PARUL UNIVERSITY - FACULTY OF ENGINEERING & TECHNOLOGY

Department of Computer Science & Engineering

SYLLABUS FOR 3rd Sem BTech PROGRAMME

Data Structures & Algorithms (203105205)

Type of Course: B.Tech

Prerequisite: Fundamentals of Knowledge of Programming & C Language

Rationale: This course is design to provide fundamentals of data structures. This subject provides basic knowledge of performance analysis and measurements and implementation of different data structure and algorithm using programming language.

Teaching and Examination Scheme:

Teaching Scheme (Hrs./Week)			Credit	Examination Scheme					Total
Lect	Tut	Lab		External		Internal			
				T	P	T	CE	P	
4	0	2	5	60	30	20	20	20	150

Lect - Lecture, **Tut** - Tutorial, **Lab** - Lab, **T** - Theory, **P** - Practical, **CE** - CE, **T** - Theory, **P** - Practical

Contents:

Sr.	Topic	Weightage	Teaching Hrs.
1	**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. **Searching:** Linear Search and Binary Search Techniques and their complexity analysis.	13%	7
2	**Stacks and Queues:** ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation –corresponding algorithms and complexity analysis. ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.	24%	13
3	**Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.	21%	12
4	**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.	21%	12
5	**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.	13%	7
6	**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.	8%	5

***Continuous Evaluation:**

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.

Reference Books:

1. An Introduction to Data Structures with Applications
   Jean-Paul Tremblay, Paul G. Sorenson; Tata McGraw-Hill;2nd Edition, (2007)
2. Data Structures using C & C++
   Tanenbaum; Prenctice-Hall International.
3. Fundamentals of Computer Algorithms
   E. Horowitz, S. Sahni, and S. Rajsekaran; Galgotia Publication
4. Fundamentals of Data Structures in C++-
   Sartaj Sahani
5. Data Structures: A Pseudo-code approach with C
   Gilberg & Forouzan Publisher; Thomson Learning.

Course Outcome:

1. For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.
2. For a given Search problem (Linear Search and Binary Search) student will able to implement it.
3. For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.
4. Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.
5. Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.

List of Practical:

1 Introduction to pointers.
(a) Call by Value (write a function that return max of two passed value)
(b) Call by reference.(write a function to swap value of two variable)

2 Introduction to Dynamic Memory Allocation.
DMA functions malloc(), calloc(), free() etc.
(a) W.A.P. to create dynamic int array using malloc() and free()
(b) W.A.P. to create dynamic char array using calloc() and free()

3 Write a program to implement structure in c.

4 Write a program to implement (a) linear Search
(b) Binary Search

5 Write a program to implement (a) Bubble Sort
(b) Insertion Sort
(c) Selection Sort

6 Implement a program for stack that performs following operations using array.
(a)PUSH (b) POP
(c) PEEP (d) CHANGE
(e) DISPLAY

7 Implement a program to convert infix notation to postfix notation using stack.

8 Implement a program to evaluate postfix notation.

9 Write a program to implement QUEUE using arrays that performs following operations
(a)INSERT (b) DELETE (c) DISPLAY

10 Write a menu driven program to implement following operations on the singly linked list.
(a) Insert a node at the front of the linked list.
(b) Insert a node at the end of the linked list.

11 Write a menu driven program to implement following operations on the singly linked list.
(a)) Insert a node at the specified position
(b) Delete a first node of the linked list.

12 Write a menu driven program to implement following operations on the singly linked list.
(a) Delete a node before specified position.
(b) Delete a node after specified position.

13 Write a program to implement following operations on the doubly linked list.
(a) Insert a node at the front of the linked list.
(b) Insert a node at the end of the linked list.

14 Write a program to implement following operations on the doubly linked list.
(a) Delete a last node of the linked list.
(b) Delete a node before specified position.

15 Write a program to implement following operations on the circular linked list.
(a) Insert a node at the end of the linked list.
(b) Insert a node before specified position.

16 Write a program to implement following operations on the circular linked list.
(a) Delete a first node of the linked list.
(b) Delete a node after specified position.

17 Write a program to implement stack using linked list.

18 Write a program to implement queue using linked list.

19 Write a program to create binary Tree Traversal.

20 Write a program to implement Prim’s and Kruskal’s algorithm