**Lab 06**

**Inheritance**

1. **Objectives:**

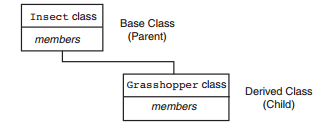
To familiarize the students with various concepts and terminologies of inheritance in object oriented Programming.

1. **Outcome:** 
   1. After this lab the students should be able to declare the derived classes along with the access of base class members.
   2. They should learn the purpose of protected members and class access as well as working with derived class constructors.
2. **Introduction:** 
   1. **Inheritance:**

Inheritance is a way of creating a new class by starting with an existing class and adding new members. It allows a new class to be based on an existing class. The new class can replace or extend the functionality of the existing class. The existing class is called the **base class** and the new class is called the **derived class**. The new class inherits all the member variables and functions (except the constructors and destructor) of the class it is based on.

* 1. **Inheritance and the “Is a” Relationship:**

When an “is a” relationship exists between classes, it means that the specialized class has all of the characteristics of the general class, plus additional characteristics that make it special. In object-oriented programming, inheritance is used to create an “is a” relationship between classes.



* 1. **Protected Members and Class Access:**

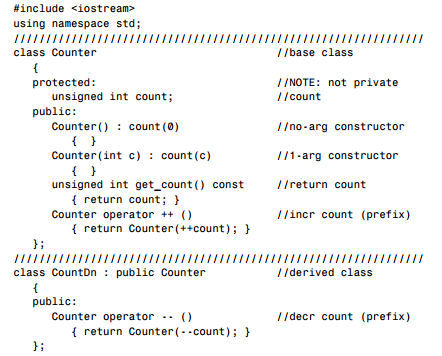
Protected members of a base class are like private members, but they may be accessed by derived classes. The base class access specification determines how private, public, and protected base class members are accessed when they are inherited by the derived classes.

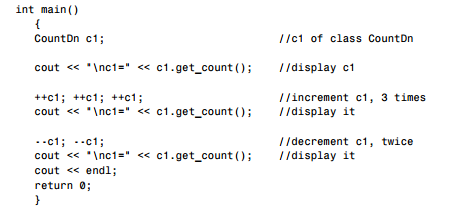
* 1. **Derived class constructor**

Constructors are not inherited, even though they have public visibility. However, the super reference can be used within the child's constructor to call the parent's constructor. In that case, the call to parent's constructor must be the first statement.

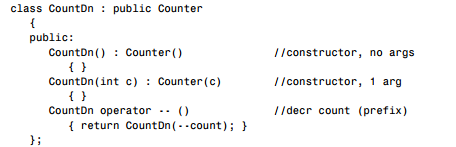


1. **Examples:**
   1. **The Counter example:**

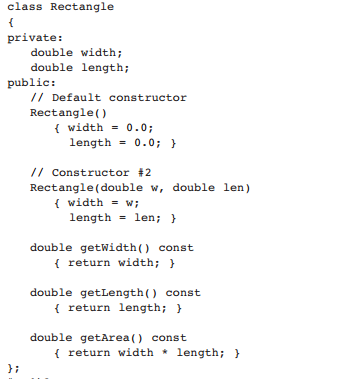


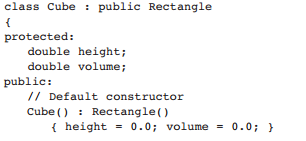


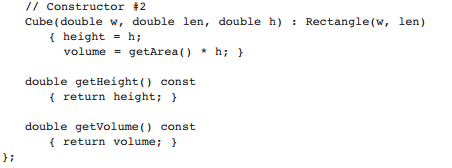
* 1. **With derived class constructor:**

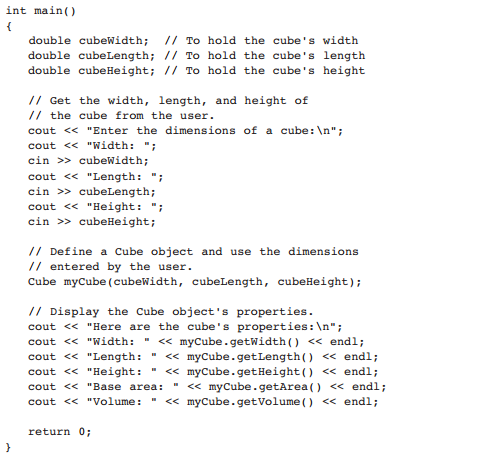


**The rectangle example:**









1. **In-Lab Tasks:**
2. Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a 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.  
   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() .
3. Assume that the publisher in task 1 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 type: either CD or DVD. You can use an enum type to store this item. The user could select the appropriate type by typing c or d.
4. Design a class namedPersonData with the following member variables:

* lastName
* firstName
* address
* city
* state
* zip
* phone

Write the appropriate accessor and mutator functions for these member variables.

Next, design a class named CustomerData, which is derived from the PersonData

class. The CustomerData class should have the following member variables:

* customerNumber
* mailingList

The customerNumber variable will be used to hold a unique integer for each customer.

The mailingList variable should be a bool. It will be set to true if the customer

wishes to be on a mailing list, or false if the customer does not wish to be on a mailing list. Write appropriate accessor and mutator functions for these member variables.

Demonstrate an object of the CustomerData class in a simple program.

1. **Post-Lab Task:**

Design a class named Employee. The class should keep the following information in  
•Employee name  
•Employee number  
•Hire date  
Write one or more constructors and the appropriate accessor and mutator functions  
for the class. Next, write a class named ProductionWorker that is derived from the Employee class.  
The ProductionWorker class should have member variables to hold the following  
information:  
•Shift (an integer)  
•Hourly pay rate (a double )  
The workday is divided into two shifts: day and night. The shift variable will hold an  
integer value representing the shift that the employee works. The day shift is shift 1, and  
the night shift is shift 2. Write one or more constructors and the appropriate accessor  
and mutator functions for the class. Demonstrate the classes by writing a program that  
uses a ProductionWorker object.

**Hint:** use **to\_string** function for converting numeric date to string.