|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in IT, CSF  Year 2 (2024/25) Semester 4 | **Week 2** |
| **1-2 Hours** |
| **Tutorial 2 – Data Abstraction** | |

1. List 3 real-life examples in additional to the two already given, where the concept of lists can be adopted.

* Name list
* Contact list
* List of employees in a company

2. If listA is an empty list of integers, what does it contain after the following statements are executed?

|  |  |
| --- | --- |
| **Statements** | **Content of listA** |
| listA.add(20); | [20] |
| listA.add(0, 30); | [30, 20] |
| listA.add(10); | [30, 20, 10] |
| listA.add(2, 50); | [30, 20, 50, 10] |
| listA.add(1, 40); | [30, 40, 20, 50, 10] |
| listA.remove(1); | [30, 20, 50, 10] |
| listA.remove(2); | [30, 20, 10] |

3. Suppose you want to include another operation in the List ADT to display all the items in the list:

(a) Specify the operation (as in .h)

|  |
| --- |
| class List {  public:     void displayList();  }; |

(b) Implement the operation (as in .cpp)

|  |
| --- |
| #include <iostream>  #include "List.h"  using namespace std;  void List::displayList() {      for (int i = 0; i < size; i++) {          cout << array[i] << " ";      }      cout << endl;  } |

4. Suppose you want to include another operation in the List ADT to replace an item in a certain specified position in the list:

(a) Specify the operation

|  |
| --- |
| class List {  public:       void replace(int index, int newValue);  }; |

(b) Implement the operation

|  |
| --- |
| void List::replace(int index, int newValue) {      if (index >= 0 && index < size) {          array[index] = newValue;  // Replace the element at the specified index      } else {          cout << "Index out of bounds!" << endl;  // Error handling for invalid index      }  }  // This is assuming that the List has a variable size which is the total number of elements present within the list (array) |