COLLEGE OF COMPUTING AND INFORMATICS

CSEB3213 DATA STRUCTURES AND ALGORITHMS

SEM 1 2024/2025

LAB 2: INTRODUCTION TO STL LIST

Objectives

Introduction on:

1. STL List as dynamic container
2. Member functions of STL List

Instruction

1. This is an individual lab exercise.
2. You are compulsory to complete **ALL QUESTIONS for Level Easy and Moderate during lab session.** Level Challenging Question is for Self-Revision Exercise.
3. You are given only **90 minutes** to complete the program.
4. Compile and submit your complete **cpp** programs via Brighten.
5. Do attach this code segment in all files:

```
/*Subject code : CSEB3213 Data Structures & Algorithms
  Section      : 02C
  Student name : XXX
  Student ID no: XXX
  Question no  : XXX */
```

LEVEL: EASY

Question 1 (10 marks)

The following C++ program contains errors and is incomplete. Transfer the code to an IDE and fix the errors.

```
#include<iostream>

using namespace std;

int main() {
    list <float>;
    int i = 0;
    for (;i<5;i++)
        mylist.pushback(10.0);
    cout << "Total data in record : " << mylist.length()<< endl;
    mylist[2] = 18.0;
    mylist.at(last) = 5.3;
    cout << "List of data in record: ";
    for (i=0;i<5;i++)
        cout <<mylist[i]<<" ";
    sort(mylist_ascending);
    cout << "\nData in ascending mode: ";
    for (i=0;i<5;i++)
        cout <<mylist[i]<<" ";
    sort(mylist_descending);
    cout << "\nData in descending mode: ";
    for (i=0;i<5;i++)
        cout <<mylist[i]<<" ";

    return 0;
}
```

Sample of Output

```
Total data in record : 5
List of data in record: 10 10 18 10 5.3
Data in ascending mode: 5.3 10 10 10 18
Data in descending mode: 18 10 10 10 5.3
```

LEVEL: MODERATE

Question 2 (15 marks)

Study the incomplete program below and provide solutions for each of the commented sections.

```
#include<iostream>
#include<list>
using namespace std;

int main() {
    list<string>greeting {"Hello","<name>"};
    list<string>word1 {":)","You have","X","credit","(s)","more to go."};
    list<string>word2 {"Congratulation!!","Hang in there!!","Almost done!!"};
    list<string>wish;
    string name; int credit,balance; char choice;

    //prompt user to enter input name and credit [2m]

    //calculate the remaining credit
    balance = 120 - credit;

    //replace "<name>" in the list greeting with the provided name [1m]

    //copy the contents of the list greeting to the list wish [1m]

    //replace "X" in the list word1 with the balance value [2m]

    if (balance == 0){
        //the list wish should contain the generated output as shown in Option 1 [3m]
    }
    else if(balance<60){
        //the list wish should contain the generated output as shown in Option 2 [3m]
    }
    else if(balance>60){
        //the list wish should contain the generated output as shown in Option 3 [3m]
    }
    cout<<"\nResult:\n";
    for(auto x : wish)
        cout<<x<<" ";

    return 0;
}
```

Sample of Output

Option 1:

```
Enter name : Naziffa Raha
Enter current total credit : 120

Result:
Hello Naziffa Raha :) You have 0 credit more to go. Congratulation!!
```

Option 2:

```
Enter name : Naziffa Raha
Enter current total credit : 90

Result:
Hello Naziffa Raha :) You have 30 credit (s) more to go. Almost done!!
```

Option 3:

```
Enter name : Naziffa Raha
Enter current total credit : 48

Result:
Hello Naziffa Raha :) You have 72 credit (s) more to go. Hang in there!!
```

LEVEL: CHALLENGING (Self-Lab Revision Exercise)

Question 3

Source: Lab Test Semester 2 2020/2021

It's essential to consume enough protein daily to meet your body's needs. Protein plays a crucial role in maintaining proper fluid balance, building and repairing tissues, transporting nutrients, and fulfilling other vital functions. The average adult requires a minimum of 0.8 grams of protein per kilogram of body weight each day. Using the requirements outlined below, create a complete C++ program that utilizes the Standard Template Library (STL) to calculate the protein intake for each user.

Requirements:

1. In the **main()** function, define a **list** named **Record** with data fields for **name** (string), **weight** (double) and **protein** (double).
2. Implement a **dataWeight()** function to accept input data for weight, utilizing a sentinel-controlled loop.
3. Implement a **dataProtein()** function to calculate and assign the protein values in grams.
4. Implement a **displayData()** function to display all records.
5. Implement an **updateData()** function to modify specific record using an iterator.

Sample of Output

```
Enter Name: John Lee
Enter weight (kg): 75.0
Press [Y] for next input: Y
Enter Name: Hussein Hassan
Enter weight (kg): 56.7
Press [Y] for next input: N

List of Records:
1. Name: John Lee, Weight (kg): 75.0, Protein required (gram): 60.00
2. Name: Hussein Hassan, Weight (kg): 56.7, Protein required (gram): 45.36

Would you like to update specific data [Press Y for Yes]: Y
Enter name: John Lee
Enter new weight input (kg): 80.0

List of Records:
1. Name: John Lee, Weight (kg): 80.0, Protein required (gram): 64.00
2. Name: Hussein Hassan, Weight (kg): 56.7, Protein required (gram): 45.36

End of program.
```