COLLEGE OF COMPUTING AND INFORMATICS

CSEB3213 DATA STRUCTURES AND ALGORITHMS

SEM 2 2024/2025

LAB 3: SINGLY LINKED LIST IMPLEMENTATION

Objectives

Introduction on Singly Linked List concept and operations using C++ programming language.

Instruction

1. This is an individual lab exercise.
2. You are compulsory to complete **ALL QUESTIONS for Level Easy and Moderate** except Level Challenging (Self-Lab Revision Exercise).
3. Compile and submit your complete **cpp** programs via Brighten.
4. Do attach this code segment in all files:

```
/*Subject code : CSEB3213 Data Structures and Algorithms
  Section      : 01A or 01B or 01B
  Student name : XXX
  Student ID no: XXX
  Question no  : XXX */
```

LEVEL: EASY

Question 1 (5 marks)

Study the STL list program below (Program Sample #1).

Program Sample #1

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

void insertNode(list<int>*Record) {
    int year; char choice;
    do {
        cout<<"Enter year : ";
           cin>>year;
        Record->push_back(year);
        cout<<"Press [y] for new input:";
        cin>>choice;
    } while(choice=='y');
    cout<<endl;
}

void display(list<int>Record) {
    cout<<"All records : ";
    for(auto i : Record)
        cout<<i<<" ";
    cout<<endl;
}

int main() {
    list<int>Record;
    insertNode(&Record);
    display(Record);

    return 0;
}
```

Without modifying the program flow, convert the STL list program to Singly Linked List implementation using the template (Program Sample #2) provided below.

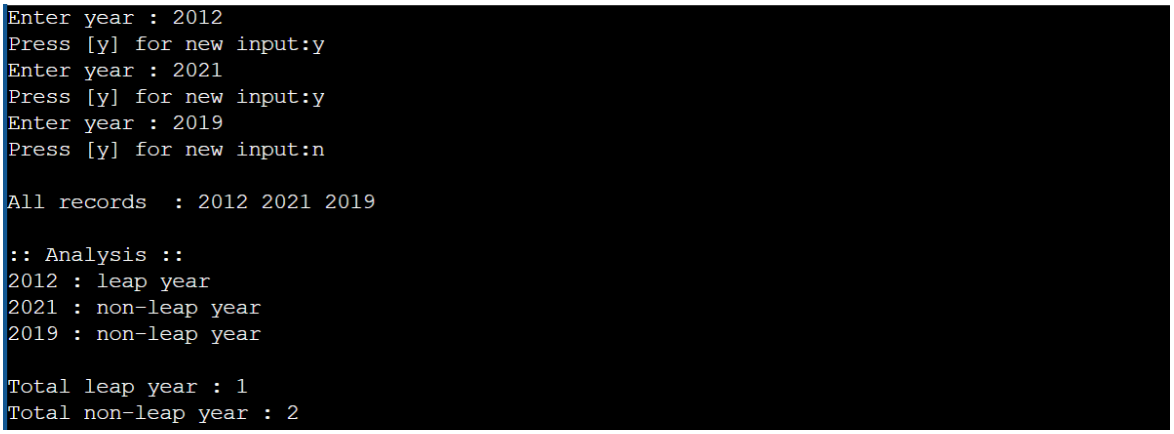
Program Sample #2

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

class Record {
    public:
    int year;
    //missing code
};


Record *createNode() {
    Record *n = new Record ();
    cout<<"Enter year : ";
    //missing code
}


void insertNode(/*suitable parameter*/) {
    //variable declaration

    do{
        n = createNode();

        //insert first node into linked list
        //missing code

        //insert second node onwards at the end of linked list
        else {
            //missing code
        }

        cout<<"Press [y] for new record:";
        cin>>choice;
    }while(choice=='y');
}

void display(/*suitable parameter*/) {
    cout<<"All records  : ";
    //missing code
    cout<<endl;
}

int main() {
    Record *head = NULL;
    insertNode(/*suitable argument*/);
    //call display()
    //call analysis()
    return 0;
}
```

Question 2 (5 marks)

Modify the program by adding function **analysis()**. The function shall display the year type in the linked list, whether leap or non-leap.

Sample of Output



```
Enter year : 2012
Press [y] for new input:y
Enter year : 2021
Press [y] for new input:y
Enter year : 2019
Press [y] for new input:n

All records  : 2012 2021 2019

:: Analysis ::
2012 : leap year
2021 : non-leap year
2019 : non-leap year

Total leap year : 1
Total non-leap year : 2
```

LEVEL: MODERATE

Question 3 (15 marks)

Referring to Program Sample #3 below, complete the following questions using **Singly Linked List implementation**.

