Computer Organization & Architecture

II B.Tech III Semester (18IT301)

Lectures	:	3 Periods/Week,Tutorial: 1	Continuous Internal Assessment	:	50
Credits	:	3	Semester End Examination (3 hours)	:	50

Prerequisites:

Digital Logic Design

Course Objectives:

Students will be able to

CO1: Conceptualize the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines, Machine Instructions.

CO2: Learn about various data transfer techniques in digital computer and the I/O interfaces.

CO3: Estimate the performance of various classes of Memories, build large memories using small memories for better performance and Relate to arithmetic for ALU implementation

CO4: Understand the basics of hardwired and micro-programmed control of the CPU, pipelined architectures , Hazards and Superscalar Operations.

Course Outcomes:

After the course the students are expected to be able to

CLO1: Explain the basics of organizational and architectural issues of a digital computer and Classify and compute the performance of machines, Machine Instructions.

CLO2: Describe various data transfer techniques in digital computer and the I/O interfaces.

CLO3: Analyze the performance of various classes of Memories, build large memories using small memories for better performance and analyze arithmetic for ALU implementation

CLO4: Describe the basics of hardwired and micro-programmed control of the CPU, pipelined architectures , Hazards and Superscalar Operations

UNIT - I (17 Periods)

Basic Structure Of Computers: Computer Types, Functional unit, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers. (8 Periods)

Machine Instructions And Programs: Numbers, Arithmetic Operations and Characters, Memory locations and addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations.(9 Periods)

UNIT - II (15 Periods)

Input/Output Organization: Interrupts, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces: PCI Bus, SCSI Bus, USB Bus. (15 Periods)

UNIT - III (17 Periods)

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. (9 Periods)

Arithmetic: Addition and Subtraction of Signed Numbers, Multiplication of Positive numbers, Signed operand multiplication, Fast multiplication, Integer Division, Floating point numbers and operations.(8 Periods)

UNIT - IV (15 Periods)

Basic Processing Unit: Some fundamental concepts, Execution of a complete instruction, Multiple Bus Organization, Hardwired control, Micro programmed control. (7 Periods)

Pipelining: Basic Concepts, Data Hazards, Instruction hazards, Influence on Instruction Sets, Data path and Control Considerations, Superscalar Operation, performance Considerations.(8 Periods)

TEXT BOOKS:

1. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, Fifth Edition, McGraw Hill.

REFERENCES:

1. Computer Architecture and Organization, John P. Hayes, Third Edition, McGraw Hill.
2. Computer Organization and Architecture, William Stallings, 6th Edition, Pearson/PHI.
3. Computer Systems Architecture, M. Morris Mano, Third Edition, Pearson/PHI

Data Structures

II B.Tech III Semester (18IT302)

Lectures	:	3 Periods/Week,Tutorial: 1	Continuous Internal Assessment	:	50
Credits	:	3	Semester End Examination (3 hours)	:	50

Prerequisites:

Problem Solving with Programming

Course Objectives:

Students will be able to

CO1: Understand and remember algorithms and its analysis procedure and Compute the complexity of various algorithms.

CO2: Introduce the concept of data structures through ADT including List, Stack, Queues, dynamic equivalence problem and smart union algorithm.

CO3: Understand the concept of Binary tree, binary search tree,AVL tree and their applications.

CO4: Learn Hashing,graph representations and traversal methods.

Course Outcomes:

After the course the students are expected to be able to

CLO1: Determine the time complexities of different algorithms, and implement ADTs of different types of linked lists and applications.

CLO2: Implement stack and queue ADTs using arrays and linked lists and their applications.

CLO3: Construct and implement different tree algorithms.

CLO4: Implement and analyze various hashing techniques and Graph traversal methods.

UNIT - I **(14 Periods)**

Algorithm Analysis: Mathematical Background, Model, what to Analyze, Running Time Calculations.
Lists: Abstract Data Types, The List ADT, Singly Linked List ADT, Doubly Linked List ADT, Circular Linked List ADT, Polynomial ADT: addition, multiplication operations.

