MKSSS's Cummins College of Engineering for Women, Pune
(An Autonomous Institute Affiliated to SavitribaiPhule Pune University)

Autonomous Program Structure of
Second Year B. Tech. Third Semester
(Information Technology)
Academic Year: 2023-2024 Onwards

Course Code	Course Title	Teaching Scheme Hours /Week			Examination Scheme				Total Marks	Credit
		Lecture	Tutorial	Practical	In Sem	End Sem	Oral	Practical		
20IT 301	Data Structures	3	0	0	50	50	0	0	100	3
20IT 302	Discrete Mathematics	3	1	0	50	50	0	0	100	4
20IT 303	Digital Electronics and Computer Architecture	3	0	0	50	50	0	0	100	3
20IT 304	Network Fundamentals	3	1	0	50	50	0	0	100	4
20HS 301	Universal Human Values - 2	2	1	0	50	50	0	0	100	3
20IT 301L	Data Structures Lab	0	0	4	25	0	0	25	50	2
20IT 303L	Digital Electronics and Computer Architecture Lab	0	0	2	25	0	25	0	50	1
20IT 305L	Object Oriented Analysis and Design Lab	0	0	4	25	0	0	25	50	2
20AC 301	Audit Course	0	0	1	0	0	0	0	0	No Credit
	Total	**14**	**3**	**11**	**325**	**250**	**25**	**50**	**650**	**22**
	Grand Total	**28**			**650**		**650**			

APPROVED BY
Secretary Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

APPROVED BY
Chairman Academic Council
MKSSS's Cummins College of Engineering
For Women, Pune-411052

20IT 301 Data Structures

Teaching scheme:
Lectures:3 hours/week
Tutorial : --

Examination Scheme:
In-Semester : 50 Marks
End-Semester : 50 Marks
Credit : 3

Prerequisites: Fundamentals of Programming Language

Course Objectives:

Familiarize students with

1. Linear & non-linear data structures
2. Iterative & recursive function implementations
3. Symbol table & hashing techniques
4. Algorithm analysis using time & space complexity

Course Outcomes:

Students should be able to

1. Apply appropriate programming language constructs to develop logical steps for solving a real world problem.
2. Analyze algorithmic complexities of an algorithm.
3. Select appropriate linear & nonlinear data structure to solve a given problem.
4. Apply different hashing techniques for efficient data storage and retrieval.

Unit – I **Introduction to Data Structures** **5 Hours**

Concept of problem solving, Revision: Concept of data types, operators, control structures, functions, arrays and collections.

Introduction to Data Structures: Types of data structures, Abstract Data Types

Unit – II **Introduction to Analysis of Algorithms** **7 Hours**

Concept of algorithm, characteristics of algorithms, pseudo code. Analysis of algorithm: frequency count and its importance in analysis of an algorithm, Time complexity & Space complexity of an algorithm, Best, Worst and Average case analysis of algorithm.

Sorting algorithm : Bubble sort, Searching algorithm : Linear search, Binary search

Unit – III **Linked List** **7 Hours**

Concept of linked organization, singly linked list, doubly linked list, circular linked list. Linked list as an ADT. Representation and manipulations of polynomials using linked lists, comparison of a sequential and linked memory organization, concept of Generalized Linked List, polynomial representation using GLL.

Unit – IV **Stack & Queue** **8 Hours**

Concept of stack, stack as ADT, Implementation of stack using array and linked organization, stack as data structure, use of stack- Recursion, expression conversion & evaluation
Concept of queues as ADT, Implementation using array and linked organization. Priority queue.

Unit – V **Trees** **8 Hours**

Concept of non-linear data structure, Trees and binary trees-concept and terminology. Binary tree as an ADT., Expression tree. Conversion of general tree to binary tree. Recursive and non-recursive algorithms for binary tree traversals, Binary search trees, Threaded binary tree.

Unit – VI **Hash Tables** **7 Hours**

Symbol Table: Symbol Table, Huffman's algorithm
Hash table: hashing function, collision resolution techniques- linear probing, rehashing, chaining without replacement and chaining with replacement.

