**CSE225L – Data Structures and Algorithms Lab**

**Lab 04**

**Unsorted List (array based)**

In today’s lab we will design and implement the List ADT where the items in the list are unsorted.

|  |
| --- |
| #ifndef UNSORTEDTYPE\_H\_INCLUDED  #define UNSORTEDTYPE\_H\_INCLUDED  const int max\_items = 5;  template<class T>  class UnsortedType  {  private:  T info[max\_items];  int length;  int currentPos;  public:  UnsortedType();  bool isEmpty();  bool isFull();  void makeEmpty();  void insertItem(T);  void deleteItem(T);  void retriveItem(T&, bool&);  int lengthIs();  void resetList();  void getNextItem(T&);  };  #endif // UNSORTEDTYPE\_H\_INCLUDED |

Generate the **driver file (main.cpp)** where you perform the following tasks. Note that you cannot make any change to the header file or the source file.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation to Be Tested and Description of Action** | | **Input Values** |  | **Expected Output** |
|  | |  |  |  |
| • Create a list of integers | |  |  |  |
|  |  |  |  |  |
| • | Insert four items | 5 7 6 9 |  |  |
|  |  |  |  | |
| • | Print the list |  |  | 5 7 6 9 |
|  |  |  |  | |
| • | Print the length of the list |  |  | 4 |
|  |  |  |  |  |
| • | Insert one item | 1 |  |  |
|  |  |  |  | |
| • | Print the list |  |  | 5 7 6 9 1 |
|  |  |  |  |  |
| • | Retrieve 4 and print whether found or not |  |  | Item is not found |
|  |  |  |  |  |
| • | Retrieve 5 and print whether found or not |  |  | Item is found |
|  |  |  |  |  |
| • | Retrieve 9 and print whether found or not |  |  | Item is found |
|  |  |  |  |  |
| • | Retrieve 10 and print whether found or not |  |  | Item is not found |
|  |  |  |  |  |
| • | Print if the list is full or not |  |  | List is full |
|  |  |  |  |  |
| • | Delete 5 |  |  |  |
|  |  |  |  |  |
| • | Print if the list is full or not |  |  | List is not full |
|  |  |  |  |  |
| • | Delete 1 |  |  |  |
|  |  |  |  | |
| • | Print the list |  |  | 7 6 9 |
|  |  |  |  |  |
| • | Delete 6 |  |  |  |
|  |  |  |  | |
| • | Print the list |  |  | 7 9 |

* Write a class studentInfo that represents a student

record. It must have variables to store the student ID, student’s name and student’s CGPA. It also must have a function to print all the values. You will also need to overload a few operators.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| • Create a list of objects of class studentInfo. | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| • | Insert 5 student records | 15234 | Jon | 2.6 |  |  |  |
|  |  | 13732 | Tyrion | | 3.9 |  |  |
|  |  | 13569 | Sandor | | 1.2 |  |  |
|  |  | 15467 | Ramsey 2 3.1 | | |  |  |
|  |  | 16285 | Arya | 3.1 | |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| • Delete the record with ID 15467 | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| • Retrieve the record with ID 13569 and print whether | |  |  |  |  |  | Item is found |
|  | found or not along with the entire record |  |  |  |  |  | 13569, Sandor, 1.2 |
|  |  |  |  |  |  |  |  |
| • | Print the list |  |  |  |  |  | 15234, Jon, 2..6 |
|  |  |  |  |  |  |  | 13732, Tyrion, 3.9 |
|  |  |  |  |  |  |  | 13569, Sandor, 1.2 |
|  |  |  |  |  |  |  | 16285, Arya, 3.1 |