B. P. PODDAR INSTITUTE OF MANAGEMENT & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACADEMIC YEAR: 2018-2019 [ODD SEMESTER]

LIST OF EXPERIMENTS

Course: Data Structure & Algorithm Lab

Code: CS392

Branch: CSE

TOPIC	LIST OF EXPERIMENTS	CO	PO/ PSO
Implementation of array operations	1. Static 1-D array representation 2. Insert and delete an element from a specified location in a 1-D static array 3. Reverse the elements present in a 1-D static array delete duplicates element along with the mother element in a 1-D static array 4. Find the largest and smallest element present in a 1-D static array 5. Write a program to store the elements in a 2D array and display it and represent it in row major order & display it.	CO1, CO2	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Stack and Queues: Adding & deleting the elements	1. Implement a stack using array implementation including the function to check whether the stack is empty. Push an element into the stack; pop an element from a stack. Return the top element from the stack,display the stack elements 2. Implement a stack using Linked list. 3. Reverse a string using stack. 4. Representation of queue and implementation of basic queue operations enqueue, dequeue, overflow, underflow using linked list. 5. Solution to Tower of Hanoi using recursion	CO2	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Circular Queue Adding & deleting elements	1. Implementation of Circular queue using array.	CO2	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Merging Problem	1. Merging two sorted arrays so that the resultant array is also sorted 2. Merging two singly linked lists	CO1, CO2, CO4	PO1, PO2, PO3,

			PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Evaluation of Expressions	1. Evaluation of postfix expression. 2. Conversion from infix to postfix	CO2	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Implementation of linked lists**:** inserting, deleting inverting a linked list	3. Representation of single linked lists (create and display it) and then implementation of relevant operations – add, delete from beginning, end and at and after specified locations 4. Reversing the links of a singly linked list 5. Sorting the nodes of a singly linked list in ascending order	CO1 CO2, CO4, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Implementation of stacks & queues using linked lists	1. Implementation of stack using linked lists 2. Implementation of queue using linked lists	CO1 CO2, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Polynomial addition, polynomial multiplication	1. linked representation of polynomials 2. Addition of two polynomials	CO1 CO2, CO5	PO1, PO2, PO3, PO4, PO8, PO9,

			PO10, PO12, PSO1, PSO2
Sparse matrices: addition, multiplication	1. Write a program to test a given matrix is sparse or not. If it is sparse then represent it as 3-tuple format. 2. Find the transpose of a sparse matrix	CO1, CO2	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Recursive & Non recursive traversal of trees	1. Write a program to create recursive binary search tree 2. Write a program to implement Recursive and non-recursive traversal of a binary search tree - post order, pre-order and in-order traversal	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Threaded binary tree traversal	Threaded binary tree implementation.	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
AVL tree implementation	Implementation of Height Balanced (AVL) tree	CO3, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1,

			PSO2
Hash table implementation: searching, inserting, deleting	1.Hash tables implementation - concepts search, insert and delete	CO2, CO5	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2
Searching and sorting technique	1. Searching - Linear/Sequential, Binary (using function) 2. Implementation of bubble sort & modified bubble sort 3. Implementation of insertion sort, selection sort 4. Implementation of Quick sort, Merge sort	CO4	PO1, PO2, PO3, PO4, PO8, PO9, PO10, PO12, PSO1, PSO2