Chaitanya Bharathi Institute of Technology
Department of Information Technology

Log book of Data Structures Lab

Following are the inputs/practice exercises for excelling in DS Concepts, and for preparing for your Lab exams and placements:

NOTE: See to it that you follow the coding standards

a. Naming conventions – variable, function, class, program
b. Indentation
c. Commenting
d. Proper usage of constructors and destructors in each and every program
e. Dynamic allocation of memory using new and delete operators
f. Overloading of operators cin and cout to read the objects of the class under consideration
g. Interface (main()) should be user friendly, the user must be free to execute any operation any no. of times in any order
h. Need not restrict yourself to a single class per program, can be more than one as your program requires
i. Check necessary exceptions that may raise in every program of yours

S.No	Program	Operations Covered in the Lab	For further practice
		Week # 1 (13.07.2017 to 18.07.2017)	
1.	Stack ADT	• Implement stack using array • Implement a generic stack using templates • Implement stack using single linked list • Handle stack full and stack empty error conditions using exception handling	• Check whether parenthesis are balanced in a program • Convert infix to postfix, infix to prefix expression using stack • Evaluate postfix expression • Check whether a string is a palindrome using stack
		Week # 2 & 3 – (20.07.2017 – 25.07.2017) & (27.07.2017 to 01.08.2017)	
2.	Single Linked List &	• Create a List ADT(single linked list) • Initialise the head node to null in the constructor of list class • Describe the node class with necessary data members (to hold data part and address part)	• Merge n lists • Create a Copy of an existing list • Split the list • Merge two sorted lists n form a sorted

	Circular Linked List	• Data members of both list class and node class must be declared under private
		• Declare list class friend of node class
		• The following operations are must:
		  ○ Use templates, so that you can insert any data typed element in the node
		  ○ Insertion
		  ▪ Insert n nodes
		  ▪ Insert at beginning
		  ▪ Insert at the end of list
		  ▪ Insert a node after or before n^{th} node
		  ▪ Insert a node after or before an element
		  ○ Deletion
		  ▪ Delete n nodes
		  ▪ Delete head node
		  ▪ Delete last node
		  ▪ Delete a node after or before n^{th} node
		  ▪ Delete a node after or before an element
		  ▪ Delete a node with a particular element
		  ○ Display
		  ▪ Display the list in moving forward
		  ▪ Display the list in reverse
		  ○ Reversal
		  ▪ Reverse the list (Try inplace reversal, using recursion)
			list
			• Sort a list
			• Find the length of list
			• Print the distinct elements in the list
			• Remove duplicate elements in the list
			• Insert a value into a sorted list. (The list should be sorted after insertion also)
			• Check if there is any supported STL for lists
			• Find the difference between vector STL, array and Linked list

3.	Double Linked List	• Create a List ADT (Double linked list) • Initialise the head node to null in the constructor of list class • Describe the node class with necessary data members (to hold data part and address part) • Data members of both list class and node class must be declared under private	• Merge n lists • Create a Copy of an existing list • Split the list • Merge two sorted lists • Sort a list • Find the length of list

		• Declare list class friend of node class
		• The following operations are must:
		  ○ Use templates, so that you can insert any data typed element in the node
		  ○ Insertion
		  ▪ Insert n nodes
		  ▪ Insert at beginning
		  ▪ Insert at the end of list
		  ▪ Insert a node after or before n^{th} node
		  ▪ Insert a node after or before an element
		  ○ Deletion
		  ▪ Delete n nodes
		  ▪ Delete head node
		  ▪ Delete last node
		  ▪ Delete a node after or before n^{th} node
		  ▪ Delete a node after or before an element
		  ▪ Delete a node with a particular element
		  ○ Display
		  ▪ Display the list in moving forward
		  ▪ Display the list in reverse
		  ○ Reversal
		  ▪ Reverse the list (Try inplace reversal, using recursion)
			• Print the distinct elements in the list
			• Remove duplicate elements in the list
			• Insert a value into a sorted list. (The list should be sorted after insertion also)
			• Check if there is any supported STL for Double Linked list
			• Find the difference between vector STL, array and Linked list

Record your observations, after executing every program every week in your notes