VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

Assessment – III

Course Name & Code: Programming in JAVA & CSE1007 Max. Marks: 10

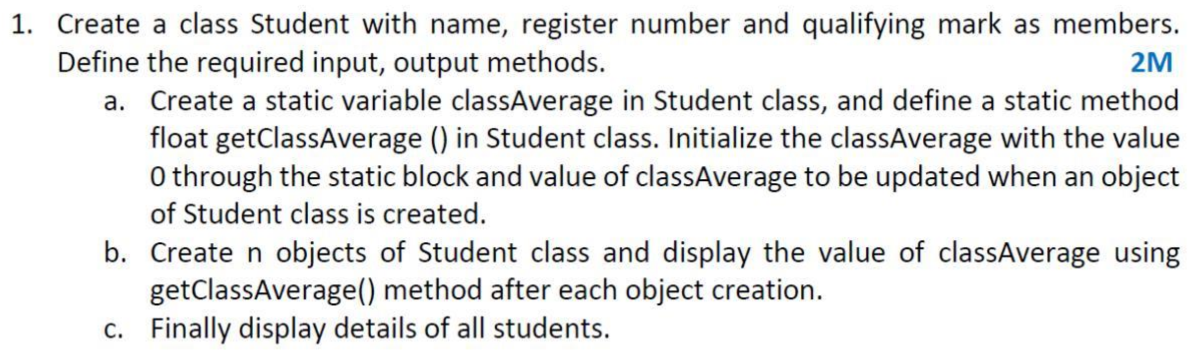
Semester: Winter 2019-20 Slot: L55 + L56

Submission Date: 28 – 02 – 2020

NAME: ALOK SINHA

REG NO:17BCE2380

SLOT: L55+L56



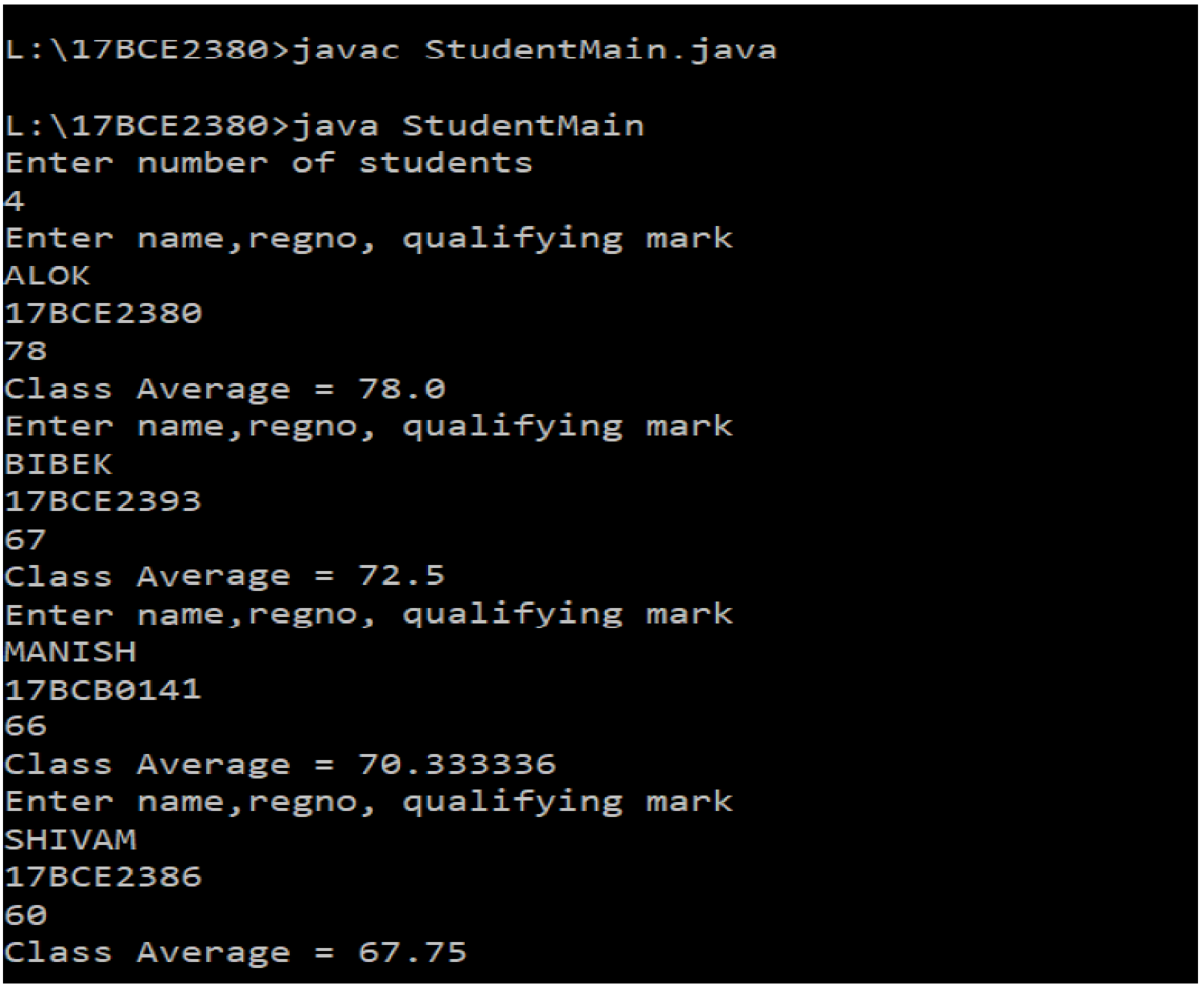
1. Create a class Student with name, register number and qualifying mark as members. Define the required input, output methods. 2M
   a. Create a static variable classAverage in Student class, and define a static method float getClassAverage () in Student class. Initialize the classAverage with the value 0 through the static block and value of classAverage to be updated when an object of Student class is created.
   b. Create n objects of Student class and display the value of classAverage using getClassAverage() method after each object creation.
   c. Finally display details of all students.

CODE:

```
import java.util.*;
import java.lang.*;
class Student{
        String name,regno;
        int qm;
        static float classAvg;
        static int ctr;
        static{
                classAvg = 0f;
        }
        public void input(){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter name,regno, qualifying
                mark"); name = sc.next();
                regno = sc.next();
                qm = sc.nextInt();
                classAvg+=qm;
                ctr++;
        }
        public static float getClassAverage(){
                return classAvg/ctr;
        }
        public void display(){
                System.out.println("Name :"+ name );
                System.out.println(" Register No. :"+ regno );
                System.out.println(" Qualifying mark :"+ qm );
                System.out.println("\n");
        }
}
class StudentMain{
        public static void main(String args[]){ Scanner sc =
                new Scanner(System.in);
                System.out.println("Enter number of
                students"); int n = sc.nextInt();
                float avg;
                Student arr[] = new Student[n];
                for(int i=0;i<n;i++){
                        arr[i] = new Student();
                        arr[i].input();
                        avg = arr[i].getClassAverage();
                        System.out.println("Class Average = " + avg);
                }
                System.out.println("\nDisplaying Student Details\n");
                for(int i=0;i<n;i++){
                        arr[i].display();
                }
        }
}
```

