TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ2300 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III [√] Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: COMPUTER SYSTEM ARCHITECTURE

Course Code: (Will be provided by the University)

Course Credit:	Theoretical	3	Practical/Tutorial	1
Marks Allotted:	Theoretical	60	Practical/Tutorial	0
	Continuing Evaluation	10	Attendance	5

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES ☐ NO [√]

Is the course based on Activity? YES ☐ NO [√]

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSMAJ23003

Course Name: COMPUTER SYSTEM ARCHITECTURE

Brief Course Description:

The course computer system architecture course aims to describe a broad range of architectural designs and to contrast them, highlighting the design decisions they incorporate, and how these design decisions impact program performance.

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

Students who have not taken Digital Systems will need to do additional background reading on combinational circuits and assembler programming.

Course Objectives:

Knowledge acquired:

Understanding computer architecture and organization is essential for computer scientists, engineers, and programmers to develop efficient and high-performance systems. It forms the foundation for designing and optimizing hardware and software.

Skills gained:

Studying computer architecture and organization provides individuals with a range of skills that are valuable in the field of computer science and related disciplines. Studying computer architecture and organization equips individuals with a strong foundation in both theoretical and practical aspects of computer systems, preparing them for a wide range of roles in computer science and technology.

Competency Developed:

Studying computer architecture and organization leads to the development of various competencies that are valuable in the field of computer science and related industries. These competencies collectively prepare individuals for roles in areas such as computer architecture design, embedded systems development, system optimization, and research in advanced computing technologies. The knowledge and skills gained contribute to a well-rounded understanding of computer systems and their applications in various domains.

Detailed Syllabus

2nd Year: Semester 3

UCMSMAJ23003: Computer System Architecture [Credits: 3, Lectures: 45]

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.

Unit3: Central Processing Unit (12 lectures)
Stack organization, microprogrammed control. Microprogrammed instruction formats, addressing modes, instruction codes, machine language, assembly language, design of CPU, RISC, CISC architectures.

Unit4: Computer Arithmetic (10 lectures)
Addition and Subtraction, Multiplication algorithm (Boots), Division algorithm, Floating point arithmetic operations.

Unit5: Memory and Input-Output Organization (10 lectures)
Memory hierarchy, Main memory, Cache memory, Virtual memory. 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. 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.

UCMSMAJ23003: Computer System Architecture Tutorial [Credits: 1, Lecture Hours:15]

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

TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ23004 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III [√] Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: OPERATING SYSTEM

Course Code: (Will be provided by the University) Course Credit: Practical/Tutorial 3 Theoretical 1

Marks Allotted: Theoretical 40 Practical/Tutorial 20

Continuing Evaluation 10 Attendance 5

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSMAJ23004

Course Name: OPERATING SYSTEM

Brief Course Description:

This course introduces students to the core principles and components of operating systems, which serve as the fundamental software layer managing computer hardware and facilitating user applications. The course combines theoretical concepts with practical applications to ensure a well- rounded understanding of operating system functionality.

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

Basic understanding of computer organization and architecture, data structures, and programming concepts is recommended.

Course Objectives:

Knowledge acquired:

Upon completion of the course, students should be able to understand the fundamental principles of operating systems, analyze and solve common operating system problems, and appreciate the role of operating systems in supporting computer hardware and user applications. This knowledge is foundational for further studies in computer science and related fields.

Skills gained:

Studying operating systems provides students with a variety of skills that are valuable in the field of computer science and related disciplines. Here are some key skills gained after studying operating systems: problem solving, programming proficiency, memory management, file system knowledge, I/O system understanding. These skills collectively prepare students for roles in software development, systems programming, system administration, and related fields where a deep understanding of operating systems is crucial.

Competency Developed:

Studying operating systems helps develop a range of competencies that are essential for computer scientists, software engineers, and IT professionals. Here are key competencies developed through operating system study: problem solving, programming proficiency, memory management, file system knowledge, I/O system optimization, security knowledge, networking and distributed system.

These competencies collectively prepare individuals for roles in system and software development, system administration, cybersecurity, and other areas where a deep understanding of operating systems is crucial. They provide a solid foundation for addressing challenges in the design, implementation, and maintenance of computer systems.

Detailed Syllabus

2nd Year: Semester 3

UCMSMAJ23004: Operating System [Credits:3, Lectures:45]

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.

Unit2: 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.

Unit5: 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.