UNIT - II **(14 Periods)**

Stacks and Queues: The Stack ADT and its applications such as Infix to Postfix expression conversions, Evaluation of Postfix expressions. The Queue ADT, Queue Application-Radix sort.
Basic Sorting Techniques: Bubble sort, Selection sort, Insertion sort, Shell sort

UNIT - III **(13 Periods)**

Trees: Preliminaries, Binary Trees, Expression trees, The Search Tree ADT, Binary Search Trees, Implementation. AVL Trees, Single Rotations, Double rotations, Implementations.

UNIT - IV **(13 Periods)**

Hashing: General Idea, Hash Function, Separate Chaining, Open Addressing.
Priority Queues (Heaps): Model, Simple implementations, Binary Heap, Heap Sort.

Disjoint Set ADT: Dynamic equivalence problem, Basic Data Structure, Smart Union Algorithms, Path Compression.

TEXT BOOKS:

1. Mark Allen Weiss, Data Structures and Algorithm Analysis in C, Second Edition, Pearson Education.

REFERENCES:

1. Y.Langsam, M.J.Augeustein and A.M.Tenenbaum, Data Structures Using C, Pearson Education Asia, 2004.
2. Richard F.Gilberg, Behrouz A. Forouzan, Data Structures A Pseudocode Approach with C, ThomsonBrooks / COLE, 1998.
3. Aho, J.E. Hopcroft and J.D. Ullman, Data Structures and Algorithms, Pearson Education Asia, 1983.

Discrete Mathematical Structures

II B.Tech III Semester (18IT303)

Lectures	:	3 Periods/Week,Tutorial: 1	Continuous Internal Assessment	:	50
Credits	:	3	Semester End Examination (3 hours)	:	50

Prerequisites:

NIL

Course Objectives:

Students will be able to

CO1: Understand set theory, relations and functions to read , understand Mathematical Induction and construct mathematical arguments.

CO2: Understand combinatorics,logic and mathematical reasoning to count or enumerate objects in systematic way.

CO3: Construct recurrence relations for elementary problems, and Apply generating functions to solve recurrence relations.

CO4: Understand the concept of lattices and graph theory.

Course Outcomes:

After the course the students are expected to be able to

CLO1: Verify the correctness of an argument using propositional and predicate logic and truth tables.

CLO2: Demonstrate the ability to solve problems using counting techniques and combinatorics in the context of discrete probability.

CLO3: Solve problems involving recurrence relations and generating functions.

CLO4: Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

UNIT - I (16 Periods)

Set Theory: Sets and subsets, Venn Diagrams, Operations on sets, laws of set theory, Power sets and products, Partition of sets, The principle of inclusion - Exclusion.
Relations: Definition, Types of relation, Composition of relations, Domain and range of a relation, Representation of Relations, Operations of relation, Special properties of a binary relation, Equivalence Relations and Partial Ordering Relations , POSET diagram and lattice, Paths and Closures.
Functions: Definition and types of functions, Composition, Inverse and Identity of functions.

UNIT - II (15 Periods)

Logic: Fundamentals of Logic, Logical Inferences, Methods of Proof of an implication, First order Logic & Other methods of proof, Rules of Inference for Quantified propositions, Mathematical Induction.
Elementary Combinatorics: Basics of Counting, Combinations and Permutations,Enumerating Combinations and Permutations with repetitions.

UNIT - III (15 Periods)

Recurrence relations: Generating functions of sequences, Calculating Coefficients of Generating Functions.

Solving recurrence relations by Substitution and generating functions. The methods of characteristic roots, solutions of inhomogeneous recurrence relations.

UNIT - IV (14 Periods)

Graphs: Basic concepts, Directed Graphs and Adjacency Matrices, Application: Topological Sorting.Isomorphisms and Subgraphs, Planar Graphs, Eulers Formula; Multigraphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four Color Problem.

TEXT BOOKS:

1. Joe L.Mott, Abraham Kandel & Theodore P.Baker, Discrete Mathematics for Computer Scientists & Mathematicians, PHI 2nd edition.
2. Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, , 5th Edition, Pearson Education. 2004.

REFERENCES:

1. Basavaraj S Anami and Venakanna S Madalli: Discrete Mathematics A Concept based approach, Universities Press, 2016.
2. Kenneth H. Rosen: Discrete Mathematics and its Applications, 6th Edition, McGraw Hill, 2007.
3. D.S. Malik and M.K. Sen: Discrete Mathematical Structures: Theory and Applications, Thomson, 2004.
4. Thomas Koshy: Discrete Mathematics with Applications, Elsevier, 2005, Reprint 2008.

