1) Exercise Objective(s):*Package*

Exercise:*Create a package called shapes. Create some classes in the package representing some common geometric shapes like Square, Triangle, Circle and so on. Create a class called TestShapes and create objects for all the shapes and print corresponding messages.*

Execute the TestShapes class.

shapes

Circle.java

```
package shapes;


public class Circle {
   private double radius;

   public Circle(double radius) {
      this.radius = radius;
   }

   public double getArea() {
      return Math.PI * radius * radius;
   }

   public void display() {
      System.out.println("Circle with radius " + radius + " has area: " + getArea());
   }
}
```

Square.java

```
package shapes;

public class Square {
   private double side;

   public Square(double side) {
      this.side = side;
   }

   public double getArea() {
      return side * side;
   }

   public void display() {
      System.out.println("Square with side " + side + " has area: " + getArea());
   }
```

```
}
```

Triangle.java

```
package shapes;


public class Triangle {
   private double base;
   private double height;

   public Triangle(double base, double height) {
      this.base = base;
      this.height = height;
   }

   public double getArea() {
      return 0.5 * base * height;
   }

   public void display() {
      System.out.println("Triangle with base " + base + " and height " + height + " has area: " + getArea());
   }
}
```

TestShapes.java

```
package shapes;


public class TestShapes {
   public static void main(String[] args) {
      Square square = new Square(4);
      Triangle triangle = new Triangle(3, 5);
      Circle circle = new Circle(2.5);

      square.display();
      triangle.display();
      circle.display();
   }
}
```

OUTPUT

Square with side 4.0 has area: 16.0
Triangle with base 3.0 and height 5.0 has area: 7.5
Circle with radius 2.5 has area: 19.634954084936208

2) Exercise Objective(s):*Overloading*

Exercise:*Create a class called shape with the following methods*

1. area
2. perimeter

Overload the area and perimeter method to calculate for both square and rectangle.
Create a main class and invoke the area method to calculate the area of the square and rectangle. Also invoke the perimeter method to calculate the perimeter of the square and rectangle.

shape.java

```
package com.oops;

public class shape {
	double side;
	double length;
	double width;

	shape(double side){
		this.side=side;
	}

	shape(double length, double width) {
    this.length = length;
    this.width = width;
  }

	public double area(double side) {
		return side*side;
	}

	public double area(double length, double width) {
    return length * width;
  }

	public double perimetre(double side) {
		return 4*side;
	}

	public double perimetre(double length, double width) {
    return 2 * (length + width);
  }


	public static void main(String[] args) {
		shape s=new shape(10.0);
		System.out.println("square");
		System.out.println("area :"+s.area(s.side));
		System.out.println("perimetre :"+s.perimetre(s.side));
		System.out.println();
		shape s2=new shape(10.0,11.0);
		System.out.println("rectangle");
		System.out.println("area :"+s2.area(s2.length,s2.width));
		System.out.println("perimetre :"+s2.perimetre(s2.length,s2.width));


	}
}
```

3) Exercise Objective(s):*Overloading*

Exercise:*Create a class called Calculator which has 4 different methods add, diff, mul and div which accepts two numbers as parameters. Overload the methods such that the parameters can be of the following pattern.*

1. *Both are of int data type.*
2. *Both are of double data type.*
3. *First parameter is of int data type and second parameter is of double data type.*
4. *First parameter is of double data type and second parameter is of int data type.*

Create anobject to access these methods and invoke these methods with different type of numbers and display the result in the corresponding methods.

Calculator.java

```
package com.oops;


public class Calculator {
   public int add(int a, int b) {
      return a + b;
   }

   public double add(double a, double b) {
      return a + b;
   }

   public double add(int a, double b) {
      return a + b;
   }

   public double add(double a, int b) {
      return a + b;
   }

   public int diff(int a, int b) {
      return a - b;
   }

   public double diff(double a, double b) {
      return a - b;
   }

   public double diff(int a, double b) {
      return a - b;
   }

   public double diff(double a, int b) {
      return a - b;
   }

   public int mul(int a, int b) {
      return a * b;
   }
```

```
public double mul(double a, double b) {
    return a * b;
}

public double mul(int a, double b) {
    return a * b;
}

public double mul(double a, int b) {
    return a * b;
}

public double div(int a, int b) {
    return (double) a / b;
}

public double div(double a, double b) {
    return a / b;
}

public double div(int a, double b) {
    return a / b;
}

public double div(double a, int b) {
    return a / b;
}

public static void main(String[] args) {
    Calculator calculator = new Calculator();

    // Testing add methods
    System.out.println("Addition:");
    System.out.println("int + int: " + calculator.add(5, 3));
    System.out.println("double + double: " + calculator.add(5.5, 3.3));
    System.out.println("int + double: " + calculator.add(5, 3.3));
    System.out.println("double + int: " + calculator.add(5.5, 3));

    // Testing diff methods
    System.out.println("\nSubtraction:");
    System.out.println("int - int: " + calculator.diff(5, 3));
    System.out.println("double - double: " + calculator.diff(5.5, 3.3));
    System.out.println("int - double: " + calculator.diff(5, 3.3));
    System.out.println("double - int: " + calculator.diff(5.5, 3));

    // Testing mul methods
    System.out.println("\nMultiplication:");
    System.out.println("int * int: " + calculator.mul(5, 3));
    System.out.println("double * double: " + calculator.mul(5.5, 3.3));
    System.out.println("int * double: " + calculator.mul(5, 3.3));
    System.out.println("double * int: " + calculator.mul(5.5, 3));

    // Testing div methods
    System.out.println("\nDivision:");
    System.out.println("int / int: " + calculator.div(6, 3));
    System.out.println("double / double: " + calculator.div(6.6, 3.3));
    System.out.println("int / double: " + calculator.div(6, 3.3));
    System.out.println("double / int: " + calculator.div(6.6, 3));
```

```
    }
}
```

4) Exercise Objective(s): *The concept of inheritance*

Exercise: *Create a class called Vehicle. Create subclasses like Truck, Bus, Car etc. Add common methods in the base class and specific methods in the corresponding class. Create a class called Road and create objects for the Truck, Car, Bus etc and display the appropriate message.*

```
package com.oops;

 class Vehicle {
        String name;
        int wheels;
         Vehicle(String name,int wheels){
                this.name=name;
                this.wheels=wheels;
        }
          void displayInfo() {
                 System.out.println("Vehicle Name: " + name + ", Wheels: " + wheels);
         }


}
class Truck extends Vehicle{
        int loadcapacity;
        Truck(String name,int wheels,int loadcapacity){
                super(name,wheels);
                this.loadcapacity=loadcapacity;
        }
        void displayInfo() {
     super.displayInfo();
     System.out.println("Load Capacity: " + loadcapacity + " tons");
   }

}

class Bus extends Vehicle {
   int passengerCapacity;

   Bus(String name, int wheels, int passengerCapacity) {
      super(name, wheels);
      this.passengerCapacity = passengerCapacity;
   }

   void displayInfo() {
      super.displayInfo();
      System.out.println("Passenger Capacity: " + passengerCapacity + " people");
   }
}
 public class Road {
   public static void main(String[] args) {
      Truck truck = new Truck("Truck", 6, 10);
      Bus bus = new Bus("Bus", 4, 50);
```

```
    truck.displayInfo();
    System.out.println();
    bus.displayInfo();
    System.out.println();
  }
}
```