UCMSMAJ23004: Operating System Lab **[Credit: 1,Lab Hours:30]**

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.

TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science [√] Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSMAJ23005 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III ☐ Semester IV [√]

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: Data Communication and Networking

Course Code: (Will be provided by the University)

Course Credit: Marks Allotted:	Theoretical	3	Practical/Tutorial	1
	Theoretical	60	Practical/Tutorial	0
		10		5

Continuing Evaluation Attendance

Course Type (tick the correct alternatives):

Major Core	[√]	AEC	☐
Interdisciplinary/ DSE	☐	SEC	☐
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES ☐ NO [√]

Is the course based on Activity? YES ☐ NO [√]

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Prepared by

Course Code: UCMSMAJ23005

Course Name: DATA COMMUNICATION AND NETWORKING

Brief Course Description:

This course provides a comprehensive introduction to the principles, protocols, and technologies that underpin data communication and computer networks. It covers the fundamentals of networking, from the basics of data transmission to the design and management of modern computer networks. Emphasisis placed on understanding key concepts, protocols, and practical applications in the context of local andwide-area networks.

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

Basic understanding of computer systems, including hardware, software, and operating systems.

Course Objectives:

Knowledge acquired:

Upon completion of the course, students should be able to understand the fundamental principles of data communication and computer networks, design and implement simple networks, troubleshoot common networking issues, and comprehend the challenges and advancements in modern networking technologies. The knowledge gained prepares students for roles in network administration, network engineering, and related fields.

Skills gained:

Studying computer networks equips individuals with a diverse set of skills that are crucial in various technology-related roles. Here are key skills gained from learning about computer networks: Network design and architecture, protocols and standards, routing and switching, IP addressing, Network security, TCP/IP, OSI, Troubleshooting etc. These skills collectively prepare individuals for roles in network administration, network engineering, cybersecurity, and other areas where a deep understanding of computer networks is essential.

Competency Developed:

Studying networking and data communication develops a range of competencies that are valuable in the field of computer science, information technology, and related industries. Here are key competencies developed from gaining knowledge in networking and data communication: Network design, Protocols and standards, Networks services and application, Cloud computing and visualization, Network performance optimization etc. These competencies collectively prepare individuals for roles in network administration, network engineering, cyber security, and other areas where a deep understanding of networking and data communication is essential.

Prepared by

Detailed Syllabus

2nd Year: Semester 3

UCMSMAJ24005: DATA COMMUNICATION & NETWORKING [Credits: 3, Lectures 45]

Unit 1: Data Communication Fundamentals and Techniques (10 Lectures)
Introduction: Data Communications, Networks, Categories of network-LAN, MAN, WAN, Modes of Communication- Simplex, Half Duplex, Full Duplex, Network topologies, Internet History, layered network architecture-OSI model and TCP/IP protocol suite. Data and Signals- Analog and Digital signals, Parallel and Serial transmission, Transmission impairment- Attenuation, Distortion and Noise, Multiplexing techniques- FDM, TDM, WDM, Transmission media-Guided and Unguided.

Unit 2: Networks Switching Techniques and Access mechanisms (5 Lectures)
Switching- Circuit switching, Packet switching; Message switching, Connectionless and Connection-oriented datagram switching.

Unit 3: Data Link Layer Functions and Protocol (7 Lectures)
Error detection and error correction technique - Hamming Code, CRC, Checksum. Flow control mechanism- Sliding Window protocol, go-back-n ARQ, stop and wait ARQ,

Unit 4: Multiple Access Protocol and Networks (7 Lectures)
Multiple Access Protocols: Radom Access - ALOHA, CSMA, CSMA/CD, CSMA/CA, Controlled Access - Reservation, Polling, Token Passing, Channelization - FDMA, TDMA, CDMA. Network devices- repeaters,hubs, switches, bridges, router and gateways;

Unit 5: Networks Layer Functions and Protocols (5 Lectures)
Routing - Routing algorithms, classes of IP, IPv4 and lPv6 Addresses,

Unit 6: Transport Layer Functions and Protocols (6 Lectures)
Transport services-Congestion control, Connection establishment and release- three way handshaking, Network Security-Cryptography

Unit 7: Overview of Application layer protocol (5 Lectures)
Overview of DNS protocol, overview of WWW & HTTP protocol.