Object Oriented Programming
II B.Tech - III Semester (18IT304)

Lectures	:	3 Periods/Week,Tutorial: 1	Continuous Internal Assessment	:	50
Credits	:	3	Semester End Examination (3 hours)	:	50

Prerequisites:

Problem Solving with Programming

Course Objectives:

CO1: This course provides an introduction to object oriented programming (OOP) features encapsulation, abstraction and inheritance using the Java programming language.

CO2: Understand the concept of Packages and Exception handling

CO3: Implement java applications using applets and events

CO4: Understand the AWT and Swing concepts in java

CO5: Be able to use the Java SDK environment to create, debug and run simple Java programs

Course Outcomes:

After the course the students are expected to be able to

CLO1: Understand fundamentals of java programming such as variables, conditional and iterative execution, methods, etc.

CLO2: Understand the principles of inheritance.

CLO3: Analyze the concept of exception handling mechanism.

CLO4: Design the java applications using Java applet and Event handling.

CLO5: Develop java applications using AWT and Swings.

UNIT - I **(15 Periods)**

The History and Evolution of Java,An Overview of Java,Data Types, Variables and Arrays,Operators,Control Statements,Introducing Classes,A Closer Look at Methods and Classes.

UNIT - II **(15 Periods)**

Inheritance
Packages and Interfaces
Strings: String Constructors, Program using 10 String methods StringBuffer class, Program using 10 StringBuffer methods Introducing StringBuilder class.
Type Wrappers Auto boxing/unboxing.
Collections: Collections Overview, Names of Collection Interfaces, Classes. Programs using Collection classes LinkedList <String>, ArrayList < String >

UNIT - III **(15 Periods)**

Exception Handling
Multithreaded Programming

I/O: I/O Basics, Reading Console Input, Writing Console Output, The PrintWriter class, Reading and Writing Files, Automatically Closing a File

UNIT - IV (15 Periods)

The Applet Class: Two Types of Applets, Applet Basics, Applet Architecture, An Applet Skeleton, Simple Applet Display Methods, Requesting Repainting, Using the Status Window, The HTML APPLET Tag, Passing Parameters to Applets, getDocumetBase(), getCodeBase(), Introducing Graphics and Color classes.

Event Handling:

AWT: basics, Program using AWT components Label, TextField, TextArea, Choice, Checkbox, CheckboxGroup, Button, Program using FlowLayout, GridLayout, BorderLayout. Advantages of Swing over AWT, Program using Swing Components JTable, JTree, JComboBox .

TEXT BOOKS:

1. Java The Complete Reference by Herbert Schildt , 9th Edition, , TMH Publishing Company Ltd, New Delhi.

REFERENCES:

1. Big Java, 2nd Edition, Cay Horstmann, John Wiley and Sons,PearsonEducation
2. Java How to Program (Early Objects), Tenth Edition, H.M.Dietel and P.J.Dietel, Pearson Education

Operating System

II B.Tech III Semester (18IT305)

Lectures	:	4 Periods/Week,Tutorial: 0	Continuous Internal Assessment	:	50
Credits	:	3	Semester End Examination (3 hours)	:	50

Prerequisites:

NIL

Course Objectives:

Students will be able to

CO1: Have a thorough understanding of the fundamentals of Operating Systems.

CO2 : Learn the mechanisms of OS to handle processes and threads and their communication

CO3 : Learn the mechanisms involved in memory management in contemporary OS

CO4 : Gain knowledge on Mutual exclusion algorithms, deadlock detection algorithms

CO5 : Know the components and management aspects of concurrency management

CO6 : Gain knowledge on file I/O operations and protection of various OS.

Course Outcomes:

After the course the students are expected to be able to

CLO1: Understand different structures, services of the operating system and the use of scheduling and operations on process.

CLO2: Understand the use of scheduling, operations on process, the process scheduling algorithms and synchronization concepts.

CLO3: Understand the concepts of deadlock, memory and virtual memory management techniques.

CLO4: Understand the concepts of File System, Input/output systems and system protection of various operating systems.

UNIT - I **(16 Periods)**

Introduction: What OSs Do, OS Structure, OS Operations.
Operating-System Structures: OS Services, System Calls, Types of System Calls, System Programs, OS Design and Implementation, OS Structure.
Processes: Process Concept, Process Scheduling, Operations on Processes, Inter- process Communication.
Threads: Overview, Multicore Programming,Multithreading Models.

