Lab No.5 Inheritance

5.1 Objectives of the lab:

Introducing the concepts of inheritance

20 Base and derived classes
21 Public inheritance
22 Private inheritance
23 Protected inheritance

5.2 Pre-Lab

5.2.1 Inheritance

1 New classes created from existing classes
2 Absorb attributes and behaviors from base class
3 Classes are often closely related
   - "Factor out" common attributes and behaviors and place these in a base class
   - Use inheritance to form derived classes
4 Derived class
   - Class that inherits data members and member functions from a previously defined base class
   - Derived class *extends* the behavior of the base class.
5 The class from which another class is derived is called ***base class*** or ***parent class*** or ***super-class***
6 The class which is derived from another class is called a ***derived class*** or ***child class*** or ***subclass***
7 The derived class is a superset of the base class

5.2.2 Syntax for Inheritance

class ChildClass: *access_specifier* BaseClass
{
...
};

5.2.3 Protected access specifier

1 Intermediate level of protection between **public** and **private** inheritance
2 Protected member can't be accessed from outside the base & derived classes
3 Derived-class members can refer to **public** and **protected** members of the base class simply by using the member names

5.2.4 Example of inheritance

#include <iostream.h>

```
class A
{
private:
    int     num1;
protected:
    int     num2;
public:
    int     num3;
    void    display()
    {
            cout<<"Number 1: "<<num1<<endl;
            cout<<"Number 2: "<<num2<<endl;
            cout<<"Number 3: "<<num3<<endl;
    }
};
class B: public A
{
private:
    int     num4;
public:
    void    print()
    {
            cout<<"Number 1: "<<num1<<endl;      //error: can't access private members
            cout<<"Number 2: "<<num2<<endl;
            cout<<"Number 3: "<<num3<<endl;
            cout<<"Number 4: "<<num4<<endl;
    }
};

void main()
{
    B       obj;
    obj.print();
}
```

Q. In case of private and protected inheritance, would the public member num3 of class B be accessible to obj in main i.e. can we write obj.num3=2; ? If not, why?

5.3 In-Lab

5.3.1 Activity

Create a class called **UETPerson** that has three data members: **id, name,** and **address** of a person. Provide a 3-argument constructor to initialize the data members sent from calling function. Provide a **display** method to display person's information. Provide a **changeAdd** method to update person's

address. It takes as an argument the new address.

Derive a class **Student** from this **UETPerson** class that has two additional data members: **course** and **year** of the **student**. Provide a 5-argument constructor: two arguments to initialize the course and year of student and the other three arguments to initialize id, name, and address of student. Modify the **display** method to show all the aspects of student information.

Write all class functions outside the class. Also, write a main function to test this class.

5.3.2 Activity

Create a class called **Point** that has two data members: **x**- and **y**-coordinates of the point. Provide a no-argument and a 2-argument constructor. Provide separate get and set functions for the each of the data members i.e. **getX**, **getY**, **setX**, **setY**. The getter functions should return the corresponding values to the calling function. Provide a **display** method to display the point in (x, y) format. Make appropriate functions **const**.

Derive a class **Circle** from this **Point** class that has an additional data member: **radius** of the circle. The point from which this circle is derived represents the center of circle. Provide a no-argument constructor to initialize the radius and center coordinates to 0. Provide a 2-argument constructor: one argument to initialize the radius of circle and the other argument to initialize the center of circle (provide an object of point class in the second argument). Provide a 3-argument constructor that takes three floats to initialize the radius, x-, and y-coordinates of the circle. Provide setter and getter functions for radius of the circle. Provide two functions to determine the radius and circumference of the circle.

Write a main function to test this class.

5.3.3 Activity

Derive a class **Cylinder** from the **Circle** class of activity 1. This class contains an additional data member: height of cylinder. Provide appropriate constructors to initialize the center, radius, and height of the cylinder. Provide functions to determine the area and volume of the cylinder. Area of a cylinder is $2\pi r*r+ 2\pi r*h$. Volume of cylinder is $2\pi r*r*h$. Use the **clac_area** of circle class where required.

5.4 Home-Lab

5.3.4 Activity

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class **publication** that stores the title (a **char** string) and price (type **float**) of a publication. From this class derive two classes: **book**, which adds a page count (type **int**); and **tape**, which adds a playing time in minutes (type **float**). Each of these three classes should have a **getdata()** function to get its data from the user at the keyboard, and a **putdata()** function to display its data. Determine whether public, private, or protected inheritance should be used. Justify your answer.

Write a **main()** program to test the **book** and **tape** classes by creating instances of them, asking the user to fill in data with **getdata()** and then displaying the data with **putdata()**.

5.3.5 Activity

Start with the **publication**, **book**, and **tape** classes of activity 3. Add a base class **sales** that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a **getdata()** function to get three sales amount from the user, and a **putdata()** function to display the sales figures. Alter the **book** and **tape** classes so they are derived from both **publication** and **sales**. An object of class **book** or **tape** should input and output sales data along with its other data. Write a **main()** function to create a **book** object and a **tape** object and exercise their input/output capabilities. Determine whether **book** and **tape** classes should be publicly, privately, or protectedly inherited from **publication** and **sales** classes. Justify your answer.

5.3.6 Activity

Assume that the publisher in activity 3 decides to add a third way to distribute books: on computer disk, for those who like to do their reading on their laptop. Add a **disk** class that, like **book** and **tape**, is derived from **publication**. The **disk** class should incorporate the same member functions as the other classes. The data item unique to this class is the disk size: either 3-1/2 inches or 5-1/4 inches. You can use an **enum** Boolean type to store this item, but the complete size should be displayed. The user could select the appropriate size by typing 3 or 5.

5.5 References:

12 Class notes

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

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