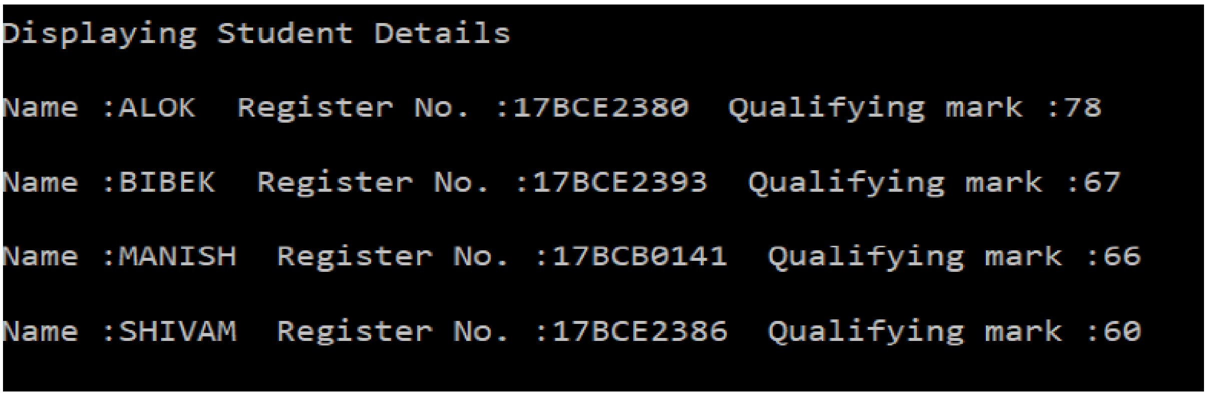
OUTPUT

FIG 1:

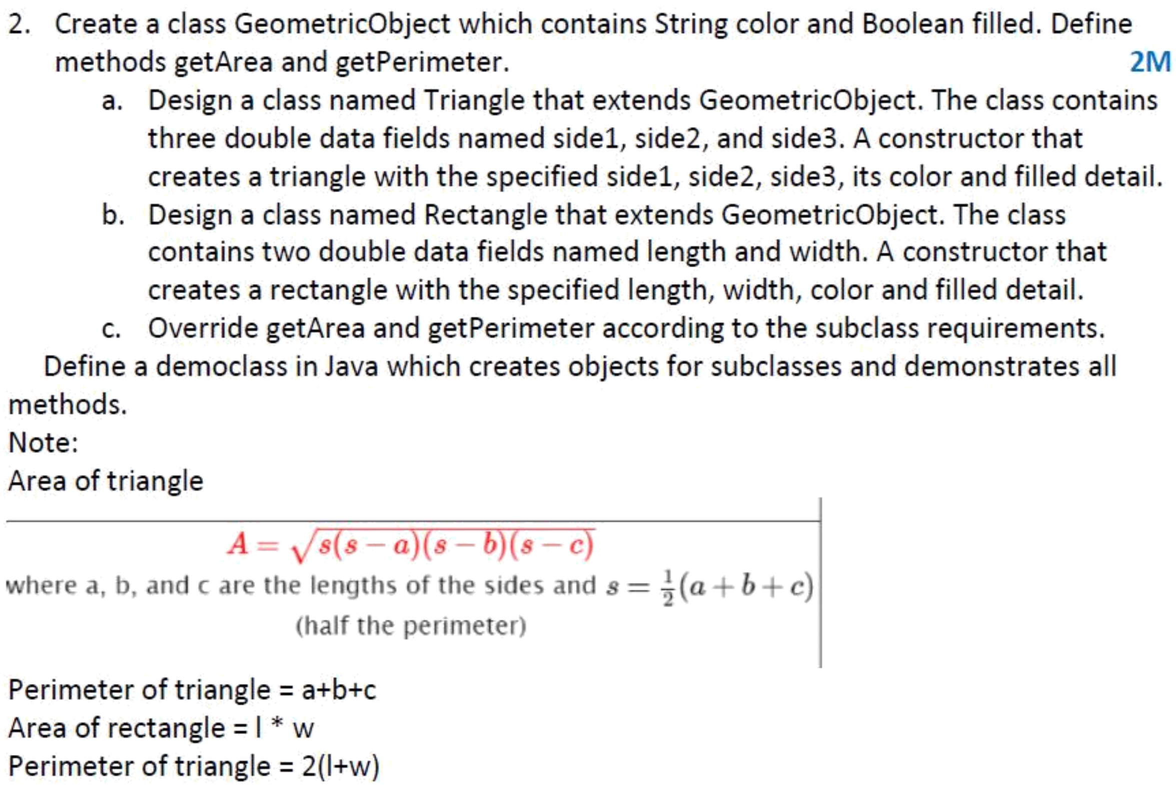


```
L:\17BCE2380>javac StudentMain.java

L:\17BCE2380>java StudentMain
Enter number of students
4
Enter name,regno, qualifying mark
ALOK
17BCE2380
78
Class Average = 78.0
Enter name,regno, qualifying mark
BIBEK
17BCE2393
67
Class Average = 72.5
Enter name,regno, qualifying mark
MANISH
17BCB0141
66
Class Average = 70.333336
Enter name,regno, qualifying mark
SHIVAM
17BCE2386
60
Class Average = 67.75
```

FIG 1:



```
Displaying Student Details

Name :ALOK  Register No. :17BCE2380  Qualifying mark :78

Name :BIBEK  Register No. :17BCE2393  Qualifying mark :67

Name :MANISH  Register No. :17BCB0141  Qualifying mark :66

Name :SHIVAM  Register No. :17BCE2386  Qualifying mark :60
```



2. Create a class GeometricObject which contains String color and Boolean filled. Define methods getArea and getPerimeter. **2M**
   a. Design a class named Triangle that extends GeometricObject. The class contains three double data fields named side1, side2, and side3. A constructor that creates a triangle with the specified side1, side2, side3, its color and filled detail.
   b. Design a class named Rectangle that extends GeometricObject. The class contains two double data fields named length and width. A constructor that creates a rectangle with the specified length, width, color and filled detail.
   c. Override getArea and getPerimeter according to the subclass requirements.

Define a democlass in Java which creates objects for subclasses and demonstrates all methods.