Text Books

1. R. Gilberg, B. Forouzan, "Data Structures: A pseudo code approach with C", Cenage Learning
2. Cay S. Horstmann, "Big Java: Early Objects", John Wiley

Reference Books

1. E. Horowitz, S. Sahani, "Fundamentals of Data Structures", Computer Science Press
2. Alfred Aho, John Hopcroft, Jeffrey Ullman, "Data Structures & Algorithms", Pearson Publication
3. Robert Lafore, "Data Structures and Algorithms in Java", Sams Publication

20IT 302 Discrete Mathematics

Teaching Scheme:
Lectures: 3 hours/week
Tutorial: 1 hours/week

Examination Scheme:

In-Semester: 50 marks
End-Semester: 50 marks
Credits: 4

Prerequisites: Basic Mathematics

Course Objectives:
Familiarize students with

1. Sets and propositions to gain knowledge to formulate and solve problems.
2. The concept of relations and functions.
3. Graph and Tree terminologies and models to be applied in real life problems.
4. The basics of algebraic structures and its applications along with number theory.

Course Outcomes:
Students should be able to

1. Analyze logical propositions using principles of logic for solving complex reasoning problems.
2. Prove mathematical theorems using structured reasoning for justification.
3. Apply algebraic techniques on discrete mathematics and algorithms for real world problems.
4. Evaluate the combinatorial problems using systematic approaches for optimizing solutions.

Unit I: Sets and Propositional logic 7 Hours

Sets: Sets, Combinations of Sets, Venn diagram, Finite and Infinite Sets, Countable Sets, Multisets, Principle of Inclusion and Exclusion, Mathematical Induction.

Propositions: Propositions, Logical Connectives, Conditional and Bi-conditional Propositions, Logical Equivalence, Validity of Arguments by using Truth Tables, Predicates and Quantifiers, Normal forms. Applications of Sets and Propositions

Unit II: Relations and Functions 7 Hours

Relations: Relations and their properties, n-ary relations and their applications, representing relations, closures of relations, equivalence relations, partial orderings, Lattices, Chains and AntiChains.

Functions: Functions: Functions, Composition of Functions, Invertible Functions, and Pigeonhole Principle.

Unit III: Graphs 7 Hours

Graphs and Graph Models , Graph Terminology and Special Types of graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths , Shortest-Path Problems, Planar Graphs, Graph Coloring.

Unit IV: Trees **7 Hours**

Introduction to Trees, Applications of Trees: Introduction, Binary search trees, Prefix codes, Tree Traversal: Preorder, in-order and post-order traversals , Minimum Spanning Trees: Introduction, Prim's algorithm, Kruskal's algorithm.

Unit V: Groups and Rings **7 Hours**

Group Theory: Groups, subgroups, generators and evaluation of powers, cosets and lagrange's theorem, permutation groups and burnside's theorem, isomorphism and automorphisms, homomorphisms and normal subgroups, rings, integral, domain and fields.

Unit VI: Counting **7 Hours**

Combinatorics: Rules of Sum and Product, Permutations, Combinations. Discrete Probability: Discrete Probability, Conditional Probability, Bayes Theorem, Information and Mutual Information, Applications of Combinatorics and Discrete Probability.

Textbooks:

1. C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", 4th Edition, McGraw-Hill

2. Kenneth H. Rosen, "Discrete Mathematics and its Applications", & 7th edition, McGraw-Hill

Reference Books:

1. Bernard Kolman, Robert C. Busby, Sharon Cutler Ross, "Discrete mathematical structures", 6th edition, Prentice Hall of India

2. Edgar G. Goodaire, Michael M. Parmenter, "Discrete Mathematics with Graph Theory", 3rd Edition, Pearson Education

3. Tremblay J. S., "Discrete mathematical structures with application", 3rdEdition, Tata McGraw Hill 4. Lipschutz Seymour, "Discrete mathematics", 4th Edition, Tata McGraw-Hill

20IT 303 Digital Electronics and Computer Architecture

Teaching Scheme:
Lectures: 3 hours/week

Examination Scheme:
In-Semester: 50 marks
End-Semester: 50 marks
Credits: 3

Prerequisites: Basic Electrical and Electronics Engineering

Course Objectives:
Familiarize students with

1. Basic digital design techniques
2. Design and implement of combinational and sequential logic circuits
3. Fundamental working of Computer Systems
4. Architecture and features of a microprocessor

Course Outcomes:
Students should be able to

1. Solve problems using binary arithmetic and codes
2. Design combinational logic circuits using reduction techniques
3. Design Sequential logic circuits
4. Explain Architectural details of a microprocessor
5. Explain Memory management and Interrupts of a microprocessor

Unit – I: Number System **7 Hours**

Introduction to Boolean algebra and Number Systems. Signed Binary number representation and Arithmetic: Signed & True Magnitude, 1's complement, 2's complement representation and arithmetic.

