Lab No.6 Multiple and Multilevel Inheritance

6.1 Objectives of the lab

Introducing the concepts of inheritance such as

24 Function overriding
25 Multiple inheritance
26 Multilevel inheritance

This lab also provides insight into composition and multi-file programming.

6.2 Pre-Lab

6.2.1 Overriding member functions of the base class

1 A derived class can override the member functions of its base class
2 To override a function of base class, the derived class provides a function with same signature as that of the base class
3 So derived class function overrides the base class function
4 Base class functions are overridden if we want to change the functionality of base class function
   - Functions can be totally changed
   - A few new additions can be made in the overriding function
5 Derived class method and base class method have same signatures but when this function is called on a derived class object, the function of derived class object will be called although the base class functions are making the interface of the derived class
6 To explicitly call the base class overridden function, use parent_class_name::member_function_name(…)

```
#include <iostream.h>
class A
{
private:
    int     num1;
public:
    A(): num1(0)
    {}
    void    display()
    {       cout<<"Number 1: "<<num1<<endl;
    }
};
class B: public A
{
private:
    int     num2;
```

```
public:
    B():num2(0)
    {}
    void    display()
    {
            cout<<"Number 2: "<<num2<<endl;
    }
};
void main()
{
    B       obj;
    obj.display();    //display of class B called
    obj.A::display(); //display of class A called
}
```

6.2.2 Multiple inheritance

1. A class can be derived from more than one classes
2. The order in which the constructors of base classes are called is not dependent on the order of mentioning constructors in base class initializer. It is dependent on the order in which inheritance is specified in the derived-class definition

6.2.3 Example

```
#include <iostream.h>
class A
{
private:
    int     num1;
public:
    A():num1(0)
    {       cout<<"Class A constructor..."<<endl;
    }
};

class B
{
private:
    int     num2;
public:
    B():num2(0)
    {
            cout<<"Class B constructor..."<<endl;
```

```
    }
};

class C: public B, public A
{
public:
    C()
    {
        cout<<"Class C constructor..."<<endl;
    }
};

void main()
{
    C    obj;
}
```

Q. if there are display functions in each of the three classes, which class' display function gets called if the statement obj.display(); gets executed? Write statements to call the display function of all the three classes on obj.

6.2.4 Multifile programming

A program can be broken down into different files as:

6.2.5 Example

Header.h

```
#ifndef HEADER_H
#define HEADER_H
class Point
{
private:
    int     x, y;
public:
    Point(): x(0), y(0)
    {}
    Point operator+(Point);
    void    display();
};
#endif
```

Header.cpp

```
#include <iostream.h>
#include "header.h"

Point Point::operator+(Point rhs)
{
	Point temp;
	temp.x= x + rhs.x;
	temp.y= y + rhs.y;
	return temp;
}

void Point::display()
{
	cout<<"("<<x<<","<<y<<")"<<endl;
}
```

Mainfile.cpp

```
#include <iostream.h>
#include "header.h"
void main()
{
	Point p1(3, 4), p2(4, 2);
	Point p3;

	cout<<"First Point is: ";
	p1.display();

	cout<<"Second Point is: ";
	p2.display();

	p3= p1 + p2;
	cout<<"The result of Addition is: ";
	p3.display();
}
```

6.2 In-Lab

6.3.1 Activity

PART A:

Create a class **Person** that contains information about a person's identification, name, city, and contact number. Note that identification and contact number are unsigned integer while name and address

being string. Provide

a) a **no-argument constructor** for initializing the values of data members to some defaults.

b) a **4-argument constructor** to initialize the data members sent from the calling function at the time of creation of an object.

c) A set function for each class data member to set it. Use the following function names: **setID**, **setName**, **setCity**, and **setContact**.

d) A get function for each class data member to retrieve data. Use the following function names: **getID**, **getName**, **getCity**, and **getContact**.

e) A **display** function to display all the attributes of a Person.

PART B:

Next, create derived class **Patient** from **Person** class. This class contains only one data member: Admit_Date. Take this as string for simplicity. Now provide

a) a **no-argument constructor** for initializing the values of data members to some defaults.

b) a **5-argument constructor** to initialize all the data members sent from the calling function at the time of creation of an object.

c) a **2-argument constructor** having one argument to set date and other being the object of class **Perosn** to set other four data members.

d) A **setAdmit_Date** function for to set Admit_Date.

e) A **getAdmit_Date** function to retrieve Admit_Date.

f) A **display** function to display all the attributes of a Person.

PART C:

Finally, create two objects of Patient class where first object uses 2-argument constructor and second object uses 5-argument constructor.

PART D:

Derive two classes **OutPatient** and **ResPatient** from **Patient** class. OutPatient class contains one data member Checkback_Date while ResPatient also contains Discharge_Date. Take these dates as string for simplicity. Provide for each class a no-argument and 2-argument constructors; get and set functions for date data member; and show function that just show the date information.

Next create two objects of Patient class where first object uses 2-argument constructor and second object uses 5-argument constructor. Make first patient OutPatient and second patient ResPatient. Next

use show functions of patient and derived classes to display complete patient information.

6.3 Home-Lab

6.3.2 Activity

Create a base class **Liquid** that contains one private data member called **specific_gravity** of type float. This class contains **no-argument constructor, one-argument constructor, setSpecific_gravity function, getSpecific_gravity function,** and **show function.**

Create another base class **Fuel** that contains one private data member called **rate** of type float. This class also contains **no-argument constructor, one-argument constructor, setRate function, getRate function,** and **show function.**

Now create derived class **Petrol** from **Liquid** and **Fuel** classes**.** It contains one function to input the values of data members of base classes and other to show these values.

Finally, write main function and create object of Petrol class to show its functionality.

6.3.3 Activity

Redo all the questions of Lab5 using multi-file programming. Create a header file for your class declaration, a .cpp file for class definition, and another .cpp file containing main function.

6.4 References

15 Class notes

16 **Object-Oriented Programming in C++ by *Robert Lafore* (Chapter 9)**

17 **How to Program C++ by *Deitel & Deitel* (Chapter 9)**