

## Unit 3

### Lab Program

#### Inheritance

Write a C++ Program display Student Marksheet using Multiple inheritance. Here's a Simple C++ Program display Student Marksheet using Multiple inheritance in C++ Programming Language.

Write a C++ Program to illustrates the use of Constructors in multilevel inheritance. Here's a Simple Program to illustrates the use of Constructors in multilevel inheritance in C++ Programming Language.

Develop a C++ program to prepare the mark sheet of a university examination with the following items from the keyboard: Name of the student, Register no., Subject name, Subject code, Internal marks and External marks. Design a base class consisting of data members - Name of the student and Register no. The derived class consists of the data members - Subject name, Subject code, Internal marks and External marks.

An electricity board charges the following rates to domestic users to discourage large consumption of energy:

For the first 100 units – 50 P per unit

Beyond 300 units – 60 P per unit

If the total cost is more than Rs.250.00 then an additional surcharge of 15% is added on the difference. Define a class Electricity in which the function Bill computes the cost. Define a derived class More\_Electricity and override Bill to add the surcharge.

Develop a C++ program to print out the subject information using hybrid inheritance

Create two classes named Mammals and MarineAnimals. Create another class named BlueWhale which inherits both the above classes. Now, create a function in each of these classes which prints "I am mammal", "I am a marine animal" and "I belong to both the categories: Mammals as well as Marine Animals" respectively. Now, create an object for each of the above class and try calling

- function of Mammals by the object of Mammal
- function of MarineAnimal by the object of MarineAnimal
- function of BlueWhale by the object of BlueWhale
- function of each of its parent by the object of BlueWhale.

Develop two classes 'Circle' and 'Rectangle' to calculate the area of circle and area of rectangle. Inherit these two classes in a class called 'Cylinder' and calculate the volume of the cylinder by accessing the radius, length and breadth from circle and rectangle classes.

#### Virtual Functions

**1.** This problem is to get you familiar with virtual functions. Create three classes Person, Professor and Student. The class Person should have data members name and age. The classes Professor and Student should inherit from the class Person.

The class Professor should have two integer members: publications and cur\_id. There will be two member functions: getdata and putdata. The function getdata should get the input from

the user: the name, age and publications of the professor. The function putdata should print the name, age, publications and the cur\_id of the professor.

The class Student should have two data members: marks, which is an array of size and cur\_id. It has two member functions: getdata and putdata. The function getdata should get the input from the user: the name, age, and the marks of the student in subjects. The function putdata should print the name, age, sum of the marks and the cur\_id of the student. For each object being created of the Professor or the Student class, sequential id's should be assigned to them starting from.

Solve this problem using virtual functions, constructors . You can create more data members if you want.

**2. Write a c++ program to calculate the area of rectangle, circle and square using virtual function**

### **Abstract class with pure virtual functions**

1. Create a C++ program that demonstrates the use of an abstract class in Hybrid Inheritance. Define an abstract class Base with a pure virtual function display(). Derive two classes Derived1 and Derived2 from Base using single inheritance. Further, derive a class Derived3 from both Derived1 and Derived2 using multiple inheritance. Implement the display() function in all classes to print a unique message. Create objects of Derived3 and invoke the display() function.

2. Create a C++ program to implement an abstract class Vehicle with virtual functions start() and stop(). Derive two classes Car and Motorcycle from Vehicle and implement their own versions of start() and stop(). Create objects of both classes and invoke the functions.

3. Create a C++ program to implement an abstract class Shape with pure virtual functions calculate\_area() and display\_shape(). Derive two classes Rectangle and Circle from Shape and implement their versions of calculate\_area() and display\_shape(). Create objects of both classes and invoke the functions.

4. Create a C++ program to implement an abstract class Bank with a data member balance and a pure virtual function display\_balance(). Derive two classes SavingsAccount and CheckingAccount from Bank and implement their versions of display\_balance(). Set initial balances for both account types and display their balances.