workbooks with shared access and security features
5. Importing/exporting data from other sources

Competency Developed:

1. Ability to use MS Excel efficiently in academic, personal, and professional contexts
2. Proficiency in data management and decision-making based on spreadsheet analysis
3. Enhanced productivity through automation and advanced Excel tools
4. Readiness for administrative, business analytics, accounting, data entry, and reporting roles
5. Foundation for certification exams like Microsoft Office Specialist (MOS) – Excel

Syllabus Overview

Unit 1:	**Introduction to MS Excel**	**12 Lectures**

Overview of spreadsheet software and applications; Getting started with Excel: interface, ribbons, toolbars, and menus; Creating, saving, and opening workbooks; Understanding cells, rows, columns, ranges ; Data entry: text, numbers, dates; Basic formatting: font, alignment, number formats, borders, and colours, Introduction to worksheet management (insert, delete, rename, move)

Unit 2:	**Working with Formulas and Functions**	**7 Lectures**

Cell referencing: relative, absolute, and mixed references; Basic arithmetic formulas and operators; Common Mathematical functions: SUM, AVERAGE, PERCENTAGE, MIN, MAX, and COUNT; Logical: IF, AND, OR; Text: CONCATENATE, LEFT, RIGHT, LEN, TRIM; Date/Time: TODAY, NOW, DAY, MONTH, YEAR

Unit 3:	**Data Management and Presentation**	**5 Lectures**

Sorting and Filtering data; Conditional Formatting, Data validation and drop-down lists, Charts and Graphs: Column, Bar, Line, Pie charts; Formatting chart elements; Printing worksheets and setting page layouts

Unit 4:	**Practical Applications and Productivity Tools**	**5 Lectures**

Using Templates in Excel; Working with Multiple Sheets; Introduction to Tables and basic data analysis; Freeze Panes, Split View, and Zoom features; Basic PivotTable creation and interpretation; Protecting worksheets and workbooks; Exporting data (PDF, CSV)

Suggested Readings

1. Microsoft Excel 2021 Step by Step , *Author:* Curtis Frye, *Publisher:* Microsoft Press
2. Excel 2021 Bible, *Author:* Michael Alexander, Richard Kusleika, *Publisher:* Wiley
3. Excel Basics In 30 Minutes (2nd Edition), *Author:* Ian Lamont, *Publisher:* i30 Media Corporation
4. Microsoft Excel Data Analysis and Business Modeling, *Author:* Wayne L. Winston *Publisher:* Microsoft Press
5. Excel All-in-One For Dummies, *Author:* Greg Harvey, *Publisher:* Wiley

Course-SEC	Paper Code- POOCSEC333L	Credits-1	Lab hours/Week-2
Paper:	MS Excel (Lab)		

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Create a new workbook and save it with the name Student_Info.xlsx.
2. Enter the details of 10 students: Name, Roll No, Class, and Marks in 3 subjects.
3. Format the worksheet by applying borders, font styles, and background color.
4. Rename the worksheet as “Student Data” and insert a header and footer.
5. Adjust column widths and row heights to fit the data properly.
6. Use the Wrap Text and Merge & Center options for a title row.
7. Copy data from one sheet to another and move a worksheet within the workbook.
8. Insert a new row and column in your data and input additional student information.
9. Calculate the Total Marks and Average Marks of each student using formulas.
10. Apply the IF function to assign grades (e.g., IF(Average ≥ 60, "Pass", "Fail")).
11. Use the MAX, MIN, and COUNT functions on marks.
12. Use the ROUND, ROUNDUP, and ROUNDDOWN functions on average marks.
13. Use TEXT functions like LEFT, RIGHT, MID, and CONCATENATE on student names.
14. Insert the current date using the TODAY() function and show how many days have passed since a given date using DATEDIF().
15. Apply Sorting to arrange the students in ascending order of total marks.
16. Apply Conditional Formatting to highlight marks below 40 in red.
17. Create a Pie Chart showing percentage-wise marks of a single student.
18. Freeze the top row and the first column to make headings visible while scrolling.
19. Prepare a print-friendly version of your worksheet using Page Layout settings.
20. Create a new sheet and apply a Template for a monthly expense tracker.
21. Create a PivotTable to summarize total marks by class.
22. Protect a worksheet with a password and try editing locked cells.
23. Use the Goal Seek tool to find out the marks required in one subject to achieve a total of 250.
24. Split the worksheet window and use Zoom to focus on a section of data.
25. Export the student data to PDF and CSV formats.