Note:

Area of triangle

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

where a, b, and c are the lengths of the sides and $s = \frac{1}{2}(a+b+c)$

(half the perimeter)

Perimeter of triangle = a+b+c

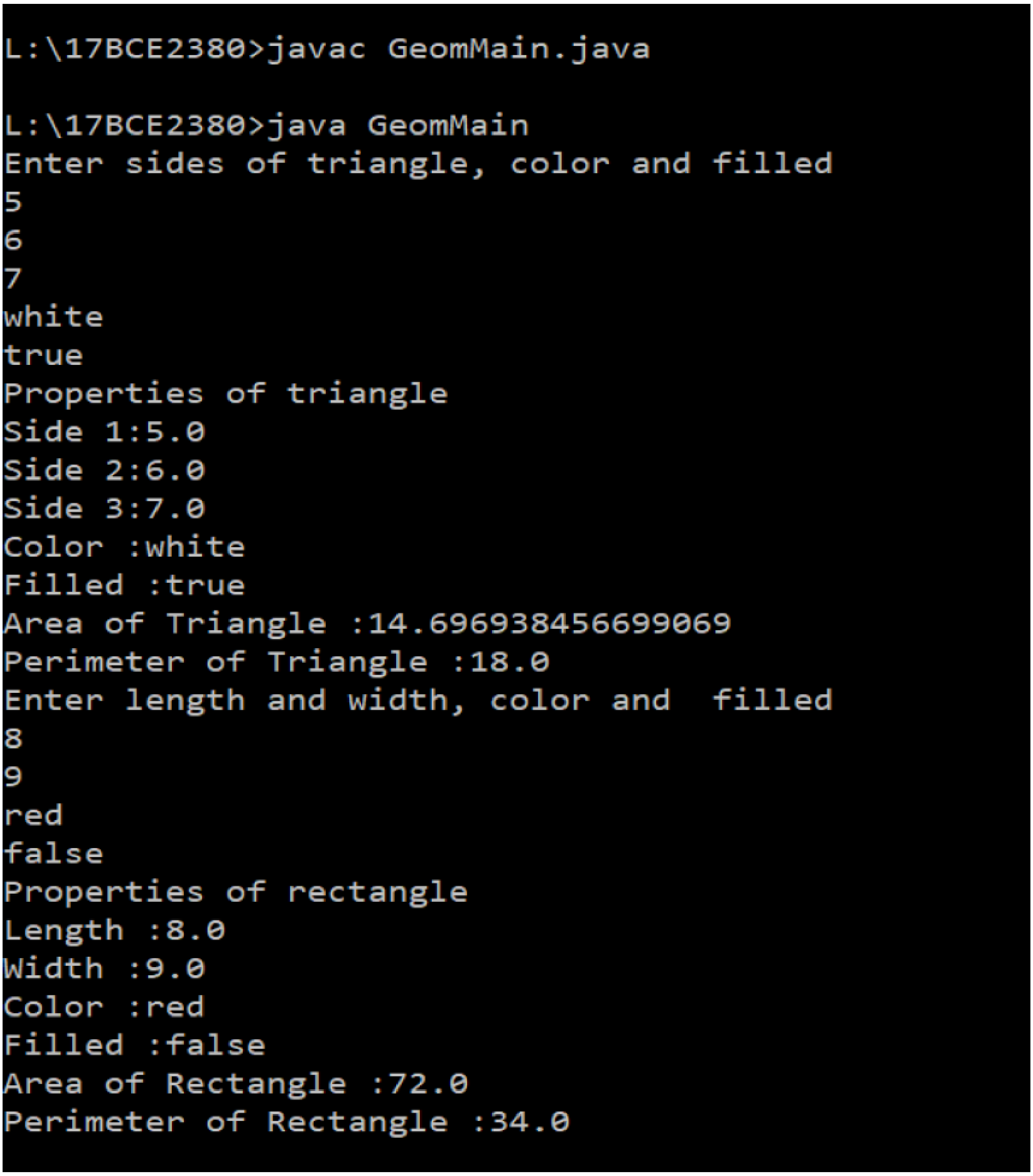
Area of rectangle = l * w

Perimeter of triangle = 2(l+w)

CODE

```
import java.util.*;
import java.lang.*;
class GeometricObject{
        String color;
        boolean filled;
        void getArea(){}
        void getPerimeter(){}
}
class Triangle extends GeometricObject{
        double s1,s2,s3;
        Triangle(double s1,double s2,double s3,String color,boolean
                filled){ this.s1 = s1;
                this.s2 = s2;
                this.s3 = s3;
                this.color = color;
                this.filled = filled;
        }
        void getArea(){
                double s = (s1+s2+s3)/2;
        double Area = Math.sqrt(s*(s-s1)*(s-s2)*(s-s3));
        System.out.println("Area of Triangle :"+Area );
    }
    void getPerimeter(){
        System.out.println("Perimeter of Triangle :"+ (s1+s2+s3) );
    }
    void display(){
        System.out.println("Side 1:"+s1);
        System.out.println("Side 2:"+s2);
        System.out.println("Side 3:"+s3);
        System.out.println("Color :"+color);
        System.out.println("Filled :"+filled);
    }
}
class Rectangle extends
    GeometricObject{ double l,w;
    Rectangle(double l,double w,String color,boolean filled){
        this.l = l;
        this.w = w;
        this.color = color;
        this.filled = filled;
    }
    void getArea(){
        System.out.println("Area of Rectangle :"+ (l*w) );
    }
    void getPerimeter(){
        System.out.println("Perimeter of Rectangle :"+ ( 2 * (l+w) ) );
    }
    void display(){
        System.out.println("Length :"+l);
        System.out.println("Width :"+w);
        System.out.println("Color :"+color);
        System.out.println("Filled :"+filled);
    }
}
class GeomMain{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter sides of triangle, color and filled");
        double s1,s2,s3;
        s1=sc.nextDouble();
        s2=sc.nextDouble();
        s3=sc.nextDouble();
        String Tcolor; boolean Tfilled;
        Tcolor = sc.next();
        Tfilled = sc.nextBoolean();
        Triangle t = new Triangle(s1,s2,s3,Tcolor,Tfilled);
        System.out.println("Properties of triangle");
        t.display();
        t.getArea();
        t.getPerimeter();
        System.out.println("Enter length and width, color and filled");
        double l,w;
        l=sc.nextDouble();
        w=sc.nextDouble();
        String Rcolor; boolean Rfilled;
        Rcolor = sc.next();
        Rfilled = sc.nextBoolean();
        Rectangle r = new Rectangle(l,w,Rcolor,Rfilled);
        System.out.println("Properties of rectangle");
        r.display();
        r.getArea();
        r.getPerimeter();
    }
}
```

OUTPUT



```
L:\17BCE2380>javac GeomMain.java

L:\17BCE2380>java GeomMain
Enter sides of triangle, color and filled
5
6
7
white
true
Properties of triangle
Side 1:5.0
Side 2:6.0
Side 3:7.0
Color :white
Filled :true
Area of Triangle :14.696938456699069
Perimeter of Triangle :18.0
Enter length and width, color and  filled
8
9
red
false
Properties of rectangle
Length :8.0
Width :9.0
Color :red
Filled :false
Area of Rectangle :72.0
Perimeter of Rectangle :34.0
```

3. Define a package to implement the following: **1M**

a. Create a class called as Coordinate which stores x and y coordinate values.

b. Develop constructors to initiate the coordinates based on the parameter passed.

c. Design a method distance() to find the distance between two coordinates.

