Inheritance Assignment

```
/*
1. Create a class with a method that prints "This is parent class" and its subclass wi
th 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
*/
class parentClass{
    public void parentMethod(){
        System.out.println("This is parent Class");
    }
}
class childClass extends parentClass{
    public void childMethod() {
        System.out.println("This is child Class");
    }
}
class callingClass{
    public static void main(String[] args) {
        parentClass objParent = new parentClass();
        childClass objChild = new childClass();
        objParent.parentMethod();
        objChild.childMethod();
        objChild.parentMethod();
    }
}
```

```
/*
2. Create a class with a method that prints "This is parent class" and its subclass wi
th 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

In the above example, declare the method of the parent class as private and then repea
t the first two
operations (You will get error in the third).
*/
private class parentClass{
    public void parentMethod(){
        System.out.println("This is parent Class");
    }
}
class childClass extends parentClass{
    public void childMethod() {
        System.out.println("This is child Class");
    }
}
class callingPrivateClass{
    public static void main(String[] args) {
        parentClass objParent = new parentClass();
        childClass objChild = new childClass();
        objParent.parentMethod();
        objChild.childMethod();
        objChild.parentMethod();
    }
}
```

```
/*
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 b
oth of
these classes and print the same.
 */
class member{
    String name;
    int age;
    String phoneNo;
    String address;
    float salary;
    public void printSalary(){
        System.out.println("Salary is: " + salary);
    }
}
class employee extends member{
    String specialization;
    public void printSpecialization(){
        System.out.println("Specialization is: " + specialization);
    }
}
class manager extends member{
    String department;
    public void printDepartment(){
        System.out.println("Department is: " + department);
    }
}
public class employeeManagement {
    public static void main(String[] args) {
        employee objEmployee = new employee();
        objEmployee.name = "John";
        objEmployee.age = 25;
        objEmployee.phoneNo = "1234567890";
        objEmployee.address = "New York";
        objEmployee.salary = 20000;
        objEmployee.specialization = "Java";

        manager objManager = new manager();
        objManager.name = "Peter";
        objManager.age = 27;
        objManager.phoneNo = "0987654321";
        objManager.address = "New Jersey";
        objManager.salary = 30000;
        objManager.department = "Sales";

        System.out.println("\nEmployee Details:");
        System.out.println("Name: " + objEmployee.name);
        System.out.println("Age: " + objEmployee.age);
        System.out.println("Phone No: " + objEmployee.phoneNo);
        System.out.println("Address: " + objEmployee.address);
        objEmployee.printSalary();
        objEmployee.printSpecialization();

        System.out.println("\nManager Details");
        System.out.println("Name: " + objManager.name);
        System.out.println("Age: " + objManager.age);
        System.out.println("Phone No: " + objManager.phoneNo);
        System.out.println("Address: " + objManager.address);
        objManager.printSalary();
        objManager.printDepartment();
    }
}
```

```
/*
4. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and t
wo methods to
print the area and perimeter of the rectangle respectively. Its constructor having par
ameters for
length and breadth is used to initialize length and breadth of the rectangle. Let clas
s '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 rec
tangle and a square.
 */

import java.util.Scanner;

class rectangle{
    static float length;
    static float breadth;

    public rectangle(float l, float b) {
        length = l;
        breadth = b;
    }
    public static void area(){
        System.out.println("Area of rectangle is: " + (length * breadth));
    }
    public static void perimeter(){
        System.out.println("Perimeter of rectangle is: " + (2 * (length + breadth)));
    }

}
class square extends rectangle{
    static float side;

    public square(float s) {
        super(s, s);
        side = s;
    }
    public static void area(){
        System.out.println("Area of square is: " + (side * side));
    }
    public static void perimeter(){
        System.out.println("Perimeter of square is: " + (4 * side));
    }
}

public class area_parameter{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.print("\nEnter length and breadth of a rectangle: ");
        new rectangle(scan.nextFloat(), scan.nextFloat());

        rectangle.area();
        rectangle.perimeter();

        System.out.print("\nEnter side of square: ");
        new square(scan.nextFloat());

        square.area();
        square.perimeter();

        scan.close();
    }
}
```

```
/*
5. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and t
wo methods to
print the area and perimeter of the rectangle respectively. Its constructor having par
ameters for
length and breadth is used to initialize length and breadth of the rectangle. Let clas
s '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 rec
tangle and a square.
```

```
Now repeat the above example to print the area of 10 squares.