2nd Year
Semester-IV

Course-MAJOR	**Paper Code- COMSMAJ407**	**Credits-3**	**Lectures/Week-3**
Paper:	**Operating System**		

Prerequisite(s) and/or Note(s):

1. Basic understanding of computer organization and architecture
2. Familiarity with data structures and fundamental programming concepts
3. Note(s): This course builds foundational knowledge essential for advanced topics in system-level programming and distributed computing.

Course Objectives:

Knowledge Acquired:

1. Understanding the fundamental principles and functions of modern operating systems
2. Ability to analyse and solve common problems related to process management, memory, storage, and device control
3. Appreciation of the role operating systems play in bridging computer hardware and user-level applications
4. Foundational knowledge that supports further studies in areas such as systems programming, distributed systems, and cloud computing

Skills Gained:

1. Proficiency in writing and understanding system-level code
2. Ability to manage memory, schedule processes, and understand file systems and I/O operations
3. Skills in debugging and performance analysis of operating system components
4. Enhanced problem-solving abilities relevant to concurrent programming and resource management

Competency Developed:

1. Ability to design, implement, and optimize operating system components
2. Understanding of security, access control, and basic networking from an OS perspective
3. Competency in solving real-world system-level problems involving concurrency, deadlocks, and scheduling
4. Readiness for roles in software development, system programming, system administration, cybersecurity, and research in OS and distributed systems

Syllabus Overview

Unit 1: Introduction **8 Lectures**

Basic OS functions, resource abstraction, types of operating systems, multi-programming systems, batch systems, time sharing systems, multiprocessing system, operating systems for personal computers & workstations, process control & real time systems.

Unit 2: Operating System Organization **5 Lectures**

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

Unit 3: Process Management **12 Lectures**

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

Unit 4: Concurrency and Synchronization **10 Lectures**

Process synchronization, critical section, semaphores, methods for inter-process communication. Deadlocks: System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery from deadlock, banker's algorithm.

Unit 5: Memory Management **10 Lectures**

Physical and virtual address space; memory allocation strategies –fixed and variable partitions, demand paging, page replacement algorithms, allocation of frames, thrashing, segmentation, and virtual memory.

Suggested Readings:

1. A.Silberschatz, P.B.Galvin, G.Gagne, Operating Systems Concepts, 8thEdition, John Wiley Publications2008.
2. A.S.Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G.Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education1997.
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 Hill1992.
6. Operating Systems, A.K Sharma, University Press.

Course-MAJOR Paper:	**Paper Code- COMSMAJ407L Operating System (Lab)**	**Credits-1**	**Lab hours/Week-2**

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. WAP in C for implementation of Priority scheduling.
2. WAP in C for implementation of preemptive and non-preemptive scheduling.
3. WAP in C for implementation of Round Robin scheduling.
4. WAP in C for implementation of FCFS scheduling.
5. WAP in C for implementation of SJF scheduling.
6. WAP in C for implementation of producer – consumer problem using semaphores.
7. WAP in C for implementation of Banker's algorithm for deadlock avoidance.
8. WAP in C for implementation of deadlock avoidance.
9. WAP in C for implementation of memory allocation methods for fixed partitioning (First fit)
10. WAP in C to create a child process using fork (), display parent and child process id. Child process should display the message "Child process" and the parent process should display the message "Parent process".
11. WAP in C to accept n integers to be sorted. Main function creates child process using fork system call. Parent process sorts the integers in ascending order and waits for child process using wait system call. Child process sorts the integers in descending order.
12. WAP in C to implement the process system calls.

