Inheritance

1. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call
   1 - method of parent class by object of parent class
   2 - method of child class by object of child class
   3 - method of parent class by object of child class
   Expected Output:
   This is parent class
   This is child class
   This is parent class
2. In the above example, declare the method of the parent class as private and then repeat the first two operations (You will get error in the third).
3. Create a class named 'Member' having the following members:
   Data members
   1 - Name
   2 - Age
   3 - Phone number
   4 - Address
   5 – Salary
   It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.
   Expected Output:
   ---- Employee Details ----
   Name: Alice
   Age: 28
   Phone Number: 0123456789
   Address: 123 Main Street
   Specialization: Software Development
   Salary: 50000.0

   ---- Manager Details ----
   Name: Bob
   Age: 40
   Phone Number: 0987654321
   Address: 456 Office Avenue
   Department: IT Operations
   Salary: 80000.0
4. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor

having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.

If Input:

Rectangle rect = new Rectangle(5, 3);
Square sq = new Square(4);

Expected Output:

Area of Rectangle: 15
Perimeter of Rectangle: 16
Area of Square: 16
Perimeter of Square: 16

5. Now repeat the above example to print the area of 15 squares. Hint-Use array of objects

   Expected Output:

   Square 1 - Area of Square: 1
   Square 2 - Area of Square: 4
   Square 3 - Area of Square: 9
   Square 4 - Area of Square: 16
   Square 5 - Area of Square: 25
   Square 6 - Area of Square: 36
   Square 7 - Area of Square: 49
   Square 8 - Area of Square: 64
   Square 9 - Area of Square: 81
   Square 10 - Area of Square: 100
   Square 11 - Area of Square: 121
   Square 12 - Area of Square: 144
   Square 13 - Area of Square: 169
   Square 14 - Area of Square: 196
   Square 15 - Area of Square: 225

   And

   Give any values of 15 squares through keyboard. It will show the respective areas.

6. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

   Expected Output:

   This is shape
   This is rectangular shape

7. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hierarchy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of Quadrilateral should be the *x*-*y* coordinate pairs for the four endpoints of the Quadrilateral. Write a program that instantiates objects of your classes and outputs each object's area (except Quadrilateral).

If input via constructor:
new Trapezoid(new Point(0, 0), new Point(6, 0), new Point(4, 4), new Point(2, 4));
Expected Output:
Trapezoid Area: 24.00
Parallelogram Area: 20.00
Rectangle Area: 12.00
Square Area: 4.00