Program Sample #3

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

struct Subject {
       //Question 3(1)
string remark = “Normal”;
};

int menu() {
       int choice;
       cout << "::SUBJECT REGISTRATION::\n";
       cout << "1. Add subject\n";
       cout << "2. Display subject\n";
       cout << "3. Update subject\n";
       cout << "4. Exit program\n";
       cout << "Enter choice: ";
       cin >> choice;
       return choice;
}

Subject *subjectInfo() {
       //Question 3(2)
       //create a new node
       //accept user’s data inputs
       //return address of new node to registerSubject()
}

void registerSubject(/*suitable parameters*/){
       //Question 3(3)
       cout<<":: Add Subject Record::"<<endl;
       //invoke subjectInfo () and accept new node’s address
       //add new node to linked list
}

void display(/*suitable parameters*/) {
       //Question 3(4)
       cout<<":: Display Subject Record::"<<endl;
       cout<<"Press [1] for individual subject or [2] for all subject : ";
       cin>>choice;

       //display record according to user’s input (option)
}

int main() {
       //declaration of variables
       cout<<"Enter student name : ";
       getline(cin, name);
       cout<<"Enter student ID no : ";
       getline(cin, idNo);
       do {
            choice = menu();
            switch (choice) {
               case 1: //call function registerSubject()
               case 2: //call function display()
              }
       } while (choice != 4);
       return 0;
}
```

1. Create a struct named ***Subject*** that holds data members as follows: **[2 marks]**

   - string : sname, scode, remark (default value for remark is "*Normal"*)
   - integer : credit
   - float : fee

2. Complete a function named **subjectInfo()**. This function should accept all input of struct data members. **[5 marks]**

   - Fee for each credit is RM500.00
   - For late subject registration, a penalty of RM100.00 will be charged for each subject. Set the remark to "*Late Registration*".

3. Complete a function named **registerSubject()**.This function should add/register the subject into linked list.

   [3 marks]

4. Complete a function named **display ()** based on following options:

   - Option 1 (individual): User will enter subject code as input and function will display the details of individual subject. **[3 marks]**
   - Option 2: Function will display details of all registered subjects in record. **[2 marks]**

LEVEL: CHALLENGING

SELF-LAB REVISION EXERCISE

Question 4

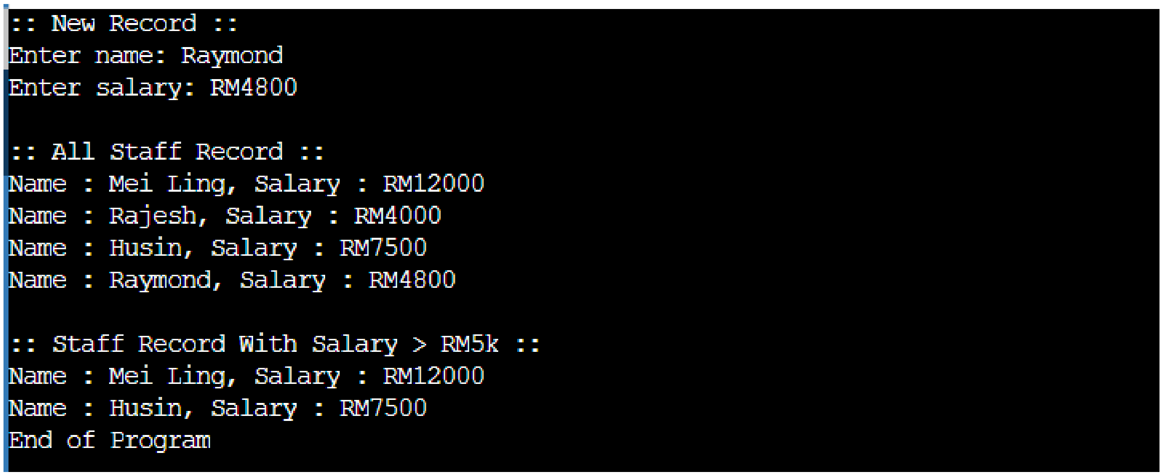
Source: Sem 1 2022/2023 Lab Test Set 1

This program will generate a list of staff with salary greater than RM5000. Using the requirements and incomplete program provided below, write a complete C++ program using combination of **STL Vector and Linked List** implementation.

Order Tracking Program

```
//suitable header(s)
using namespace std;

struct Data{
    string name;
    float salary;
};

struct Node{
    string name;
    float salary;
    Node *next;
};

void filterRecord(/*suitable parameter(s)*/){

    /*this function shall copy all data from STL staff and store it in a singly
      linked list, ONLY if the value of the salary is greater than RM5000 */

}

void display(Node *head){
    cout<<"\n:: Staff Record With Salary > RM5k ::"<<endl;

    /*this function shall display all data in the singly linked list */
}

int main()
{
    vector<Data>staff = {{"Mei Ling",12000}, {"Rajesh", 4000}, {"Husin", 7500}};
    Data temp;
    /*suitable variable(s)*/

    cout<<":: New Record ::"<<endl;
    /* this section shall prompt user to input data for staff name and salary.
    Store the input data at the end of the STL staff */

    cout<<"\n:: All Staff Record ::"<<endl;
    /*this function shall display all data in the STL staff using iterator */

    filterRecord(/*suitable argument(s)*/);
    display(head);
    cout<<"End of Program";

    return 0;
}
```

Sample of Output



```
:: New Record ::
Enter name: Raymond
Enter salary: RM4800

:: All Staff Record ::
Name : Mei Ling, Salary : RM12000
Name : Rajesh, Salary : RM4000
Name : Husin, Salary : RM7500
Name : Raymond, Salary : RM4800

:: Staff Record With Salary > RM5k ::
Name : Mei Ling, Salary : RM12000
Name : Husin, Salary : RM7500
End of Program
```