Hint-Use array of objects
 */
import java.util.Scanner;

class rectangle{
    static float length;
    static float breadth;

    public rectangle(float l, float b) {
        length = l;
        breadth = b;
    }

    public void area(){
        System.out.println("Area of rectangle is: " + (length * breadth));
    }
    public void perimeter(){
        System.out.println("Perimeter of rectangle is: " + (2 * (length + breadth)));
    }

}
class square extends rectangle{
    static float side;

    public square(float s) {
        super(s, s);
        side = s;
    }

    public void area(){
        System.out.println("Area of square is: " + (side * side));
    }
    public void perimeter(){
        System.out.println("Perimeter of square is: " + (4 * side));
    }
}

public class areaOfTenSquares {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        rectangle[] objRectangle = new rectangle[10];
        square[] objSquare = new square[10];

        for(int i = 0; i < 10; i++){
            System.out.print("\nEnter length and breadth of rectangle [" + (i+1) + "]:
");
            objRectangle[i] = new rectangle(scan.nextFloat(), scan.nextFloat());
            objRectangle[i].area();
            objRectangle[i].perimeter();

            System.out.print("\nEnter side of square [" + (i+1) + "]: ");
            objSquare[i] = new square(scan.nextFloat());
            objSquare[i].area();
            objSquare[i].perimeter();
        }
        scan.close();
    }
}
```

```
/*
6. Create a class named 'Shape' with a method to print "This is This is shape". Then c
reate two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both h
aving a method to print "This is rectangular shape" and "This is circular shape" respe
ctively. 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 'S
quare' class.
 */
class shape{
    void printShape(){
        System.out.println("This is This is shape");
    }
}
class rectangle extends shape{
    void printRectangle(){
        System.out.println("This is rectangular shape");
    }
}
class circle extends shape{
    void printCircle(){
        System.out.println("This is circular shape");
    }
}
class square extends rectangle{
    void printSquare(){
        System.out.println("Square is a rectangle");
    }
}
public class shapeCalling {
    public static void main(String[] args) {
        square objSquare = new square();
        objSquare.printShape();
        objSquare.printRectangle();
        objSquare.printSquare();
    }
}
```

```
/*
 * 7. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogr
am, 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 possibl
e. Specify the
 * instance variables and methods for each class. The private instance variables of Qu
adrilateral should
 * be the x-y coordinate pairs for the four endpoints of the Quadrilateral. Write a pr
ogram that instantiates
 * objects of your classes and outputs each object’s area (except Quadrilateral).
 */
```

```
class quadrilateral{
    int x1, y1, x2, y2, x3, y3, x4, y4;
    double l1, l2;

    public quadrilateral(double l1, double l2) {
        this.l1 = l1;
        this.l2 = l2;
    }

    public quadrilateral(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y
4){
        this.x1 = x1;
        this.y1 = y1;
        this.x2 = x2;
        this.y2 = y2;
        this.x3 = x3;
        this.y3 = y3;
        this.x4 = x4;
        this.y4 = y4;

        l1 = Math.sqrt(   ((x1-x2)*(x1-x2)) + ((y1-y2)*(y1-y2))    );
        l2 = Math.sqrt(   ((x3-x4)*(x3-x4)) + ((y3-y4)*(y3-y4))    );
    }

    public void displayArea(){
        System.out.println( (l1*l2) );
    }
}

class trapezoid extends quadrilateral{
    public trapezoid(double a1, double a2, double b){
        super(a1+a2, b);
    }
}

class parallelogram extends quadrilateral{
    public parallelogram(double base, double height){
        super(base, height);
    }
}

class rectangle extends parallelogram{
    public rectangle(double base, double height){
        super(base, height);
    }
}

class square extends rectangle{
    public square(double side){
        super(side, side);
    }
}

public class areaByInheritanceHierarchy {
    public static void main(String[] args) {
        quadrilateral objQuadrilateral = new quadrilateral(1, 2, 3, 4, 5, 6, 7, 8);
        trapezoid objTrapezoid = new trapezoid(10, 12, 7);
        parallelogram objParallelogram = new parallelogram(10, 14);
        rectangle objRectangle = new rectangle(14, 18);
        square objSquare = new square(7);

        System.out.print("The area of Quadrilateral is "); objQuadrilateral.displayAre
a();
        System.out.print("The area of Trapozoid is "); objTrapezoid.displayArea();
        System.out.print("The area of Parallelogram is "); objParallelogram.displayAre
a();
        System.out.print("The area of Rectangle is "); objRectangle.displayArea();
        System.out.print("The area of Square is "); objSquare.displayArea();

    }
}
```