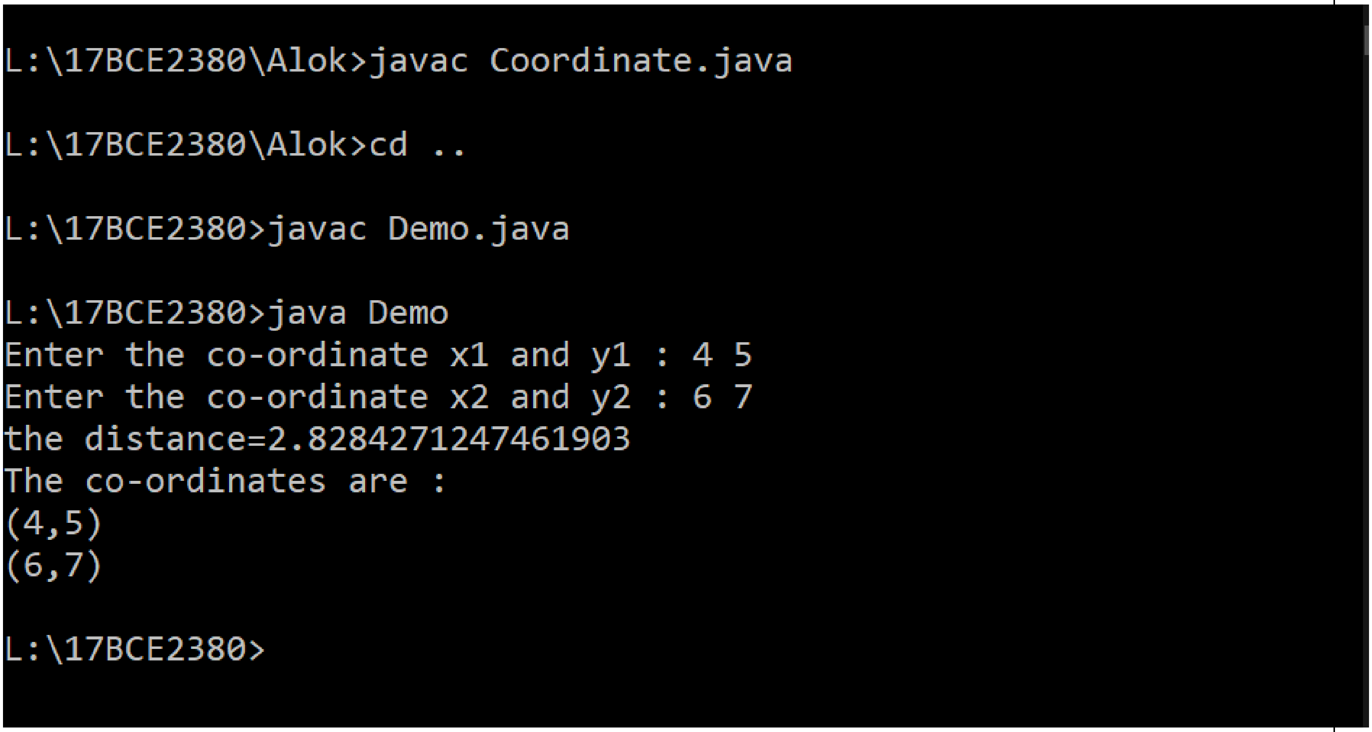
d. Design method print() to print the members of the Coordinate class in the “(x,y)” format.

e. Develop a demo class to test above processes

CODE

```
// CREATION OF PACKAGE
package Alok;
import java.util.*;
import java.lang.*;
public class Coordinate{
        int x1,x2,y1,y2;
public Coordinate(int x1,int y1,int x2,int y2){
        this.x1=x1;
        this.x2=x2;
        this.y1=y1;
        this.y2=y2;
}

public void Distance() {
  double d=Math.pow(x2-x1, 2)+Math.pow(y2-y1, 2);
  double s=Math.sqrt(d);
  System.out.println("the distance="+s);
}
public void print() {
        System.out.println("The co-ordinates are : ");
        System.out.println("("+x1+","+y1+")");
        System.out.println("("+x2+","+y2+")");
}
public static void main(String args[]) {

}
}
//DEMO CLASS FOR CHECKING THE PACKAGES


import java.util.Scanner;
import Alok.Coordinate;
public class Demo {
public static void main(String args[]) {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the co-ordinate x1 and y1 : ");
        int a=sc.nextInt();
        int b=sc.nextInt();
        System.out.print("Enter the co-ordinate x2 and y2 : ");
        int c=sc.nextInt();
        int d=sc.nextInt();
		Coordinate obj1=new Coordinate(a,b,c,d);
		obj1.Distance();
		obj1.print();
}
}
```

OUTPUT



```
L:\17BCE2380\Alok>javac Coordinate.java

L:\17BCE2380\Alok>cd ..

L:\17BCE2380>javac Demo.java

L:\17BCE2380>java Demo
Enter the co-ordinate x1 and y1 : 4 5
Enter the co-ordinate x2 and y2 : 6 7
the distance=2.8284271247461903
The co-ordinates are :
(4,5)
(6,7)

L:\17BCE2380>
```

4. Write a program in Java to raise exception for data validation and typo error. **1M**
   a. Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException.
   b. If the Mobile Number contains any character other than a digit, raise a NumberFormatException.
   c. If the Register Number contains any character other than digits and alphabets, throw a NoSuchEl

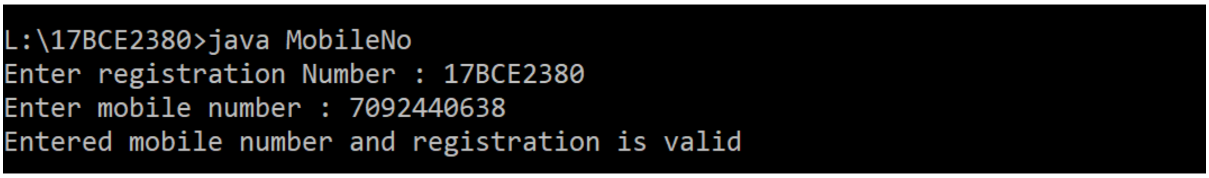
CODE

```
import java.util.*;
public class MobileNo{
        public static Boolean isAlphaNumeric(String str) {
                for(int i=0;i<str.length();i++) {
                        char c=str.charAt(i);
                        if(!(c>='A' && c<='Z') &&
                                !(c>='a' && c<='z') &&
                                !(c>='0' && c<='9')) {
                                return Boolean.FALSE;
                        }

                }
                return Boolean.TRUE;
        }
public static Boolean isDigit(String st) {
                for(int i=0;i<st.length();i++) {
                        char ch=st.charAt(i);
                        if(!(ch>='0' && ch<='9')){
                                return Boolean.FALSE;
                        }

                }
                return Boolean.TRUE;
        }

public static void main(String args[]) {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter registration Number : ");
        String reg=sc.next();
        sc.nextLine();
        System.out.print("Enter mobile number : ");
        String mob=sc.next();

        if((mob.length() !=10) || (reg.length() !=9)) {
                  throw new IllegalArgumentException("enter 10 digit valid mobile number or 9 digit
reg number") ;
        }

     if(!isDigit(mob)){
      throw new NumberFormatException("entered mobile number is not numeric");
     }
```

```
            if(!isAlphaNumeric(reg)) {
                        throw new NoSuchElementException("register number must contain only
alphaNumeric");
            }

           System.out.println("Entered mobile number and registration is valid");

}
}
```

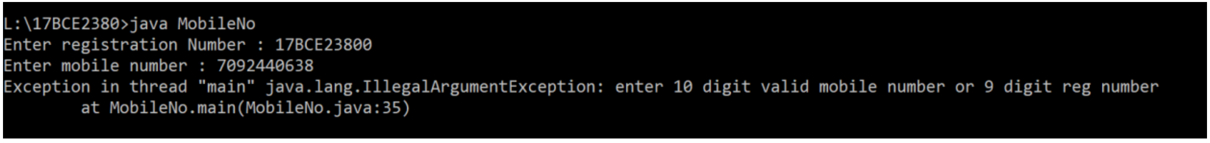
OUTPUT

CASE -1: Valid Details



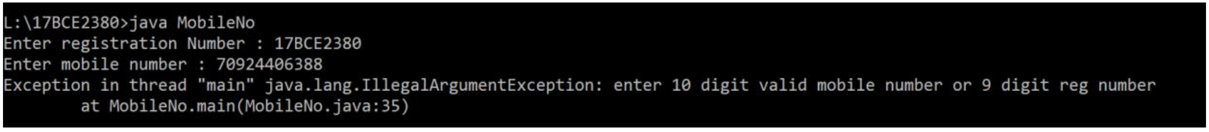
```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE2380
Enter mobile number : 7092440638
Entered mobile number and registration is valid
```

CASE-2: Entered reg no. is more than 9 digits.



