NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL

Minor in

COMPUTER SCIENCE AND ENGINEERING

SCHEME OF INSTRUCTION AND SYLLABI

(for B.tech. students other than Computer Science and Engineering)

Effective from 2021-22

Department of Computer Science and Engineering

SCHEME OF INSTRUCTION

Minor Program – Computer Science and Engineering

S. No.	Course Code	Course Title	L	T	P	Credits	Sem.
1	CSM01	Data Structures	2	0	2	3	III
2	CSM02	Computer Organization and Architecture	3	0	0	3	IV
3	CSM03	Principles of Database Systems	2	0	2	3	V
4	CSM04	Operating System Principles	3	0	0	3	VI
5	CSM05	Computer and Communication Networks	3	0	0	3	VII
6	CSM06	Software Engineering Principles	3	0	0	3	VIII
		Total				**18**	

B.Tech. (Minors) – Computer Science and Engineering

DETAILED SYLLABUS

CSM01	DATA STRUCTURES	Credits 2-0-2: 3

Pre-requisites:

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand the concept of ADT, identify data structures suitable to solve problems
CO2	Develop and analyze algorithms for stacks, queues
CO3	Develop algorithms for binary trees and graphs
CO4	Implement sorting and searching algorithms
CO5	Implement symbol table using hashing techniques and multi-way search trees

Syllabus:

Introduction to Iterative and Recursive Algorithms

Abstract Data Types (ADTs), Implementation and Applications of Stacks, Operations and Applications of Queues, Array Implementation of Circular Queues, Implementation of Stacks using Queues, Implementation Queues using Stacks, Linked Lists, Search and Update Operations on Varieties of Linked Lists, Linked List Implementation of Stacks and Queues

Introduction to Trees, Implementation of Trees, Binary Trees, Tree Traversals with an Application, Binary Search Trees (BSTs), Query and Update Operations on BSTs, AVL Trees, Rotations, Search and Update Operations on Balanced BSTs, Splay Trees, B-trees, Trie, C-Trie

Hashing: Implementation of Dictionaries, Hash Function, Collisions in Hashing, Separate Chaining, Open Addressing, Analysis of Search Operations

Priority Queues: Priority Queue ADT, Binary Heap Implementation and Applications of Priority Queues, Disjoint Sets.

Sorting Algorithms: Stability and In Place Properties, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Lower Bound for Comparison Based Sorting Algorithms, Linear Sorting Algorithms: Counting Sort, Radix Sort, Bucket Sort

Graph Algorithms: Graphs and their Representations, Graph Traversal Techniques: Breadth First Search (BFS) and Depth First Search (DFS), Applications of BFS and DFS, Minimum Spanning Trees (MST), Prim's and Kruskal's algorithms for MST, Connected Components, Dijkstra's Algorithm for Single Source Shortest Paths, Biconnected Components.

Text Books / Reference Books / Online Resources:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "*Introduction to Algorithms"*, Second Edition, PHI, 2009.
2. Mark Allen Weiss, "*Data Structures and Algorithm Analysis in C++"*, Third Edition, Pearson Education, 2006
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "*Fundamentals of Computer Algorithms"*, Second Edition, Universities Press, 2011.
4. Michael T.Goodrich and Roberto Tamassia, "*Algorithm Design: Foundations, Analysis and Internet Examples"*, Second Edition, Wiley-India, 2006.

CSM02	COMPUTER ORGANIZATION AND ARCHITECTURE	Credits 3-0-0: 3

Pre-requisites:

Course Outcomes:

At the end of the course, the student will be able to

CO1	Identify functional units, bus structure and addressing modes
CO2	Design the hardwired and micro-programmed control units.
CO3	Identify memory hierarchy and performance.
CO4	Design Arithmetic Logic Unit.
CO5	Interface I/O devices

Syllabus:

Basic Structures of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multicomputers, Historical Perspective

Machine instructions and Programs: Numbers, Arithmetic Operations and Characters, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Assembly Language, Basic Input Output Operations, Stacks and Queues, Subroutines, Additional Instructions, Example Programs, Encoding of Machine Instructions

Registers and Addressing, IA-32 Instructions, IA-32 Assembly Language, Program Flow Control, Logic and Shift/Rotate Instructions, I/O Operations, Subroutines, Other Instructions, Program Examples

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces

The Memory System: Some Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage

Arithmetic: Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-Operand Multiplication, Fast Multiplication, Integer Division, Floating Point Numbers and Operations, Implementing Floating Point Operations

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Microprogrammed Control

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets, Data Path and Control Considerations, Super Scalar Operation, UltraSPARC 2 Example, Performance Consideration

Large Computer Systems: Forms of Parallel Processing, Array Processors, the Structure of General-Purpose Multiprocessors, Interconnection Networks

Text Books / Reference Books / Online Resources:

1. Carl Hamacher, "*Computer Organization"*, 5th Edition, McGraw Hill Publishers, 2002.
2. Wiiliam Stallings, "*Computer Organization and Architecture Designing for Performance"*, 8th Edition, Pearson Education, 2010
3. John P Hayes, "Computer Architecture and Organization", 3rd revised Ed., McGraw-Hill, 1998

CSM03	PRINCIPLES OF DATABASE SYSTEMS	Credits 2-0-2: 3

Pre-requisites: CSM01 – Data Structures

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand functional components of the DBMS.
CO2	Devise queries using Relational Algebra, Relational Calculus and SQL.
CO3	Design database schema.
CO4	Develop E-R model.
CO5	Evaluate and optimize queries.
CO6	Analyze transaction processing, concurrency control and recovery techniques.

Syllabus:

Introduction to DBMS: Historical perspective, File Versus a DBMS, Advantages of DBMS, Describing and storing data in DBMS, Architecture of a DBMS, Different Data Models;

Entity Relationship(ER) model: Features of ER model, conceptual design using ER model, design for large enterprises; Relational model–structure and operations, Integrity constraints over relations;

Query languages: Relational Algebra, Relational Calculus and SQL– Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL, triggers and Embedded SQL;

Database Design: Mapping ER model to Relational form; Functional Dependency–Closer of functional dependencies, closer of attributes, canonical cover and Properties of Decompositions; Normalization process – 1NF, 2NF, 3NF and BCNF; Multivalued dependency– Closer properties of Multivalued dependency and 4NF; Join dependency– PJNF, Decomposition Algorithms;

Transaction Management: ACID properties, transactions, schedules and concurrent execution of transactions; Concurrency control – lock based protocol, Serializability, recoverability, dealing with deadlocks and Concurrency control without locking;

Query Processing: Overview of Query Evaluation, operator evaluation; Algorithms for relational operations– Selection operation, General selection condition, Projection operation, Join operation, set operation and aggregate operation, Evaluation of relational operations; Query optimization: Alternative plans, functions of query optimizer, translating SQL queries into relational algebra, estimating the cost of a plan, relational algebra equivalences, and other approaches to query optimization;

Database Recovery: Failure classification, Recovery and atomicity, Log-based recovery shadow paging and Advanced Recovery Techniques:

Security and Authorization: Access control, direct access control and Mandatory access control, Role of DBA, Application development.

Text Books / Reference Books / Online Resources:

1. Elamsri, Navathe, Somayajulu and Gupta, "*Fundamentals of Database Systems"*, 6th Edition, Pearson Education, 2011.
2. Raghu Ramakrishnan, Johannes Gehrke, "*Database Management Systems"*, 3nd Edition, McGraw Hill, 2003.
3. Silberschatz, Korth and Sudharshan, "*Database System Concepts"*, 6th Edition, McGraw Hill, 2010.

CS304	OPERATING SYSTEM PRINCIPLES	Credits 3-0-0: 3

Pre-requisites: CSM02 – Computer Organization and Architecture, CSM01- Data Structures

Course Outcomes:

At the end of the course, the student will be able to

CO1	Distinguish functional architectures of operating systems and file systems
CO2	Develop algorithms for subsystem components
CO3	Design device drivers and multi-threading libraries for an OS
CO4	Develop application programs using UNIX system calls
CO5	Design and solve synchronization problems

Syllabus:

Introduction: Batch, iterative, time sharing, multiprocessor, distributed, cluster and real-time systems, UNIX system introduction and commands

Operating system structures: Computer system structure, Network structure, I/O Structure, Storage Structure, Dual mode operation, System components, Operating-System Services, System Calls, System Programs, System structure, Virtual Machines, System Design and Implementation, System Generation

Processes and Threads : Process Concept, Process Scheduling, Operations on Processes, Cooperating Processes, Interprocess Communication, Communication in Client – Server Systems, Multithreading Models, Threading Issues, Pthreads Basic Concepts,

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Algorithm Evaluation, Process Scheduling Models

Process Synchronization: Synchronization Background, the Critical-Section Problem, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Critical Regions, Monitors, OS Synchronization

Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock

Memory Management : Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations

File System: File Concept, Access Methods, Directory Structure, File-System Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File System, NFS

I/O Systems : Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure , Disk Scheduling , Disk Management, Swap-Space Management, RAID Structure , Disk Attachment, Stable-Storage Implementation, Tertiary-Storage Structure

Text Books / Reference Books / Online Resources:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, "*Operating System Principles"*, Wiley, 8/e
2. Richard Stevens, Stephen Rago, "*Advanced Programming in the UNIX Environment"*, Pearson Education, 2/e

CSM05	COMPUTER AND COMMUNICATIN NETWORKS	Credits 3-0-0: 3

Pre-requisites: CSM04-Operating Systems

Course Outcomes:

At the end of the course, the student will be able to

CO1	Understand OSI and TCP/IP models
CO2	Analyze MAC layer protocols and LAN technologies
CO3	Design applications using internet protocols
CO4	Implement routing and congestion control algorithms
CO5	Develop application layer protocols

Syllabus:

Introduction – network architecture - protocol implementation issues - network design. Reference models- The OSI Reference Model- the TCP/IP Model - A Comparison of the OSI and TCP/IP Models Datalink Layer-Ethernet, Token ring, wireless LANs-Issues with data link Protocols-Encoding framing and error detection and correction-sliding window Protocol-Medium access control

Network layer – network layer design issues - Routing algorithms - Congestion control algorithms – Internetworking - The network layer in the internet - Internet Protocol (IP) - Unicast, multicast, and inter domain routing

Transport layer - Elements of transport protocol - Congestion control – The Internet's Transmission Control Protocol (TCP) - Remote Procedure Call (RPC) – Implementation semantics of RPC – BSD sockets - client-server applications

Application layer - Domain name server – Simple Mail Transfer Protocol – File Transfer Protocol - World wide web - Hypertext transfer protocol -Presentation formatting and data compression-

Introduction to Network security - Web Services architectures for developing new application protocols.

Text Books / Reference Books / Online Resources:

1. Larry L Peterson, Bruce S Davis, "*Computer Networks"*, 5th Edition, Elsevier, 2012.
2. Andrew S. Tanenbaum, David J Wetherall, "*Computer Networks"*, 5th Edition, Pearson Edu, 2010.

CSM06	SOFTWARE ENGINEERING PRINICIPLES	Credits 3-0-0: 3

Pre-requisites: None

Course Outcomes:

At the end of the course, the student will be able to

CO1	Comprehend software development life cycle
CO2	Prepare SRS document for a project
CO3	Apply software design and development techniques
CO4	Identify verification and validation methods in a software engineering project
CO5	Implement testing methods for software
CO6	Analyze and Apply project management techniques for a case study

Syllabus:

The Software Problem - Cost, Schedule, and Quality, Scale and Change ; Software Processes- Process and Project , Component Software Processes; Software Development Process Models - Waterfall Model, Prototyping , Iterative Development , Rational Unified Process , Time boxing Model , Extreme Programming and Agile Processes , Using Process Models in a Project. Software Requirements Analysis and Specification - Value of a Good SRS , Requirement Process , Requirements Specification; Formal Specification- Formal Specification in the Software process, Sub-system interface specification, Behavioural Specification; Desirable Characteristics of an SRS - Components of an SRS, Structure of a Requirements Document; Functional Specification with Use Cases - Basics , Examples , Extensions, Developing Use Cases; Other Approaches for Analysis - Data Flow Diagrams , ER Diagrams , Validation; Software Architecture - Role of Software Architecture, Architecture Views - Component and Connector View - Components, Connectors, An Example. Architecture Styles for C&C View - Pipe and Filter, Shared-Data Style , Client-Server Style, Some Other Styles, Documenting Architecture Design - Evaluating Architectures; Design - Design Concepts - Coupling , Cohesion , The Open-Closed Principle . Function-Oriented Design (from Pressman) - Structure Charts, Structured Design Methodology, An Example. Object-Oriented Design (from Jalote)- OO Concepts, Unified Modeling Language (UML) , A Design Methodology , Examples; Detailed Design - Logic/Algorithm Design, State Modeling of Classes; Verification - Metrics - Complexity Metrics for Function-Oriented Design, Complexity Metrics for OO Design; Coding and Unit Testing -Programming Principles and Guidelines - Structured Programming , Information Hiding, Some Programming Practices, Coding Standards; Incrementally Developing Code - An Incremental Coding Process ,Test-Driven Development, Pair Programming; Managing Evolving Code - Source Code Control and Build, Refactoring; Unit Testing - Testing Procedural Units, Unit Testing of Classes; Code Inspection - Planning, Self-Review, Group Review Meeting; Metrics - Size Measures, Complexity Metrics; Testing - Testing Concepts - Error, Fault, and Failure, Test Case, Test Suite, and Test Harness , Psychology of Testing , Levels of Testing; Testing Process - Test Plan, Test Case Design, Test Case Execution; Black-Box Testing - Equivalence Class Partitioning, Boundary Value Analysis , Pairwise Testing, Special Cases, State-Based Testing; White-Box Testing - Control Flow-Based Criteria, Test Case Generation and Tool Support; Metrics - Coverage Analysis, Reliability, Defect Removal Efficiency.

Text Books / Reference Books / Online Resources:

1. Pankaj Jalote, "*Software Engineering Precise Approach*" , Wiley Publishers, 2012
2. Ian Sommerville, "*Software Engineering", 8/e* Pearson Publishers, 2012.
3. Roger Pressman, "*Software Engineering*", 5th edition, MCgrawHill, 2002