Codes: BCD, Excess-3, Gray code, Binary Code and their conversion.

Logic minimization: Representation of truth-table, Simplification of logical functions, Minimization of SOP and POS forms, don't care Conditions, K-Maps.

Unit – II: Combinational Logic Design **7 Hours**

CLC design using SSI chips – Code converters, Adder, Subtractor, n bit Binary adder, Introduction to MSI functions & chips - Multiplexers, Decoder / Demultiplexer, Encoder, Binary adder.

CLC design using MSI chips – BCD & Excess 3 Adder and Subtractor

Unit – III: Sequential Logic Design **7 Hours**

Introduction to sequential circuits. Difference between combinational circuits and sequential circuits, memory element – latch.

Flip- Flops: Design, truth table, excitation table of SR, JK, D, T flip flops. Study of flip flops with asynchronous and synchronous Preset & Clear, conversion from one type to another type of flip flop.

Application of flip-flops – Counters- asynchronous, synchronous and modulo counters. Study of modulus n counter ICs & their applications to implement mod counters.

Unit – IV: Sequential Logic Design **7 Hours**

Registers- Buffer register, shift register types - SISO, SIPO, PISO & PIPO, applications of shift registers - ring counter, twisted ring counter, Sequence generators using counters & shift register, Sequence Detectors using Mealy and Moore model

Unit – V: Processor Architecture **7 Hours**

Introduction to 8086: internal Architecture, generation of physical address, minimum/maximum mode, study of 8086 supporting chips 8288(Latch), 8284(Clock Generator), 8286(trans receiver), 8288(Bus controller), Timing diagram read Write machine cycle.

Unit – VI: Assembly Language Programming and Interrupt Structure **7 Hours**

Introduction to assembly language programming- Instruction Descriptions, Assembler Directives addressing modes, Examples of programming, Procedures and Macros. Interrupt Structure, Interrupt service routine, Interrupt vector table, hardware and software interrupts, INTR, NMI, Interrupt response, Execution of ISR, Priorities of interrupt

Text Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4
2. Douglas Hall, "Microprocessors and Interfacing, Programming and Hardware", McGraw-Hill, ISBN: 0–07–100462–9Book 2

Reference Books:

1. Malvino Leach, "Digital Principles and Applications", Tata Mc-Graw Hill, (5th Edition)
2. 8086 Intel Manual

Web References:

1. NPTEL Series: Digital Systems Design, Prof. Roychoudhary, IIT Kharagpur
2. NPTEL NOC:Microprocessors and Microcontroller, Prof. Santanu Chattopadhyay, IIT Kharagpur
3. NPTEL NOC:Microprocessors and Interfacing, Prof. Shaik Rafi Ahamed, IIT Guwahati

20IT304 Network Fundamentals

Teaching Scheme:
Lectures: 3hours/week
Tutorial: 1hour/week

Examination Scheme:
In-Semester: 50 Marks
End-Semester: 50 Marks
Credits: 4

Course Objectives:

1. To interpret OSI and TCP/IP models.
2. To differentiate between media access schemes.
3. To analyze error detection and control mechanisms.
4. To demonstrate IP addressing.

Course Outcomes:

1. Differentiate between OSI and TCP/IP models.
2. Analyze the different types of network delays in packet-switched networks
3. Differentiate between data link layer services and multiple access techniques
4. Design the IP addressing scheme for a small network.

Unit – I: Introduction **7 Hours**

The Architecture of the Internet, Layering and encapsulation, LAN, WAN, MAN, Networking Devices, Network Topologies Point to Point, Point to Multipoint Topologies, The platform for communications, Protocols, OSI Model

Unit – II: Communicating over the Network **7 Hours**

TCP/IP Model, Protocol Data Units and Encapsulation, The Internet evolution, Comparison between OSI and TCP/IP Model, Network Addressing and Virtual circuit networks

Unit – III: Network Layer **7 Hours**

IP Addressing, Network Layer Protocol, IPv4, Subnetting, Static Routing, Dynamic Routing and IPv6, Network delay

Unit – IV: Ethernet **7 Hours**

Ethernet Basics, Collision Domain, Broadcast Domain, CSMA/CD, Half and Full-Duplex Ethernet, Ethernet at the Data Link Layer, Ethernet Addressing, Ethernet Frames, Channel Bonding, Ethernet at the Physical Layer.