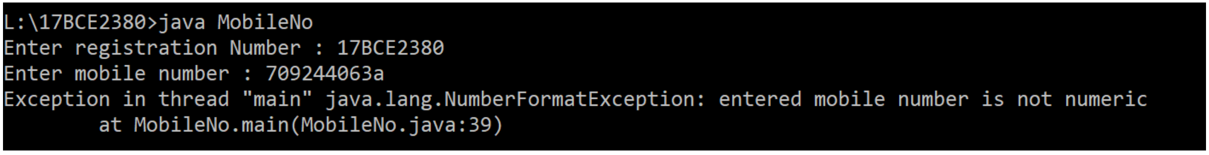
```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE23800
Enter mobile number : 7092440638
Exception in thread "main" java.lang.IllegalArgumentException: enter 10 digit valid mobile number or 9 digit reg number
        at MobileNo.main(MobileNo.java:35)
```

CASE-3: Entered mobile no. is more than 10 digits.



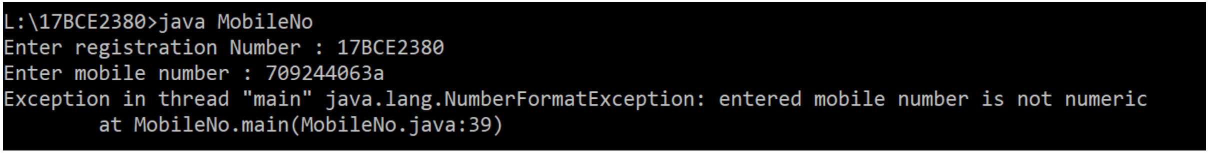
```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE2380
Enter mobile number : 70924406388
Exception in thread "main" java.lang.IllegalArgumentException: enter 10 digit valid mobile number or 9 digit reg number
        at MobileNo.main(MobileNo.java:35)
```

CASE-4: Mobile no. containing special character.



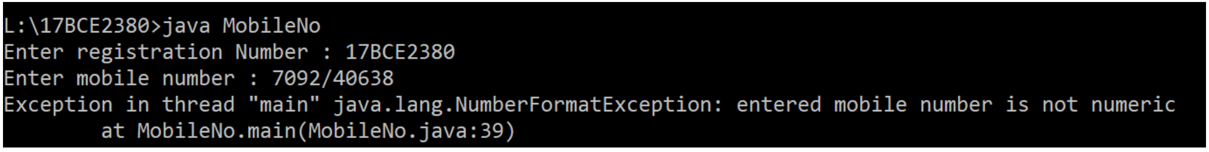
```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE2380
Enter mobile number : 709244063a
Exception in thread "main" java.lang.NumberFormatException: entered mobile number is not numeric
        at MobileNo.main(MobileNo.java:39)
```

CASE-5: mobile no. consists of alphabet.



```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE2380
Enter mobile number : 709244063a
Exception in thread "main" java.lang.NumberFormatException: entered mobile number is not numeric
        at MobileNo.main(MobileNo.java:39)
```

CASE-6: Mobile No. consist of special character.



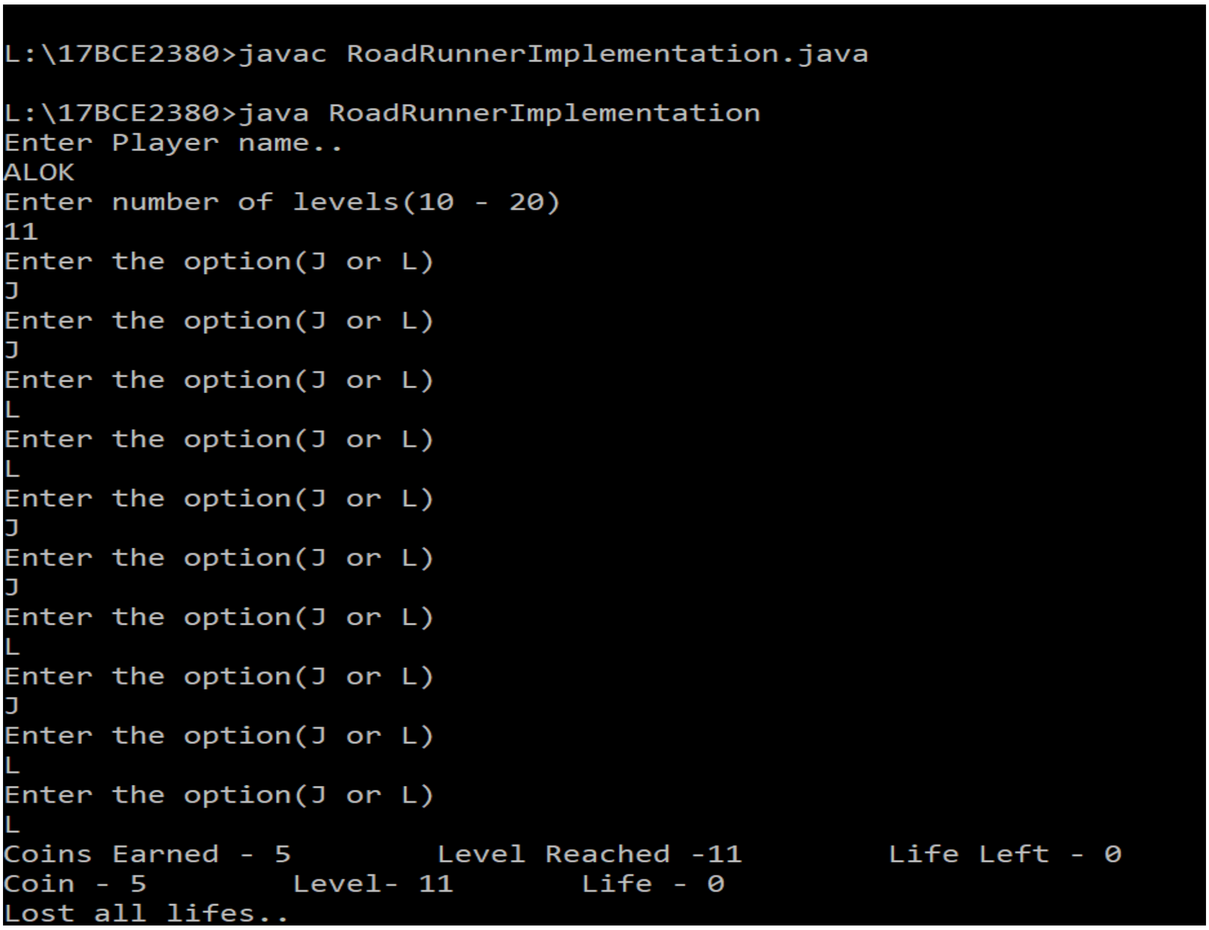
```
L:\17BCE2380>java MobileNo
Enter registration Number : 17BCE2380
Enter mobile number : 7092/40638
Exception in thread "main" java.lang.NumberFormatException: entered mobile number is not numeric
        at MobileNo.main(MobileNo.java:39)
```