UNIT - II **(15 Periods)**

CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms.
Process Synchronization: Background, The Critical-Section Problem, Petersons Solution, Synchronization Hardware,Mutex Locks, Semaphores, Classic problems of Synchronization, Monitors.

UNIT - III (15 Periods)

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

Main Memory: Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of Page Table.

Virtual-Memory : Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Other Considerations.

UNIT - IV (14 Periods)

File System Interface: File concept, Access Methods, Directory and Disk Structure,
File Sharing- Multiple Users, Remote File Systems, The Client-Server Model, Distributed Information Systems.

I/O Systems: Overview, Application I/O Interface.

Protection: Goals of Protection, Principles of Protection, Domain of Protection- Domain Structure, Access Matrix, Implementation of Access Matrix.

TEXT BOOKS:

1. Silberschatz & Galvin, Operating System Concepts, 9th edition, John Wiley & Sons (Asia) Pvt.Ltd.,.

REFERENCES:

1. William Stallings, Operating Systems Internals and Design Principles, 5/e, Pearson.
2. Charles Crowley, Operating Systems: A Design-Oriented Approach, Tata McGraw Hill Co., 1998 edition.
3. Andrew S.Tanenbaum, Modern Operating Systems, 2nd edition, 1995, PHI

Technical English

II B.Tech III Semester (18EL002)

Lectures	:	3 Periods/Week,Tutorial: 0	Continuous Internal Assessment	:	50
Credits	:	2	Semester End Examination (3 hours)	:	50

Prerequisites:

NIL

Course Objectives:

Students will be able to

CO1: at enhancing the vocabulary competency of the students

CO2: to introduce corrective measures to eliminate grammatical errors in speaking and writing

CO3: to learn writing as a process, including various invention heuristics (such as brainstorming), gathering evidence, considering audience, drafting, revising, editing, and proofreading

CO4: use grammatical, stylistic, and mechanical formats and conventions appropriate for a variety of purposes

CO5: produce coherent, organized, readable prose for a variety of rhetorical situations

Course Outcomes:

After the course the students are expected to be able to

CLO1: build academic vocabulary to enrich their writing skills

CLO2: make use of contextual clues to infer meanings of unfamiliar words from context

CLO3: produce accurate grammatical sentences

CLO4: Participate actively in writing activities (individually and in collaboration) that model effective technical communication in the workplace.

CLO5: understand how to apply technical information and knowledge in practical documents for a variety of purposes.

CLO6: practice the unique qualities of professional writing style that includes sentence conciseness, readability, clarity, accuracy, honesty, avoiding wordiness or ambiguity, previewing, using direct order organization, objectivity, unbiased analyzing, summarizing, coherence and transitional devices.

CLO7: use grammatical, stylistic, and mechanical formats and conventions appropriate to various audiences and disciplines

CLO8: collect, analyze, document, and report clearly, concisely, logically, and ethically; understand the standards for legitimate interpretations of data within technical communities.

UNIT - I

Vocabulary Development: Familiarising Idioms & Phrases
Grammar for Academic Writing: Making Requests
Language Development:Using Transition & Link words
Technical Writing:Letter Writing & Email Writing

UNIT - II

Vocabulary Development: Analogous words
Grammar for Academic Writing: Tenses: SimplePast/Present Perfect, The Future: Predicting & Proposing
Language Development: Cloze tests
Technical Writing: Technical Reports

UNIT - III

Vocabulary Development: Abbreviations & Acronyms
Grammar for Academic Writing: Describing(People/Things/Circumstances) : Adjectival & Adverbial groups
Language Development: Transcoding (Channel convertion from chart to text)
Technical Writing: Circular, Memos, Minutes of Meeting

UNIT - IV

Vocabulary Development: Corporate vocabulary
Grammar for Academic Writing: Inversions & Emphasis
Language Development: Reading Comprehension
Technical Writing: Resume Preparation

REFERENCES:

1. Communication Skills, Sanjay Kumar & Pushpa Latha. Oxford University Press:2011.
2. Technical Communciation Principles and Practice. Oxfor University Press:2014.
3. Objective English(Third Edition), Edgar Thorpe & Showick. Pearson Education:2009.
4. English Grammar: A University Course (Second Edition), Angela Downing & Philip Locke, Routledge Taylor & Francis Group: 2016.