Unit – V: Physical Layer **7 Hours**

Transmission media (Guided and Unguided), Performance (Bandwidth, Throughput, latency and BDP), Digital Modulation and Multiplexing.

Unit – VI: Data Link Layer **7 Hours**

Data Link Layer Design Issues, Sliding Window Protocol, Error Correction and Detection, Medium Access Control Sublayer, Carrier Sense Multiple Access Protocols.

Text Books:

1. Mark A. Dye, Rick McDonald, Antoon W. Rufi, **'Network Fundamentals,** Cisco Press (2008)
2. Behrouz Forozoun , 'Data Communications and networking' McGraw Hill Education (5E)

Reference Books:

1. Andrew S. Tennabaum, David J. Weatherall **"Computer Networks",** *Pearson* (5th edition), (2011)
2. Jim Kurose, Keith Ross "**Computer Networking: A Top Down Approach**", Pearson (7th edition) (2016)

20IT 301L Data Structures Laboratory

Teaching Scheme:
Practical : 4 hours/week

Examination Scheme:
In-Semester : 25 Marks
Practical : 25 Marks
Credit : 2

Prerequisites: Fundamentals of Programming Language

Course Objectives:
Familiarize students with
1. Linear data structures to solve real world problems
2. Non-Linear data structures to solve real world problems
3. Hashing techniques
4. Debugging of different codes & detect logical errors

Course Outcomes:
Students should be able to
1. Make use of linear data structures to solve a given problem
2. Make use of nonlinear data structures to solve a given problem
3. Utilize appropriate hashing techniques to solve a given problem
4. Validate program functionality using multiple inputs.

Suggested List of Laboratory Assignments
The laboratory assignments are designed in a set of group A, B and C such that students will be able to design and implement solution for a given problem. The laboratory assignments of group A, B and C are to be implemented using JAVA object-oriented programming language. Group A assignments are mandatory. Group B assignment is mandatory & may be performed in a group of 2 to 4 students. Group C assignments are extra assignments.

Group A

1. Operations on set
   a) Use Java Collection Framework - Set - addAll (union), retainAll(intersection), removeAll(symmetric difference)
   b) Without using Java Collection Framework - union, intersection, difference, symmetric difference

2. Operations on linked list
   a) Use Java Collection Framework - LinkedList - add, add(index, element), addFirst, addLast, clear, get, getFirst, getLast, remove, removeFirst, removeLast
   b) SLL - Without using Java Collection Framework - add, add(index, element), addFirst, addLast, clear, get, getFirst, getLast, remove, removeFirst, removeLast, reverse
   c) DLL - Without using Java Collection Framework - add, add(index, element), addFirst, addLast, clear, get, getFirst, getLast, remove, removeFirst, removeLast, reverse

3. Operation on stack
   a) Use Java Collection Framework - Stack - push, pop, peek, empty
   b) Without using Java Collection Framework - Stack - push, pop, peek, empty - implement stack as ADT
   c) Using the stack ADT - implement expression conversion algorithms - infix_to_postfix, infix_to_prefix, postfix_to_infix, prefix_to_infix

4. Operations on queue
   a) Use Java Collection Framework - LinkedList - add, remove, poll, peek, element
   b) Use Java Collection Framework - PriorityQueue - add, remove, poll, peek, element
   c) Without using Java Collection Framework - add, remove, peek - implement queue as ADT
   d) Using the queue ADT - implement priority queue - patient treatment, vehicle traffic management

5. Operations on binary search tree
   a) Use Java Collection Framework - TreeMap
   b) Implement binary search tree and perform the following operations - Insert, Delete, Search, Display, mirror image, display level-wise

6. Construct an expression tree from a postfix expression and perform recursive and non-recursive traversals – inorder, preorder and postorder

7. Operation on hash table
   a) Use Java Collection Framework - HashMap
   b) Implementation of Hash table using array and handle collisions using Linear probing, without replacement without chaining, without replacement with chaining, with replacement without chaining, with replacement with chaining, chaining using linked list

Group B (Any 1)

Design a mini project which uses the different data structures with or without Java Collection Framework. Few suggested assignments:

1. Library management system
2. Blood bank management system
3. Student Attendance management system

Group C (Extra)