5. Develop a java program to design a RoadRunner game, The player wins the game if he plays 100 levels and earns 100 coins in within 5 life. Create a class RoadRunner with datamembers - name, coins, level, life. To play the game, call play() which in turn calls run(). Inside the run() count the no of jumps; for every jump, count 1 coin. Once the count of the coin reaches 100 and life is >0&&life<5, return the status true else return false. **2M**
   a. If the status returned is true, Increase the level, else raise an exceptions
   b. If the life=0 then raise an exception.
   c. If the player completes 100 levels, Declare win else raise an exception.

CODE

```
import java.util.*;
class MyException extends Exception{
String exceptionName;
public MyException(String n) {
exceptionName = n;
}
public String toString(){
return(exceptionName);
}
}
class RoadRunner extends Thread{
String playerName;
int level, coin,life;
public RoadRunner(String name, int level){
this.playerName = name;
this.level = level;
this.coin = 0;
this.life =5;
}
public void play(){
run();
}
public void run(){
String playOption;
Scanner inp = new Scanner(System.in);
int countLevel=0;
try{
while((countLevel < level) && (life>0)){
System.out.println("Enter the option(J or L)");
playOption = inp.next();
if (playOption.equalsIgnoreCase("j"))
coin++;
else
life--;
countLevel++;}
System.out.println("Coins Earned - "+coin+"\tLevel Reached -"+ level+"\t Life Left - "+life);
System.out.println("Coin - "+coin+"\tLevel- "+level+"\tLife - "+life);
if ((coin == level) && (life >0))
System.out.println(playerName+" Won the game. Congrats");
if (life ==0)
throw new MyException("Lost all lifes..");
if (coin < level)
throw new MyException(playerName+" Lost the game with "+life+" life");
}catch(MyException e) {
System.out.println(e.toString());
}}}
public class RoadRunnerImplementation{
public static void main(String []arr) {
String name;
Scanner inp = new Scanner(System.in);
System.out.println("Enter Player name..");
name = inp.nextLine();
System.out.println("Enter number of levels(10 - 20)");
int level = inp.nextInt();
RoadRunner rr = new RoadRunner(name,level);
rr.play();
}
}
```

OUTPUT



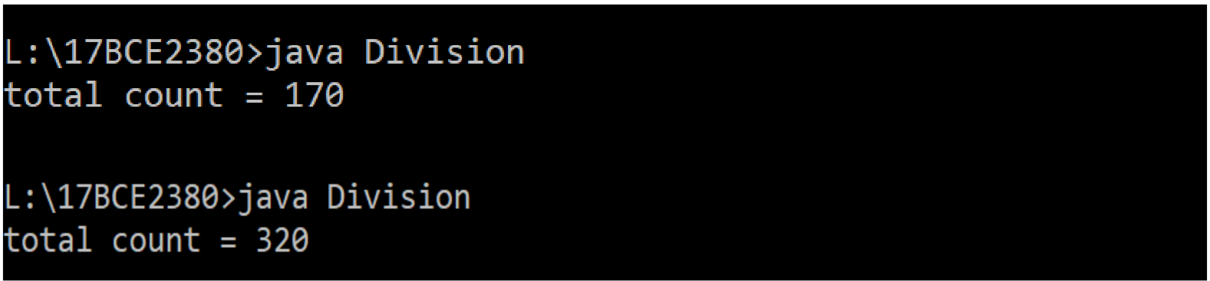
```
L:\17BCE2380>javac RoadRunnerImplementation.java

L:\17BCE2380>java RoadRunnerImplementation
Enter Player name..
ALOK
Enter number of levels(10 - 20)
11
Enter the option(J or L)
J
Enter the option(J or L)
J
Enter the option(J or L)
L
Enter the option(J or L)
L
Enter the option(J or L)
J
Enter the option(J or L)
J
Enter the option(J or L)
L
Enter the option(J or L)
J
Enter the option(J or L)
L
Enter the option(J or L)
L
Coins Earned - 5        Level Reached -11        Life Left - 0
Coin - 5        Level- 11       Life - 0
Lost all lifes..
```

6.Design a Java multi-threaded program to count numbers divisible by 3 and 4 which are ranges between 1 and 10000. Create five child thread objects to implement this process parallel.

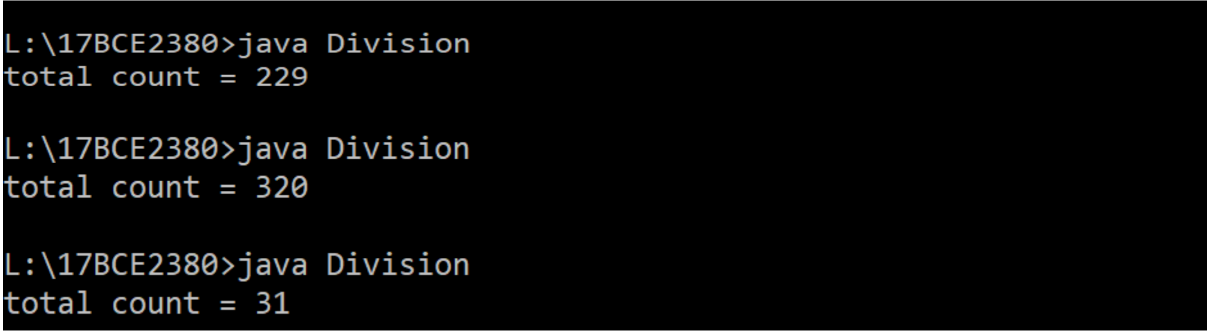
CODE:

```
public class Division implements Runnable{
int cnt=0;
public void run() {
for(int i=0;i<10000;i++) {
        if((i%3==0) && (i%4==0)) {
                cnt++;
        }
}
}
public void print() {
        System.out.println("total count = "+cnt);
}
public static void main(String args[]) {
        Division obj1=new Division();
        Thread t1=new Thread(obj1);
        Thread t2=new Thread(obj1);
        Thread t3=new Thread(obj1);
        Thread t4=new Thread(obj1);
        Thread t5=new Thread(obj1);
        t1.start();
        t2.start();
        t3.start();
        t4.start();
        t5.start();
        obj1.print();

}
}
```

OUTPUT



```
L:\17BCE2380>java Division
total count = 170


L:\17BCE2380>java Division
total count = 320
```



