Topic: Method Overriding, Runtime Polymorphism, Abstract class, instanceof, Garbage collection

1. Write a java program that will calculate the area of Circle or Rectangle depending upon the user choice.
   Ask the user for the choice.
   1. Enter and the radius of circle
   2. Input the length and breadth of the Rectangle.
   3. Quit.

   If user enters 1, your program will take input (Enter Length:) from user and compute and display the area of square and again give him the option to continue or quit
   If user will enter 2, your program will take input (Enter Radius:) from user and compute and display the area of circle and again give him the option to continue or quit
   If user enters 3, end the program.
   If user will enter other, then 1,2 or 3 as choice your code should display wrong choice message and again give him options.

2. Create a class Car which contains members speed, noOfGear. The class has a method drive () which is responsible to provide starting speed and noOfGears to a Car. Implement display () method which will display all attributes of Car class. The class SportCar is derived from the class Car which adds new features AirBallonType. When this method is invoked, initial speed and gear status must be displayed on console. Override the display method which display all attribute of the SportCar. Make use of super class display () method.

3. Define a class named Payment that contains a member variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment.

4. Next, define a class named CashPayment that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s). Define a class named CreditCardPayment that is derived from Payment. This class

should contain member variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout.
Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

5. Create an abstract class Instrument which is having the abstract method play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message
    - "Piano is playing tan tan tan tan " for Piano class
    - "Flute is playing toot toot toot toot" for Flute class
    - "Guitar is playing tin tin tin " for Guitar class

a. You must not allow the user to declare an object of Instrument class.
b. Create an array of 10 Instruments.
c. Assign different type of instrument to Instrument reference.
d. Check for the polymorphic behaviour of play method.
e. Select the instrument randomly and call play method.
f. Use the instanceof operator to print that which object stored at which index of instrument array.

6. Create an abstract class Compartment to represent a rail coach. Provide an abstract function notice in this class. Derive FirstClass, Ladies, General, Luggage classes from the compartment class. Override the notice function in each of them to print notice suitable to the type of the compartment. Create a class TestCompartment . Write main function to do the following:
    - Declare an array of Compartment of size 10
    - Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4
    - Check the polymorphic behaviour of the notice method

7. Develop a java class that has finalize method which displays "Finalize method called". Create another class which creates objects of the

previous class and it uses the same object reference for creating these objects. For example, if Test is the class name, then the objects are created as below:

```
Test test = new Test ();

test = new Test ();

test = new Test ();
```

When the statement Runtime.getRuntime().gc() is invoked, how many times the finalize method is called?

8. Create a package called test package. Define a class called foundation inside the test package. Inside the class, you need to define 4 integer variables.
    - Var1 as private
    - Var2 as default
    - Var3 as protected
    - Var4 as public

   Import this class and packages in another class. Try to access all 4 variables of the foundation class and see what variables are accessible and what are not accessible