

**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
<b>Week # 1 (05.07.2017 to 11.07.2017)</b>			
1.	Stack ADT	<ul style="list-style-type: none"><li>• Implement stack using array</li><li>• Implement a generic stack using templates</li><li>• Implement stack using single linked list</li><li>• Handle stack full and stack empty error conditions using exception handling</li></ul>	<ul style="list-style-type: none"><li>• Check whether parenthesis are balanced in a program</li><li>• Convert infix to postfix, infix to prefix expression using stack</li><li>• Evaluate postfix expression</li><li>• Check whether a string is a palindrome using stack</li></ul>
<b>Week # 2 &amp; 3 – (13.07.2017 – 18.07.2017) &amp; (20.07.2015 – 25.07.2017)</b>			
2.	Single Linked List	<ul style="list-style-type: none"><li>• Create a List ADT(single linked list)</li><li>• Initialise the head node to null in the constructor of list class</li><li>• Describe the node class with necessary data members (to hold</li></ul>	<ul style="list-style-type: none"><li>• Merge n lists</li><li>• Create a Copy of an existing list</li><li>• Split the list</li></ul>



**Chaitanya Bharathi Institute of Technology**  
**Department of Information Technology**

	&  Circular Linked List	data part and address part) <ul style="list-style-type: none"> <li>• Data members of both list class and node class must be declared under private</li> <li>• Declare list class friend of node class</li> <li>• The following operations are must:             <ul style="list-style-type: none"> <li>○ Use templates, so that you can insert any data typed element in the node</li> <li>○ Insertion                 <ul style="list-style-type: none"> <li>▪ Insert n nodes</li> <li>▪ Insert at beginning</li> <li>▪ Insert at the end of list</li> <li>▪ Insert a node after or before n<sup>th</sup> node</li> <li>▪ Insert a node after or before an element</li> </ul> </li> <li>○ Deletion                 <ul style="list-style-type: none"> <li>▪ Delete n nodes</li> <li>▪ Delete head node</li> <li>▪ Delete last node</li> <li>▪ Delete a node after or before n<sup>th</sup> node</li> <li>▪ Delete a node after or before an element</li> <li>▪ Delete a node with a particular element</li> </ul> </li> <li>○ Display                 <ul style="list-style-type: none"> <li>▪ Display the list in moving forward</li> <li>▪ Display the list in reverse</li> </ul> </li> <li>○ Reversal                 <ul style="list-style-type: none"> <li>▪ Reverse the list (Try inplace reversal, using recursion)</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Merge two sorted lists n form a sorted list</li> <li>• Sort a list</li> <li>• Find the length of list</li> <li>• Print the distinct elements in the list</li> <li>• Remove duplicate elements in the list</li> <li>• Insert a value into a sorted list. (The list should be sorted after insertion also)</li> <li>• Check if there is any supported STL for lists</li> <li>• Find the difference between vector STL, array and Linked list</li> </ul>
3.	Double Linked List	<ul style="list-style-type: none"> <li>• Create a List ADT (Double linked list)</li> <li>• Initialise the head node to null in the constructor of list class</li> <li>• Describe the node class with necessary data members (to hold data part and address part)</li> </ul>	<ul style="list-style-type: none"> <li>• Merge n lists</li> <li>• Create a Copy of an existing list</li> <li>• Split the list</li> <li>• Merge two sorted lists</li> </ul>

**Chaitanya Bharathi Institute of Technology**  
**Department of Information Technology**

		<ul style="list-style-type: none"> <li>Data members of both list class and node class must be declared under private</li> <li>Declare list class friend of node class</li> <li>The following operations are must: <ul style="list-style-type: none"> <li>Use templates, so that you can insert any data typed element in the node</li> <li>Insertion <ul style="list-style-type: none"> <li>Insert n nodes</li> <li>Insert at beginning</li> <li>Insert at the end of list</li> <li>Insert a node after or before n<sup>th</sup> node</li> <li>Insert a node after or before an element</li> </ul> </li> <li>Deletion <ul style="list-style-type: none"> <li>Delete n nodes</li> <li>Delete head node</li> <li>Delete last node</li> <li>Delete a node after or before n<sup>th</sup> node</li> <li>Delete a node after or before an element</li> <li>Delete a node with a particular element</li> </ul> </li> <li>Display <ul style="list-style-type: none"> <li>Display the list in moving forward</li> <li>Display the list in reverse</li> </ul> </li> <li>Reversal <ul style="list-style-type: none"> <li>Reverse the list (Try inplace reversal, using recursion)</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Sort a list</li> <li>Find the length of list</li> <li>Print the distinct elements in the list</li> <li>Remove duplicate elements in the list</li> <li>Insert a value into a sorted list. (The list should be sorted after insertion also)</li> <li>Check if there is any supported STL for Double Linked list</li> <li>Find the difference between vector STL, array and Linked list</li> </ul>
<b>Week # 4 (27.07.2017 to 01.08.2017)</b>			
4.	Applications of Stacks	<ul style="list-style-type: none"> <li>Implement Infix to Postfix Conversion using user defined stack from the previous week (your expression should consist of parentheses also. Eg: 2*3+(5-2)/3 )</li> <li>Implement Infix to prefix conversion using stack STL (also</li> </ul>	<ul style="list-style-type: none"> <li>Create your own header file which should have the definitions of SLL, DLL, CLL, stack in it (Your definition)</li> <li>Write a program to convert a postfix</li> </ul>



**Chaitanya Bharathi Institute of Technology**  
**Department of Information Technology**

		include logical, arithmetic and relational operators). <ul style="list-style-type: none"> <li>Reverse the content of a 1Dimensional array using stack</li> <li>Evaluate a postfix expression using Stack</li> <li>Can you modify the above program so that it evaluates expressions of the form <math>4.99 * 1.06 + 5.99 + 6.99 * 1.06</math></li> <li>Check for balanced parentheses in an expression (consider all three kinds of parentheses( ),{ },[ ])</li> <li>Towers of Hanoi</li> </ul>	expression to infix
<b>Week #5 (03.08.2017 to 10.08.2017)</b>			
5.	Queues	<ul style="list-style-type: none"> <li>Efficiently implement a queue ADT using a circular array. You may use a vector (rather than a primitive array) as the underlying array structure.</li> <li>Implement Generic Queue using linked list (let the queue handle string (STL) type data (note: Not character array)).</li> <li>Implement circular Queue</li> <li>A deque is a data structure consisting of a list of items on which the following operations are possible:               <ul style="list-style-type: none"> <li>push(x): Insert item x on the front end of the deque.</li> <li>pop(): Remove the front item from the deque and return it.</li> <li>inject(x): Insert item x on the rear end of the deque.</li> <li>eject(): Remove the rear item from the deque and return it.</li> </ul> </li> <li>Write routines to support the deque that take O(1) time per operation.</li> </ul>	<ul style="list-style-type: none"> <li>How to implement a Stack using two Queues?</li> <li>How you can implement all the functions of the deque ADT using two stacks.</li> <li>Implement Deque using STL deque.</li> <li>Implement an iterator ADT for DLL</li> </ul>

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