Data Structures

Lab Manual



**Topic: List**

Session: Spring 2023

# Faculty of Information Technology

UCP Lahore Pakistan

**Objectives:**

To write a program that implements basic concept of list. The purpose is understanding the first basic concept of Data Structure and Algorithm, basic operations (insertion, deletion, searching) and brining the best of students.

Data Structure

Non

-

Linear

Elements in non

-

linear

data structure are not

in sequence.

Elements are arranged

in particular order.

Easy to implement.

Linear

Linear Data

Structure

Array

Stack

Queue

Linked

list

Non

-

Linear

Data Structure

Tree

Graph

Following are the operations that you can perform

* Insertion
* Deletion
* Sorting
* Searching

Here is an example code for the help.

#include <iostream> using namespace std;

class List { public : List () { }

List (int s){size=s; p= new int[size];use=-1;} bool isEmpty (){ if(use==-1) return true; else return false;

} bool isFull (){ if(use==size-1) return true; else return false;

}

int getUse() { return use;} bool insert(int data){

use++; if (!isEmpty()){ p[use]=data; return true; } else return false;

} bool deleteValue(int index){ if ((index>=0)&&(index<=use)){ for(int j=index;j<use;j++) p[j]=p[j+1]; use=use-1; return true; } else return false;

} void print(){ int i=0; while(i<=use){ cout<<p[i]; i++;

}

}

int getValue(int i) { return p[i];}

private: int \*p; int size; int use;

};

int main(){ List a(3); while (!a.isFull()){ int i; cin>>i;

a.insert(i); } int k;

cout<<" Enter index for deletion"; cin>>k; if (a.deleteValue(k))

cout<<k <<" value successfully deleted"; else

cout<<k <<" index out of bounds ";

int i=0; int value; while (i<=a.getUse()){ value=a.getValue(i); cout<<value; i++;

}

}

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

**Students are required to complete the following tasks in lab timings.**

## Task 1

Create a C++ generic abstract class named as **List**, with the following:

**Attributes:**

1. Type \* arr;
2. int maxSize;
3. int currentSize;

**Functions:**

virtual void addElement (Type) = 0;

* + Should add the element on the **List** virtual void addElementAtFirstIndex(Type) = 0;
  + Should add the element at the first position of the **List** virtual void addElementAtLastIndex(Type) = 0;
  + Should add the element at the last position of the **List** virtual Type

* + Write parameterized constructor with default arguments for the above class.
  + Write Copy constructor for the above class.
  + Write Destructor for the above class.

## Task 2

Create a menu based program for the following functions, using the class made in task 1, make a class named as **MyList**, having following additional functionalities:

**bool** [**empty()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Returns whether the MyList is empty or not **bool** [**full()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) **:** Returns whether the MyList is full or not **int** [**size()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Returns the current size of the MyList

**bool insertAt(int index, T value):** Adds a value at the index passed to the function, returns true if the index is present and value is added else returns false. **Type** [**last()**](https://www.geeksforgeeks.org/stack-top-c-stl/) : Returns the last element of the MyList

* Write parameterized constructor with default arguments for the above class.
* Write Copy constructor for the above class.
* Write Destructor for the above class.