Suggested Readings:

1. Behrouz A. Forouzan, “Data Communications and Networking”, 5th Edition, McGraw HillEducation,2013.
2. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, 5th Edition, Prentice Hall,2011.
3. Larry L. Peterson and Bruce S. Davie, “Computer Networks A System Approach”, 5th Edition,MKP,2012.
4. James F. Kurose, Keith W. Ross, “Computer Networking, A Top-Down Approach”, 5thEdition,Pearson, 2012.
5. Dr. Rakesh Kumar Mandal: Computer Networks for Students, First Edition, SPD, 2018.

Prepared by

TEMPLATE FOR COURSE SYLLABUS FOR NEP IMPLEMENTATION

Discipline: Science ☐ Arts, Humanities & Social Science ☐

Commerce ☐ BBA ☐ BCA ☐

Subject Name: COMPUTER SCIENCE

Subject Code: UCMSSEC23003 (Will be provided by the University)

Semester: Semester I ☐ Semester II ☐ Semester III [√] Semester IV ☐

Semester V ☐ Semester VI ☐ Semester VII ☐ Semester VIII ☐

Course Name: DATA STRUCTURE THROUGH C OR C++

Course Code: (Will be provided by the University) Course Credit: Practical/Tutorial 3 Theoretical 1

Marks Allotted: Theoretical 40 Practical/Tutorial 20

Continuing Evaluation 10 Attendance 5

Course Type (tick the correct alternatives):

Major Core	☐	AEC	☐
Interdisciplinary/ DSE	☐	SEC	[√]
Minor / Generic Elective	☐	VAC	☐
Research Project/Dissertation	☐	Vocational	☐

Is the course focused on employability / entrepreneurship? YES [√] NO ☐

Is the course focused on imparting life skill? YES [√] NO ☐

Is the course based on Activity? YES [√] NO ☐

Remarks by Chairman, UG BOS, if any

UG BOS Meeting Reference Number: Date:

Course Code: UCMSSEC23003

Course Name: DATA STRUCTURE THROUGH C OR C++

Brief Course Description:

This course introduces students to the fundamental concepts of data structures and their implementation using the C programming language. Emphasizing theoretical understanding and practical application, the course covers the design, analysis, and implementation of essential data structures that form the building blocks of efficient algorithms.

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

Basic knowledge of C programming language and familiarity with fundamental programming concepts.

Course Objectives:

Knowledge acquired:

Upon completing the course, students should be able to design, implement, and analyze the performance of fundamental data structures using the C programming language. They will gain problem-solving skills and be prepared for more advanced coursework in algorithms and software development.

Skills gained:

Studying data structures provides individuals with a variety of skills that are valuable in computer science, software engineering, and other related fields. Here are key skills gained from learning about data structures: algorithmic thinking, problem solving, analytical skills, critical thinking, abstraction and modeling etc. These skills collectively contribute to a well-rounded foundation in computer science and software engineering, making individuals more proficient in developing efficient and scalable software solutions.

Competency Developed:

Studying data structures helps individuals develop a range of competencies that are essential in computer science, software engineering, and related fields. Here are key competencies developed from learning about data structures: efficient resource utilization, memory management, debugging proficiency, adaptability, collaboration and teamwork, documentation, continuous learning etc. These competencies collectively prepare individuals for roles in software development, algorithm design, system optimization, and other areas where a deep understanding of data structures is crucial. They contribute to the ability to create efficient, scalable, and robust software solutions.

Detailed Syllabus

2nd Year: Semester 3

UCMSSEC23003: Data structure through C or C++ Credits: 2, Lectures:30]

Unit 1: Arrays (2 Lectures)

Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

Unit 2: Stacks (5 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.

Unit 3: Linked Lists (7 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 4: Queues (3 Lectures)

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

Unit 5: Recursion (3 Lectures)

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion;

Unit 6: Trees (6 Lectures)

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

Unit 7: Searching and Sorting (4 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.

UCMSSEC23003: Data structure through C or C++ Lab [Credit:1, Lab Hours: 30]

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

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 calculate GCD of two number (i)with recursion(ii)without recursion
13. WAP to reverse the order of the elements in the stack using additional stack.
14. WAP to reverse the order of the elements in the stack using additional Queue.
15. WAP to implement Diagonal Matrix using one-dimensional array.
16. WAP to implement Lower Triangular Matrix using one-dimensional array.
17. WAP to implement Upper Triangular Matrix using one-dimensional array.
18. WAP to implement Symmetric Matrix using one-dimensional array.