1. Write a Java program which creates a base class Num and contains an integer number along with a method *shownum()* which displays the number. Now create a derived class *HexNum* which inherits Num and overrides *shownum()* which displays the hexadecimal value of the number. Demonstrate the working of the classes.
2. Write a Java program which creates a base class Num and contains an integer number along with a method *shownum()* which displays the number. Now create a derived class *OctNum* which inherits Num and overrides *shownum()* which displays the octal value of the number. Demonstrate the working of the classes.

3. Combine Question number 7 and 8 and have all the three classes together. Now describe the working of all classes.
4. Create a base class Distance which stores the distance between two locations in miles and a method *travelTime()*. The method prints the time taken to cover the distance whenthe speed is 60 miles per hour. Now in a derived class ***DistanceMKS***, override *travelTime()* so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.
5. Create a base class called "**vehicle**" that stores number of wheels and speed.
   Create the following derived classes –
   "**car**" that inherits "vehicle" and also stores number of passengers."**truck**" that inherits "vehicle" and also stores the load limit.
   Write a main function to create objects of these two derived classes and display allthe information about "car" and "truck". Also compare the speed of these two vehicles - car and truck and display which one is faster.
6. Write a program to define a class Item containing code and price. Accept this data for five objects using array of objects. Display code, price in tabular form and also, display total price of all items.
7. Write a program to define a class 'employee' with data members as empid, name and salary. Accept data for 5 objects using Array of objects and print it.
8. Define a class called circle that contains:
   a. Two private instance variables: radius (of type double) and color (of type String),
   b. Initialize the variables radius and color with default value of 1.0 and "red", respectivelyusing default constructor.
   c. Include a second constructor that will use the default value for color and sets theradius to the value passed as parameter.
   d. Two public methods: getRadius() and getArea() for returning the radius and area ofthe circle
   e. Invoke the above methods and constructors in the main.
9. Design a class named Account that contains:
   a) A private int data field named id for the account (default 0).
   b) A private double data field named balance for the account (default 0).
   c) A private double data field named annualInterestRate that stores the cur-rent interest rate (default 0). Assume all accounts have the same interest rate.
   d) A private Date data field named dateCreated that stores the date when the

account was created.

e) A no-arg constructor that creates a default account.
f) A constructor that creates an account with the specified id and initial balance.
g) The accessor and mutator methods for id,balance, and annualInterestRate.
h) The accessor method for dateCreated.
i) A method named getMonthlyInterestRate() that returns the monthly interest rate.
j) A method named getMonthlyInterest() that returns the monthly interest.
k) A method named withdraw that withdraws a specified amount from the account.
l) A method named deposit that deposits a specified amount to the account.

10. 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 theoptions:

a. Admit new student
b. Migrate a student
c. Display details of a student

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

11. 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 classes called Child, Musician, and Actor that all implement Player. Create an application
that demonstrates the use of the classes(UsePlayer.java)

12. 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()

13. 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 thefollowing details:

Data Members:

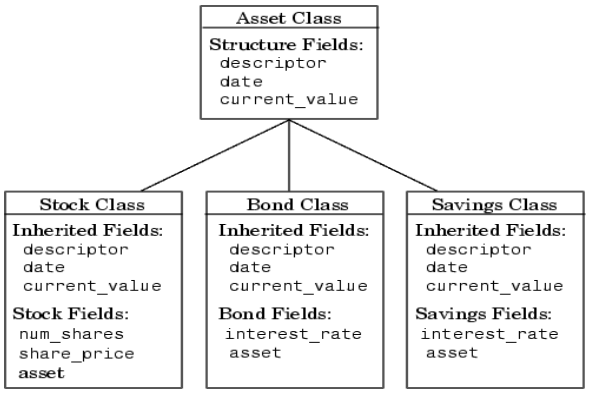
(b) discount Rate**Methods:**

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

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

14. Implement the below Diagram.

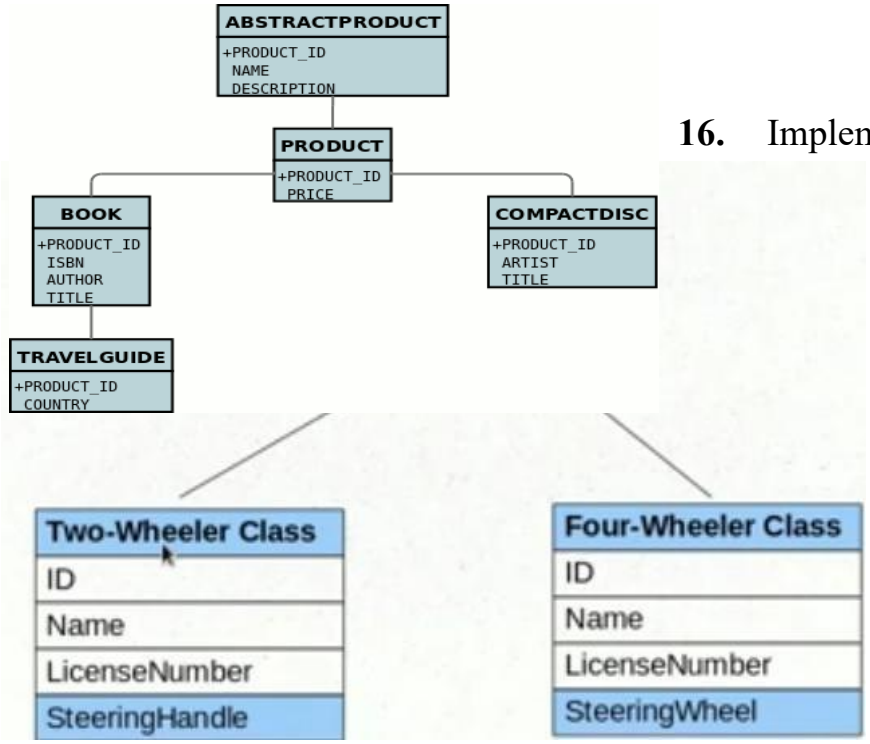
Here, Asset class is an abstract class containing an abstract method displayDetails() method. Stock, bond and Savings class inherit the Asset



class and displayDetails() method is defined in every class.

15. Implement the below Diagram. Here AbstractProduct is only abstract class.

16. Implement the below Diagram



17. Implement the following diagram