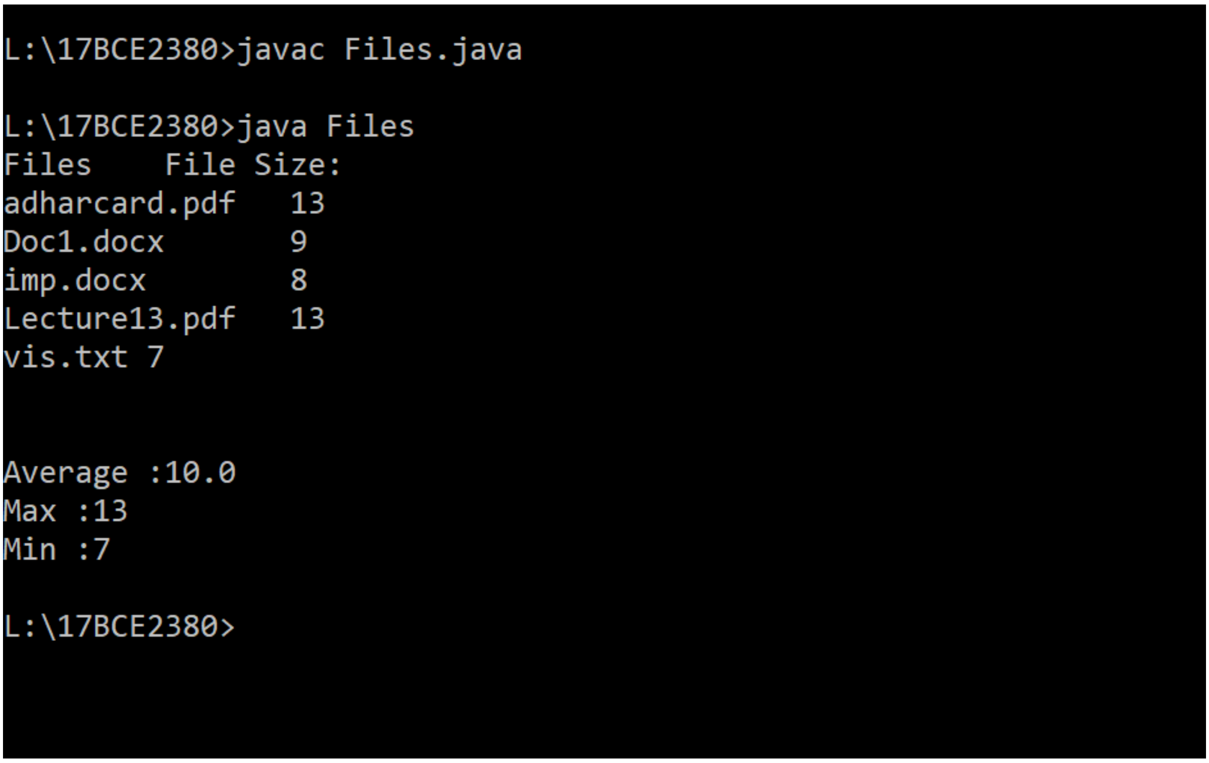
```
L:\17BCE2380>java Division
total count = 229

L:\17BCE2380>java Division
total count = 320

L:\17BCE2380>java Division
total count = 31
```

7. Design a Java program to list the files of a directory along with size. Also find the average file size, minimum file size and maximum file size of that directory. **1M**

CODE

```
import java.io.File;
public class Files {
  public static void main(String[] args) {
    File folder = new File("L:/17BCE2380/FILE");
            String[] files = folder.list();
            int i=0;
            float total=0;
            int max=0;
            System.out.println("Files \t File Size:");
            for(String file:files){
                        System.out.println(file+"\t"+file.length());
                        i++;
                        total = total+file.length();
                        if(file.length()>max){
                                    max=file.length();
                        }
            }
            int min=max;
            for(String file:files){
                        if(file.length()<min){
                                    min=file.length();
                        }
            }
            System.out.println();
            System.out.println();
            System.out.println("Average :"+total/i);
            System.out.println("Max :"+max);
            System.out.println("Min :"+min);
  }
}
```

OUTPUT



```
L:\17BCE2380>javac Files.java

L:\17BCE2380>java Files
Files    File Size:
adharcard.pdf   13
Doc1.docx       9
imp.docx        8
Lecture13.pdf   13
vis.txt 7


Average :10.0
Max :13
Min :7

L:\17BCE2380>
```

8. Write Java programs to implement the following: **2M**

a. Create class **Loan** with data members **client name, address, age, salary, loan amount, loan type (housing, vehicle, personal).** Define a method print, to display all details.

b. Take the necessary inputs and write ‘n’ objects into the file “YourREGNUM.txt”.

c. Let the bank manager will fetch loan details from the file and verify the details for approval when the salary is more than the loan amount/12.

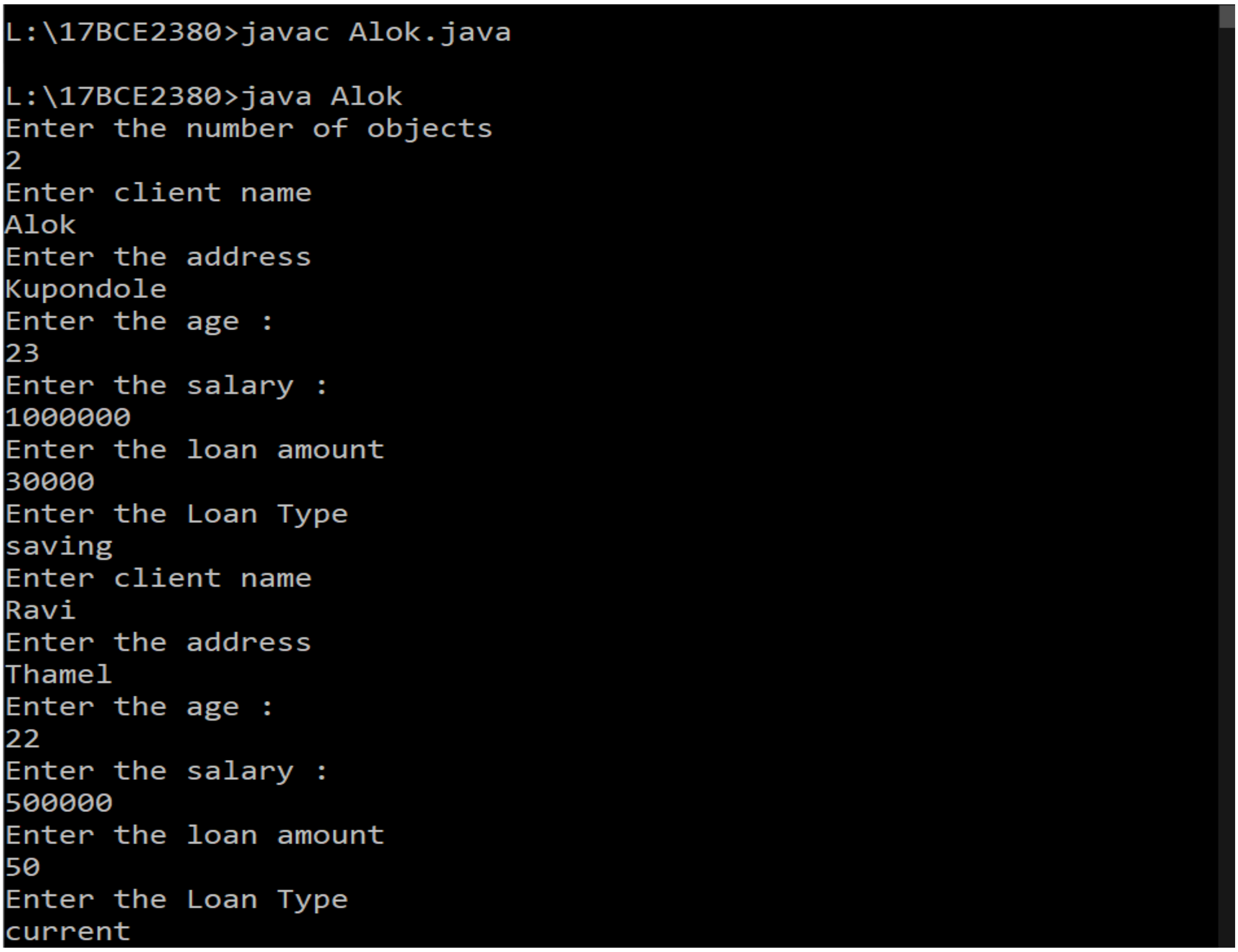
d. Write a program to display the details of approved loans, number of loans approved in each category.

