Exercise 1:
Define a class Point in a file called Point.java with three attributes name of type String, x and y of type double representing Cartesian coordinates of a geometric point.

• Define the following public methods for the class Point:

1) A constructor Point() without parameter, giving the following default values to the attributes x=0 and y=0.

2) A constructor Point(double xValue, double yValue) with parameters that provide the values for all the attributes.

3) Methods (setters, getters) allowing to access to attributes.

4) A method translate(double dx, double dy) that moves the point by the amount (dx, dy).

5) A method double distance(Point p) that accepts an object of type Point, calculates the distance between the current Point object and the received one p, and returns this distance.

6) A method display() that prints the point to the screen in a suitable form.

• Write a main program with class TestPoint.java where you test the class Point.

• In this program declare two objects p1 and p2 of the class Point.

• Use the first constructor for the first point p1. Then set this point p1 to a user input values.

• Use the second constructor for point p2 with x=1 and y=3, move this point by dx=2 and dy=-4, then calculate the distance between this point p2 and point p1.

• Print the distance between these two points as following:

  - i. “The distance between the two points Point name1(x1,y1), Point name2(x2,y2) = z”. For example: “The distance between Point m(3,0), Point n(0,0) = 3”.
Solution:
Class Point:

```java
public class Point {

    private double x;
    private double y;
    public Point(){
        x=0;
        y=0;
    }
    public Point(double xValue, double yvalue){
        x = xValue;
        y = yvalue;
    }
    public double getX(){
        return x;
    }
    public double getY(){
        return y;
    }
    public void setX(double xValue){
        x = xValue;
    }
    public void setY(double yValue){
        y = yValue;
    }
    public void translate(double dx, double dy){
        x += dx;  // same as x = x + dx;
        y += dy;  // same as y = y + dy;
    }
    public double distance(Point p){
        double distance = Math.sqrt( (x-p.x)*(x-p.x) + (y-p.y)*(y-p.y));
        return distance;
    }
    public void display(){
        System.out.print("Point"+"("+x+","+y+")");
    }
    public double distance1(Point p){
        double distance = Math.sqrt( (x-p.x)*(x-p.x) + (y-p.y)*(y-p.y));
        return distance;
    }
}
```
Class TestPoint:

```java
import java.util.Scanner;

public class TestPoint {

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

        Point p1 = new Point();
        System.out.println("Enter x for p1:");
        p1.setX(S.nextDouble());
        System.out.println("Enter y for p1:");
        p1.setY(S.nextDouble());

        Point p2 = new Point(1, 3);
        p2.translate(2, -4);

        double dist = p1.distance(p2);

        System.out.print("The distance between the two points ");
        p1.display();
        System.out.print(", ");
        p2.display();
        System.out.print(" = "+dist);
    }
}
```
Exercise 2:
Define a class Date in a file called Date.java with three attributes: day, month and year all of type int.
• Define the following public methods for the class Date:
  1) A constructor Date() without parameter, giving reasonable default values to the attributes.
  2) A constructor Date(int d, int m, int y) with parameters that provide the values for all the attributes.
  3) Methods (setters, getters) allowing to access to attributes.
  4) A method increment() that adds one day to the current date.
  5) A method decrement() that subtracts one day from the current date.
  6) Write a method display() that prints the date to the screen in a suitable form.

• Write a main program with a class TestDate.java where you test the class Date.
• Declare two objects d1 and d2 of the class Date, using for each of them one of the two constructors of the class.
• Increment the first date object by one day.
• Decrement the second date object by one day.
• Display each of the two date objects.
(Hint: to avoid all the difficulties with date calculations assume that each month has exactly 30 days).

Solution:
Class Date:

```java
public class Date {
    private int year;
    private int month;
    private int day;

    public Date() {
        year = 2000;
        month = 1;
        day = 1;
    }

    public Date(int y, int m, int d) {
        if(y>=0)
            year = y;
        else year = 2000;

        if(m>0 && m<=12)
            month = m;
        else month = 1;

        if(d>0 && d<=30)
            day = d;
        else day = 1;
    }
}
```
public int getYear()
{
    return year;
}

public void setYear(int y)
{
    if(y>=0)
    {
        year = y;
    }
    else
    {
        year = 2000;
    }
}

public int getMonth()
{
    return month;
}

public void setMonth(int m)
{
    if(m>0 & & m<=12)
    {
        month = m;
    }
    else
    {
        month = 1;
    }
}

public int getDay()
{
    return day;
}

public void setDay(int d)
{
    if(d>0 & & d<=30)
    {
        day = d;
    }
    else
    {
        day = 1;
    }
}

public void display()
{
    System.out.println(day+"/"+month+"/"+year);
}

public void increment()
{
    day++;  
    if(day == 31)
    {
        day = 1;
        month ++;
    }
    if(month == 13)
    {
        month = 1;
        year ++;
    }
}

public void decrement()
{
    day --;
    if(day == 0)
    {
        day = 30;
        month --;
    }
    if(month == 0)
    {
        month = 12;
        year --;
    }
} //end of if
} //end of method
} //end of class
Class TestDate:

import java.util.Scanner;

public class TestDate {

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

        System.out.print("Enter year of date1: ");
        int year = S.nextInt();

        System.out.print("Enter month of date1: ");
        int month = S.nextInt();

        System.out.print("Enter day of date1: ");
        int day = S.nextInt();

        Date d1 = new Date(year, month, day);
        Date d2 = new Date();
        d1.increment();
        d2.decrement();
        System.out.print("the value of date1 after incrementing by one is ");
        d1.display();
        System.out.print("the value of date2 after decrementing by one is ");
        d2.display();
    }
}

}