Data Structures Lab

II B.Tech III Semester (18ITL31)

Practical	:	3 Periods/Week	Continuous Internal Assessment	:	50
Credits	:	1	Semester End Lab Examination (3 hours)	:	50

List Of Experiments

1. Write a program to perform the following operations on Array List
   (a) Creation
   (b) Insertion
   (c) Deletion
   (d) Search
   (e) Display
2. Write a program that reads two lists of elements, prints them, reverses them,Prints the reverse list, sort the lists, print the sorted lists, merges the list, prints merge list using array list.
3. Write a program to perform the following operations on Single Linked List.
   (a) Creation
   (b) Insertion
   (c) Deletion
   (d) Search
   (e) Display
4. Write a program to perform the following operations on Doubly Linked List.
   (a) Creation
   (b) Insertion
   (c) Deletion
   (d) Search
   (e) Display
5. Write a program to perform addition and multiplication of two polynomials using single Linked List.
6. Write a program to implement the following using stack.
   (a) infix to postfix conversion
   (b) postfix evaluation
7. Write a program that performs Radix sort on a given set of elements using queue.
8. Write a program to read n numbers in an array. Redisplay the arraylist with elements being sorted in ascending order using the following Techniques
   (a) Bubble sort
   (b) Selection sort
   (c) Insertion sort
   (d) Shell sort
9. Write a program to demonstrate Binary Expression tree.
10. Write a program to perform Binary Search tree operations and traversals.
11. Write a program to implement AVL tree that interactively allows

(a) Insertion
(b) Deletion
(c) Find min
(d) Find max

12. Write a program to read n numbers in an array. Redisplay the arraylist with elements being sorted in ascending order using Heap Sort.

13. Write a program to find an element using Open Addressing.

14. Write a program to perform the following operations on Disjoint Set.

    (a) Make-Set
    (b) Find-Set
    (c) Union

Object Oriented Programming Lab

II B.Tech III Semester (18ITL32)

Practical	:	3 Periods/Week	Continuous Internal Assessment	:	50
Credits	:	1	Semester End Lab Examination (3 hours)	:	50

List Of Experiments

1. Write a java program to demonstrate static member, static method and static block.
2. Write a java program to demonstrate method overloading and method overriding.
3. Write a java program to implement multiple inheritance.
4. Write a java program to demonstrate final, blank final, final methods, and final classes.
5. Write a program to demonstrate packages.
6. Write a java program to demonstrate interfaces.
7. Write a java program to create user defined exception class and test this class.
8. Write a java program to demonstrate synchronous keyword.
9. Write am applet program to demonstrate Graphics class.
10. Write GUI application which uses awt components like label, button, text field, text area, choice, checkbox, checkbox group.
11. Write a program to demonstrate MouseListener, MouseMotionListener, KeyboardListener, ActionListener, ItemListener.
12. Develop swing application which uses JTree, Jtable, JComboBox.

Operating Systems Lab

II B.Tech III Semester (18ITL33)

Practical	:	3 Periods/Week	Continuous Internal Assessment	:	50
Credits	:	1	Semester End Lab Examination (3 hours)	:	50

List Of Experiments

1. Write a program to simulate FCFS Scheduling Algorithm to find turnaround time and waiting time.
2. Write a program to simulate SJF-non pre-emtive Scheduling Algorithm to find turnaround time and waiting time.
3. Write a program to simulate Priority-non pre-emtive Scheduling Algorithm to find turnaround time and waiting time.
4. Write a program to simulate Round Robin Scheduling Algorithm to find turnaround time and waiting time.
5. Write a Program to simulate the concept of Dining-Philosophers problem.
6. Write a program to simulate producer-consumer problem using semaphores.
7. Write a program to simulate Bankers Algorithm for deadlock avoidance.
8. Write a program to simulate Deadlock Detection algorithm.
9. Write a program to simulate FIFO page replacement algorithms.
10. Write a program to simulate LRU page replacement algorithms.
11. Write a program to simulate OPR page replacement algorithms .
12. Write a program to simulate the following Contiguous Memory Allocation techniques:
    (a) worst-fit
    (b) best-fit
    (c) first-fit
13. Implement Paging technique of memory management.