**Week 6**

**Objective:** To revise inheritance and to understand the concepts of Abstract class & Interface in Java.

**Assignments:**

**1.** Design an abstract class having two methods. Create Rectangle and Triangle classes by inheriting the shape class and override the above methods to suitably implement for Rectangle and Triangle class.

**2.** Write a program in Java to illustrate the use of interface in Java.

**3.** Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume( ). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.

**4.** Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method.

**5.** Create three interfaces, each with two methods. Inherit a new interface from the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In main ( ), create an object of your class and pass it to each of the methods.

**6**. Create an interface Department containing attributes deptName and deptHead. It also has abstract methods for printing the attributes. Create a class hostel containing hostelName, hostelLocation and numberofRooms. The class contains methods for getting and printing the attributes. Then write Student class extending the Hostel class and implementing the Department interface. This class contains attributes studentName, regdNo, electiveSubject and avgMarks. Write suitable getData and printData methods for this class. Also implement the abstract methods of the Department interface. Write a driver class to test the Student class. The program should be menu driven containing the options:

i) Admit new student

ii) Migrate a student

iii) Display details of a student

For the third option a search is to be made on the basis of the entered registration number.

**7.** Create an interface called Player. The interface has an abstract method called play()that displays a message describing the meaning of “play” to the class. Create classescalled Child, Musician, and Actor that all implement Player. Create an applicationthat demonstrates the use of the classes(UsePlayer.java)

**8.** Create an abstract class Accounts with the following details:

Data Members:

(a) Balance (b) accountNumber (c) accountHoldersName (d) address

Methods:

(a) withdrawl()- abstract

(b) deposit()- abstract

(c) display() to show the balance of the account number

Create a subclass of this class SavingsAccount and add the following details:

Data Members:

(a) rateOfInterest

Methods:

(a) calculateAount()

**9.** Create an abstract class MotorVehicle with the following details:

Data Members:

(a) modelName (b)modelNumber (c) modelPrice

Methods:

(a) display() to show all the details

Create a subclass of this class Carthat inherits the class MotorVehicle and add the following details:

Data Members:

(b) discountRate

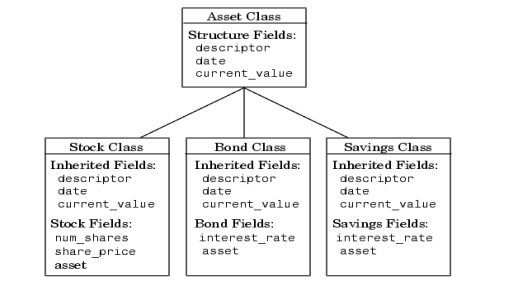
Methods:

(a) display() method to display the Car name, model number, price and the discount rate.

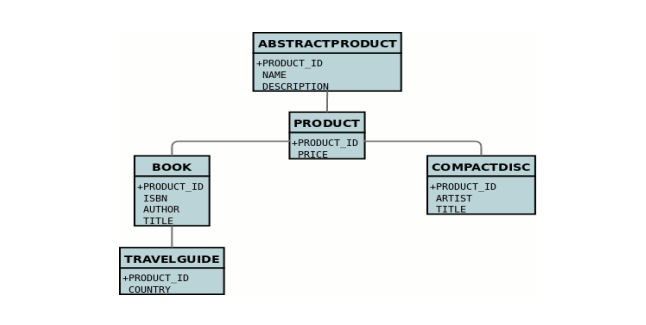
(b) discount() method to compute the discount.

**10.** Implement the below Diagram.

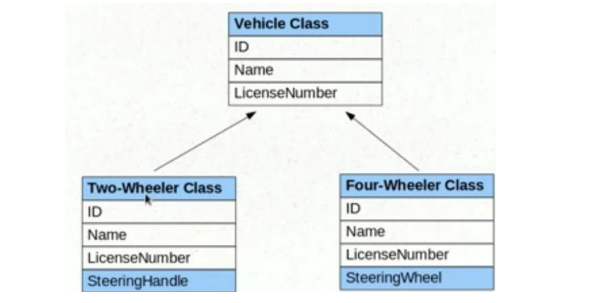
Here, Asset class is an abstract class containing an abstract method displayDetails() method. Stock, bond and Savings class inherit the Asset class and displayDetails() method is defined in every class.



**11.** Implement the below Diagram. Here AbstractProduct is only abstract class.



**12.** Implement the below Diagram



**13.** Write a program to implement the Multiple Inheritance (Bank Interface, Customer & Account

classes).

**14.** Write a program to implement the Multiple Inheritance (Gross Interface, Employee & Salary

classes).

**15.** Program to create a interface 'Mango' and implement it in classes 'Winter' and 'Summer'.

**16.** Program to implement the Multiple Inheritance (Exam Interface, Student & Result classes).

**17.** Program to demonstrate use of hierarchical inheritance using interface.

**18.** Java program to Perform Payroll Using Interface (Multiple Inheritance).

**19.** Implement the following diagram.