CODE:

```
import java.io.*;
import java.util.*;
class Loan implements Serializable{
String clientname;
String address;
int age;
int salary;
int loan_amount;
String loan_type;
boolean loan_approved = false;
void print(){
System.out.println("Printing Details .....");
System.out.println("client name: " + clientname);
System.out.println("address :" + address);
System.out.println(" client age is : " + age);
System.out.println("client salary is : " + salary);
System.out.println("client loan amount is : " + loan_amount);
System.out.println(" client loan type is : " + loan_type);
}
void input(){
Scanner inp = new Scanner(System.in);
System.out.println("Enter client name");
this.clientname = inp.nextLine();
System.out.println("Enter the address");
this.address = inp.nextLine();
System.out.println("Enter the age : ");
this.age = inp.nextInt();
System.out.println("Enter the salary : ");
this.salary = inp.nextInt();
System.out.println("Enter the loan amount");
this.loan_amount = inp.nextInt();
System.out.println("Enter the Loan Type");
inp.nextLine();
this.loan_type = inp.nextLine();
}
}
public class Alok {
public static void main(String[] args) throws Exception{
int numobjects;
System.out.println("Enter the number of objects");
Scanner inp = new Scanner(System.in);
numobjects = inp.nextInt();
Loan arr[] = new Loan[numobjects];
File f = new File("YourREGNUM.txt");
FileOutputStream fos = new FileOutputStream(f);
ObjectOutputStream oos = new ObjectOutputStream(fos);
for(int i=0;i<numobjects;i++){
arr[i] = new Loan();
arr[i].input();
oos.writeObject(arr[i]);
}
FileInputStream fis = new FileInputStream(f);
ObjectInputStream ois = new ObjectInputStream(fis);
for(int j=0;j<numobjects;j++){
Loan obj = (Loan) ois.readObject();
// System.out.println(obj.loan_amount);
if(obj.salary > (obj.loan_amount/12)){
obj.loan_approved = true;
arr[j].loan_approved = true;
}
}
int loan_approved_saving = 0;
int loan_approved_current = 0;
int others = 0;
for(int i=0;i<numobjects;i++){
if(arr[i].loan_approved == true){
arr[i].print();
String s = "saving";
String c = "current";
if(arr[i].loan_type.equals(s)){
loan_approved_saving++;
}
else if(arr[i].loan_type.equals(c)){
loan_approved_current++;
}
else{
others++;
}
}
}
System.out.println("Total Number Of Saving Account Loan Approved : " +
loan_approved_saving);
System.out.println("Total Number of Current Account Loan Approved : " +
loan_approved_current);
System.out.println("Total Number of Others Accoutn Loan Approved " +
others);
}
}
```

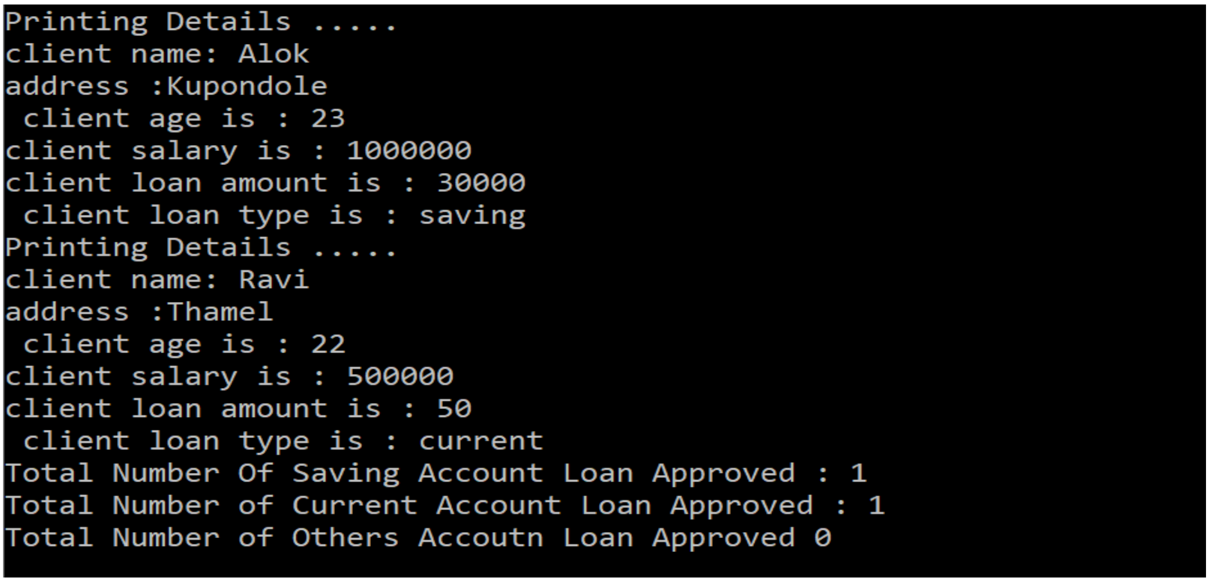
OUTPUT

FIG 1:



```
L:\17BCE2380>javac Alok.java

L:\17BCE2380>java Alok
Enter the number of objects
2
Enter client name
Alok
Enter the address
Kupondole
Enter the age :
23
Enter the salary :
1000000
Enter the loan amount
30000
Enter the Loan Type
saving
Enter client name
Ravi
Enter the address
Thamel
Enter the age :
22
Enter the salary :
500000
Enter the loan amount
50
Enter the Loan Type
current
```

FIG 2:



```
Printing Details .....
client name: Alok
address :Kupondole
 client age is : 23
client salary is : 1000000
client loan amount is : 30000
 client loan type is : saving
Printing Details .....
client name: Ravi
address :Thamel
 client age is : 22
client salary is : 500000
client loan amount is : 50
 client loan type is : current
Total Number Of Saving Account Loan Approved : 1
Total Number of Current Account Loan Approved : 1
Total Number of Others Accoutn Loan Approved 0
```