1. Implement Huffman coding
2. Implement Heap sort
3. Implement optimal binary search tree
4. Implement threaded binary tree

Text Books

1. R. Gilberg, B. Forouzan, "Data Structures: A pseudocode approach with C", Cenage Learning
2. Cay S. Horstmann, "Big Java: Early Objects", John Wiley

Reference Books

1. E. Horowitz, S. Sahani, "Fundamentals of Data Structures", Computer Science Press
2. Robert Lafore, "Data Structures and Algorithms in Java", Sams Publication
3. William J. Collins , "Data Structures and the Java Collections Framework", John Wiley

20IT 303L Digital Electronics and Computer Architecture Laboratory

Teaching Scheme:
Practical: 2 hours/week

Examination Scheme:
In-Semester: 25 marks
Oral: 25 marks
Credits: 1

Prerequisites: Basic Electrical and Electronics Engineering

Course Objectives:
Familiarize students with

1. Basic digital Integrated Circuits (IC)
2. Analyze and Test basic digital circuits
3. Writing Assembly Language Program
4. Executing Assembly Language Program

Course Outcomes:

Students should be able to

1. Use appropriate IC's for designing digital circuits
2. Implement digital circuits for various inputs
3. Use the processor instructions to write basic assembly language programs
4. Use assembly language for modular programming

Suggested List of Laboratory Assignments

1. Design (truth table, K-map) and implementation of 4 bit BCD & Excess 3 Adder using IC7483.
2. Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. (Verification, cascading & logic function implementation)
3. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down synchronous Counter using master slave JK flip-flop IC 7476
4. Design and implementation of Mod 'n' counter with IC7490
5. Design (State Diagram, State Table, K Map) and implementation of Sequence Generator.
6. Write Assembly Language Program for addition and subtraction of two 8 bit numbers
7. Write Assembly Language Program for converting two digit BCD number to its equivalent HEX and vice- versa.
8. Write ALP to perform string operations like
    1. Find length of string
    2. Compare two strings

3. Concatenation of two strings
4. Reverse string

Text Books:

1. R.P. Jain, "Modern Digital Electronics", 3rd Edition, Tata McGraw-Hill, ISBN: 0–07–049492–4
2. Douglas Hall, "Microprocessors and Interfacing, Programming and Hardware", McGraw-Hill, ISBN: 0–07–100462–9Book 2

Reference Books:

1. Malvino Leach, "Digital Principles and Applications", Tata Mc-Graw Hill,(5th Edition)
2. 8086 Intel Manual

Web References:

1. NPTEL Series: Digital Systems Design, Prof. Roychoudhary, IIT Kharagpur
2. NPTEL NOC:Microprocessors and Microcontroller, Prof. Santanu Chattopadhyay, IIT Kharagpur
3. NPTEL NOC:Microprocessors and Interfacing, Prof. Shaik Rafi Ahamed, IIT Guwahati

20IT 305L Object Oriented Analysis and Design Laboratory

Teaching Scheme:
Practical: 4 hours/week

Examination Scheme:
In-Semester: 25 Marks
Practical: 25 Marks
Credits: 2

Prerequisites: Fundamentals of Programming Language 2

Course Objectives:

Familiarize students with

1. Introduction of UML 2.0 diagrams
2. Class modeling of a system
3. State modeling of a system
4. Interaction modeling of a system

Course Outcomes:

Students should be able to

1. Construct class model from a given description of the system
2. Organize the class model in the form of class relationships
3. Build a state model from a given description of the system
4. Develop the code for the state model in an object oriented language

Implement a mini project using the following steps as guidelines

1. Identify the classes, their attributes and methods for a given system
2. Convert the identified classes in the system to java code.
3. Identify the HAS-A relationship among the classes, represent those relationships in a class diagram and code the class diagram into a java code
4. Identify the IS-A relationships among the classes, represent those relationships in a class diagram and code the class diagram into a java code
5. Inspect all the classes and identify whether an object of a class changes its state during its life-cycle, draw the state transition using the state diagram for that object
6. Convert the state transitions into java code
7. Save the persistent data into a file and refine your code

Text Books

1. Michael Blaha, James Rumbaugh Object oriented modeling and Design with UML second edition, Pearson

Reference Books

1. Grady Booch, Object Oriented Analysis and Design with applications third edition, Adison Wesley Object Technology Series