LAB - 9

Q1) WAP to declare a class which stores a complex number. Include a member function which compares the modulus of the two complex class objects and returns the object with higher value. Include a parameterized constructor which arguments with same name as that of the class data members.

```
#include <iostream>
using namespace std;
class Complex
{
public:
    int real;
    int imag, C1 = 0, C2 = 0;

    void setvalue()
    {
        cin >> real;
        cin >> imag;
    }
    void compare(Complex c1, Complex c2)
    {
        C2 = c2.real + c2.imag;
        C1 = c1.real + c1.imag;
        real = (c1.real * c1.real) + (c1.imag * c1.imag);
        imag = (c2.real * c2.real) + (c2.imag * c2.imag);
        if (real > imag)
        {
            cout << "C1 is bigger";
        }
        else if (imag > real)
        {
            cout << "C2 is bigger";
        }
    }
};
int main()
{
    Complex c1, c2, c3;
    cout << "Enter real and imaginary part of first complex number" << endl;
    c1.setvalue();
    cout << "Enter real and imaginary part of second complex number" << endl;
    c2.setvalue();
    cout << "------ Comparision -----" << endl;
    c3.compare(c1, c2);
    return 0;
}
```

```
Enter real and imaginary part of first complex number
10 20
Enter real and imaginary part of second complex number
20 30
------- Comparision ------
C2 is bigger
```

Q2) WAP in which there is a global variable, a local variable for main function and a variable in a nested scope inside main, with the same name. Print all the three variables.

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

string str="global scope";

int main()
{
    string str="local scope";
    if (true){
        string str="if scope";
        cout << str << endl;
    }
    cout << str << endl;
    cout << ::str << endl;
}
```

```
if scope
local scope
global scope
```

Q3) WAP to take input for two integer variables. Assign the value -1 to the variable with higher value using a function. [Use return by reference]

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

void ChangeHigherValue(int &a, int &b)
{
    if (a > b)
    {
        a = -1;
    }
    else if (a < b)
    {
        b = -1;
    }
}

int main()
{
    int a, b;
```

```
    cout << "Input for two integers: ";
    cin >> a >> b;

    if (!cin)
    {
        cout << "Error: invalid data\n";
        return 1;
    }

    ChangeHigherValue(a, b);

    cout << "a = " << a << "\n";
    cout << "b = " << b << "\n";

    return 0;
}
```

```
Input for two integers: 12
20
a = 12
b = -1
```

Q4) Create a class student which stores name, date-of-birth and date-of-joining of a student. The data members date-of-birth and date-of-joining should be the objects of another class called 'date'. Input the data for 10 students and display it.

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

#define MAX 10
class student
{
private:
    char name[30];
    int rollNo;

public:
    void getDetails(void);
    void putDetails(void);
};

class Date
{
private:
    int Dob, m1, m2, y1, y2;
    int Doj;

public:
    void DOBDetails(void);
    void DOJDetails(void);
};
```

```
void student::getDetails(void)
{
    cout << "Enter name: ";
    cin >> name;
    cout << "Enter roll number: ";
    cin >> rollNo;
}
void student::putDetails(void)
{
    cout << "Student details:\n";
    cout << "Name: " << name << ", Roll Number: " << rollNo << endl;
}
void Date::DOBDetails(void)
{
    cout << "Enter Date of birth";
    cin >> Dob;
    cin >> m1;
    cin >> y1;
    cout << "Enter Date of joining";
    cin >> Doj;
    cin >> m2;
    cin >> y2;
}
void Date::DOJDetails(void)
{
    cout << "Student details:\n";
    cout << "Date of birth: " << Dob << " / " << m1 << " / " << y1 <<",Date of joining: " << Doj << " / "
<< m2 << " / "<< y2 <<endl;
}
int main()
{
    student std[MAX];
    Date std1[MAX];
    int n, loop;
    cout << "Enter total number of students: ";
    cin >> n;
    for (loop = 0; loop < n; loop++)
    {
        cout << "Enter details of student " << loop + 1 << ":\n";
        std[loop].getDetails();
        std1[loop].DOBDetails();
    }
    cout << endl;
    for (loop = 0; loop < n; loop++)
    {
        cout << "Details of student " << (loop + 1) << ":\n";
        std[loop].putDetails();
        std1[loop].DOJDetails();
    }
    return 0;
}
```

```
Enter total number of students: 2
Enter details of student 1:
Enter name: Anupam
Enter roll number: 1
Enter Date of birth20 05 2002
Enter Date of joining01 07 2018
Enter details of student 2:
Enter name: Moharana
Enter roll number: 2
Enter Date of birth05 06 2002
Enter Date of joining03 08 2019

Details of student 1:
Student details:
Name: Anupam, Roll Number: 1
Student details:
Date of birth: 20 / 5 / 2002,Date of joining: 1 / 7 / 2018
Details of student 2:
Student details:
Name: Moharana, Roll Number: 2
Student details:
Date of birth: 5 / 6 / 2002,Date of joining: 3 / 8 / 2019
```

Q5) Write a program to demonstrate the order of call of constructors and destructors in case of multiple inheritance where one or more base classes are virtual.

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

class A
{
public:
    A()
    {
        cout << "A" << endl;
    }
};

class B
{
public:
    B()
    {
        cout << "B" << endl;
    }
};

class C : public A, public B
{
public:
    C()
    {
```

```
        cout << "C" << endl;
    }
};

int main()
{
    C c;
    return 0;
}
```