Course-MAJOR Paper:	Paper Code- COMSMAJ408 Data Structure	Credits-3	Lectures/Week-3

Prerequisite(s) and/or Note(s):

1. Basic knowledge of the C or C++ programming language
2. Familiarity with fundamental programming concepts such as loops, functions, and arrays
3. Note(s): This course provides the foundation for advanced studies in algorithms, software development, and system design

Course Objectives:

Knowledge Acquired:

1. Ability to design and implement fundamental data structures such as arrays, linked lists, stacks, queues, trees, and graphs using C or C++
2. Understanding of performance analysis and trade-offs in choosing appropriate data structures for various problems
3. Preparation for more advanced coursework in algorithms, databases, and software engineering

Skills Gained:

1. Development of algorithmic thinking and structured problem-solving approaches
2. Enhanced analytical and critical thinking skills to select and apply suitable data structures
3. Proficiency in abstraction and modelling of real-world problems using data structures
4. Strengthened coding skills through implementation and debugging of data structures in C or C++

Competency Developed:

1. Efficient use of computational resources through appropriate data structure design
2. Enhanced memory management skills and understanding of dynamic allocation
3. Debugging proficiency and adaptability in solving both theoretical and practical programming challenges
4. Readiness for roles in software development, system optimization, algorithm design, and technical interviews

Syllabus Overview

Unit 1:	Introduction and Arrays	4 Lectures

Definition of data structure, need, operations on data structures. Single and Multi-dimensional Arrays, Insertion, and deletion operations.

Unit 2:	Linked Lists	10 Lectures

Singly, Doubly and Circular Lists; Normal and Circular representation of Stack in Lists. Insertion and deletion operations on singly, doubly and circular lists.

Unit 3:	Stacks and Queues	12 Lectures

Implementing single/multiple stack/s in an Array, (Array and Linked representation), Push, Pop operations, Prefix, Infix and Postfix expressions, Applications of stack; Evaluation of theses expressions, conversion of these Expressions from one to another, Limitations of Array representation of stack. Array and Linked representation of Queue, De-queue, Priority Queues, Operations of queue.

Unit 4:	Trees	12 Lectures

Introduction to Tree as a data structure; General tree, Binary Trees, conversion of a general tree to a binary tree, traversal of a binary tree, Binary search tree, AVL tree.

Unit 5: Sorting and Searching **7 Lectures**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Bubble Sort, Insertion Sort, Comparison of Sorting Techniques

Suggested Readings

1. Seymour Lipschutz, "Data Structures with C", Tata McGraw Hill Education Pvt Ltd.
2. Yashvant Kanetkar, "Data structures through C", bpb
3. Deshpande Kakde, "C and Data structures", dreamtech
4. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++",2nd Edition, PHI, 2009.
5. Robert L.Kruse, "Data Structures and Program Design in C++", Pearson,1999.
6. D.S Malik, Data Structure using C++, 2nd Edition, Cengage Learning,2010.

Course-MAJOR Paper: **Paper Code- COMSMAJ408L Data Structure (Lab)** **Credits-1** **Lab hours/Week-2**

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems in C or C++:

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search.
2. WAP to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists.
4. Implement Doubly Linked List. Include functions for insertion, deletion and search of a number.
5. Implement Circular Linked List. Include functions for insertion, deletion and search of a number.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation.
8. Perform Queues operations using Circular Array implementation.
9. Create and perform different operations on Double-ended Queues using Linked List implementation and array implementation.
10. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
11. WAP to display Fibonacci series (i) using recursion,(ii)using iteration
12. WAP to reverse the order of the elements in the stack using additional stack.
13. WAP to reverse the order of the elements in the stack using additional Queue.
14. WAP to implement Diagonal Matrix using one-dimensional array.
15. WAP to implement Lower Triangular Matrix using one-dimensional array.
16. WAP to implement Upper Triangular Matrix using one-dimensional array.
17. WAP to implement Symmetric Matrix using one-dimensional array.