[[[1]]] // Rectangle

```
public class Rectangle {
        double width = 1 ;
        double height = 1 ;

        Rectangle() {

        }
        Rectangle(double newWidth , double newHeight) {
                width = newWidth ;
                height = newHeight ;
        }
        public double getArea() {
                return width * height ;
        }
        public double getPerimeter () {
                return 2 * (width + height) ;
        }
}

public class TestRectangle {

        public static void main(String[] args) {

                Rectangle r1 = new Rectangle();
                Rectangle r2 = new Rectangle(3.5,35.9);
                r1.width = 4 ;
                r1.height = 40 ;

                System.out.println("Area of rectangle 1 is :" +r1.getArea()+
"\nPerimeter of rectangle 1 : " +r1.getPerimeter());
                System.out.println("\nArea of rectangle 2 is :" +r2.getArea()+
"\nPerimeter of rectangle 2 : " +r2.getPerimeter());
        }
}
```

[[[2]]] // Stock change rate

```
public class Stock {
        String symbol ;
        String name ;
        double previousClosingPrice ;
        double currentPrice ;

        Stock(String newSymbol , String newName) {
                symbol = newSymbol ;
                name = newName ;
        }
        public double getChangePercent() {
                return (Math.abs(currentPrice -
```

```
previousClosingPrice)*previousClosingPrice/100.0) ;
        }
}

public class TestStock {
        public static void main(String[] args) {
                Stock s = new Stock("ORCL","Orcale Corporation");
                s.previousClosingPrice = 34.5 ;
                s.currentPrice = 34.35 ;

                System.out.println("Corporation name  : " +s.name);
                System.out.println("Corporation symbol : " +s.symbol);
                System.out.println("Percentage chnage : " +s.getChangePercent());
        }
}
[[[3]]] // Account

public class Account {

        private int id = 0;
        private double balance = 0 ;
        private double annualInterestRate = 0 ;
        private java.util.Date dateCreated = new java.util.Date();

        public Account() {
                dateCreated = new java.util.Date() ;
        }
        public Account(int newId , double newBalance) {
                id = newId ;
                balance = newBalance ;
        }
        public int getId() {
                return id;
        }
        public void setId(int newId) {
                id = newId;
        }
        public double getBalance() {
                return balance;
        }
        public void setBalance(double newBalance) {
                balance = newBalance;
        }
        public double getAnnualInterestRate() {
                return annualInterestRate;
        }
        public void setAnnualInterestRate(double newAnnualInterestRate) {
                annualInterestRate = newAnnualInterestRate;
        }
        public double getMonthlyInterestRate() {
```

```
                return (annualInterestRate / 100.0) / 12.0 ;
        }
        public double getMonthlyInterest () {
                return balance * getMonthlyInterestRate() ;
        }
        public void withdraw (double amount) {
                balance-= amount ;
        }
        public void deposit (double amount) {
                balance+= amount ;
        }
        public String getDateCreated () {
                return dateCreated.toString();
        }
}

public class TestAccount {
        public static void main(String[] args) {
                Account account = new Account(1122, 20000);
        account.setAnnualInterestRate(4.5);
        account.withdraw(2500.0);
        account.deposit(3000.0);

        System.out.println("Balance: $" + account.getBalance());
        System.out.println("Monthly Interest: " + account.getMonthlyInterest());
        System.out.println("Date created : " + account.getDateCreated());
        }
}
[[[4]]] // Fan

public class Fan {
        public final int SLOW ;
        public final int MEDIUM  ;
        public final int FAST ;

        private int speed ;
        private boolean on ;
        private double radius ;
    public String color ;

        public Fan() {
                SLOW = 1 ;
                MEDIUM =2 ;
                FAST = 3 ;
                speed = SLOW ;
                on = false ;
                radius = 5 ;
                color = "blue" ;
        }
        public int getSpeed() {
```

```
                return speed;
        }
        public void setSpeed(int neWspeed) {
                speed = neWspeed;
        }
        public boolean isOn() {
                return on;
        }
        public void setOn(boolean neWon) {
                on = neWon;
        }
        public double getRadius() {
                return radius;
        }
        public void setRadius(double neWradius) {
                radius = neWradius;
        }
        public String getColor() {
                return color;
        }
        public void setColor(String neWcolor) {
                color = neWcolor;
        }
        public String toString () {
                if(on) {
                        return "Fan [ Speed : "+speed+ ", Radius : "+radius+ ",
Color : "+color+ " ]" ;
                }
                else {
                return " Fan [ radius : " +radius+ " Color : " +color+ " ] the fan
is off " ;
                }
        }
}

public class TestFan {
        public static void main(String[] args) {

                Fan fan1 = new Fan();
                fan1.setSpeed(3);
                fan1.setRadius(10);
                fan1.setColor ("yellow");
                fan1.setOn(true);

                Fan fan2 =new Fan();
                fan2.setRadius(5);
                fan2.setColor ("blue");
                fan2.setSpeed(2);
                fan2.setOn(false);
```

```
                System.out.println(fan1.toString());
                System.out.println(fan2.toString());
        }
}
[[[ Location ]]]

public class Location {
        // Data fields
        int row;                // Row index of maximal value
        int column;                     // Column index of maximal value
        double maxValue;        // Maximal value

        /** Constructs a default Loction object */
        Location(double[][] a) {
                maxValue = a[0][0];
                row = 0;
                column = 0;
                for (int i = 0; i < a.length; i++) {
                        for (int j = 0; j < a[i].length; j++) {
                                if (a[i][j] > maxValue) {
                                        maxValue = a[i][j];
                                        row = i;
                                        column = j;
                                }
                        }
                }
        }
}
import java.util.Scanner;

public class TestLocation {
        public static void main(String[] args) {

                Scanner input = new Scanner(System.in);

                System.out.print("Enter the number of rows and columns in the array:
");
                int rows = input.nextInt();
                int columns = input.nextInt();

                double[][] array = new double[rows][columns];
                System.out.println("Enter the array: ");
                for (int i = 0; i < array.length; i++) {
                        for (int j = 0; j < array[i].length; j++) {
                                array[i][j] = input.nextDouble();
                        }
                }
                Location max = locateLargest(array);

                System.out.println("The location of the largest element is " +
```

```
                        max.maxValue + " at (" + max.row + ", " + max.column + ")");
        }
        public static Location locateLargest(double[][] a) {
                return new Location(a);
        }
}
```