Q1. Design a class named Rectangle to represent a rectangle. The class contains: Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.

      (i)A default constructor that creates a default rectangle.

      (ii)A constructor that creates a rectangle with the specified width and height.

      (iii)A method named getArea() that returns the area of this rectangle.

      (iv)A method named getPerimeter() that returns the perimeter.

```java
import java.util.Scanner;
public class Rectangle
{
    public double width;
    public double height;
    Rectangle()
    {
        width=1;
        height=1;
    }
    Rectangle(double width, double height)
    {
        this.width=width;
        this.height=height;
    }
    public double getArea()
    {
        return (width*height);
    }
    public double getPerimeter()
    {
        return (2*(width+height));
    }
    public static void main(String[] args)
    {
        Rectangle r1= new Rectangle();
        Scanner input=new Scanner(System.in);
        System.out.println("Default Width : "+r1.width);
        System.out.println("Default Height : "+r1.height);
        System.out.println("Default Area : "+r1.getArea());
        System.out.println("Default Perimeter : "+r1.getPerimeter());
        System.out.println("Enter width: ");
        double width = input.nextDouble();
        System.out.println("Enter height: ");
        double height= input.nextDouble();
        Rectangle r2= new Rectangle(width,height);
        System.out.println("Width : "+r2.width);
        System.out.println("Height : "+r2.height);
        System.out.println("Area : "+r2.getArea());
        System.out.println("Perimeter : "+r2.getPerimeter());
        System.out.println("Enter width: ");
        width = input.nextDouble();
```

```java
        System.out.println("Enter height: ");
        height= input.nextDouble();
        Rectangle r3= new Rectangle(width,height);
        System.out.println("Width : "+r3.width);
        System.out.println("Height : "+r3.height);
        System.out.println("Area : "+r3.getArea());
        System.out.println("Perimeter : "+r3.getPerimeter());
    }
}
```


2.Write a Java program to create a class called Student having data members Regno, Name, Course being studied and current CGPA. Include constructor to initialize objects. Create array of objects with at least 10 students and find 9-pointers.

```java
import java.util.Scanner;
public class Student
{
    public String Regno;
    public String Name;
    public String Course;
    public float CGPA;
    Student()
    {
        Regno="NULL";
        Name="NULL";
        Course="NULL";
        CGPA=0;
    }
    Student(String Regno, String Name, String Course, float CGPA)
    {
```

```java
            this.Regno=Regno;
            this.Name=Name;
            this.Course=Course;
            this.CGPA=CGPA;
    }
    public void display()
    {
            System.out.println("Regno: "+Regno);
            System.out.println("Name: "+Name);
            System.out.println("Course: "+Course);
            System.out.println("CGPA: "+CGPA);
    }
    public static void main(String[] args)
    {
            Scanner input=new Scanner(System.in);
            Student s[]= new Student[10];
            Student def= new Student();
            System.out.println("Default values are: ");
            def.display();
            for(int i=0;i<10;i++)
            {
                    System.out.println("Regno: ");
                    String Regno=input.nextLine();
                    System.out.println("Name: ");
                    String Name=input.nextLine();
                    System.out.println("Course: ");
                    String Course=input.nextLine();
                    System.out.println("CGPA: ");
                    float CGPA=input.nextFloat();
                    input.nextLine();
                    s[i]=new Student(Regno,Name,Course,CGPA);
            }
            System.out.println("The 9 pointers are: ");
            for(int i=0;i<10;i++)
            {
                    if(s[i].CGPA>=9)
                    {
                            s[i].display();
                    }
            }
    }
}
```

Screenshot 1 — Notepad++ (Student.java) with command prompt

```java
22      }
23      public void display()
24      {
25          System.out.println("Regno: "+Regno);
26          System.out.println("Name: "+Name);
27          System.out.println("Course: "+Course);
28          System.out.println("CGPA: "+CGPA);
29      }
30      public static void main(String[] args)
31      {
32          Scanner input=new Scanner(System.in);
33          Student s[]= new Student[10];
34          Student def= new Student();
35          System.out.println("Default values a
36          def.display();
37          for(int i=0;i<10;i++)
38          {
39              System.out.println("Regno: ");
40              String Regno=input.nextLine();
41              System.out.println("Name: ");
42              String Name=input.nextLine();
43              System.out.println("Course: ");
44              String Course=input.nextLine();
45              System.out.println("CGPA: ");
46              float CGPA=input.nextFloat();
47              input.nextLine();
48              s[i]=new Student(Regno,Name,Course,CGPA);
49          }
50          System.out.println("The 9 pointers are: ");
51          for(int i=0;i<10;i++)
52          {
53              if(s[i].CGPA>=9)
```

Command prompt output:
```
C:\Windows\System32\cmd.exe
D:\VTOP\java-19bce0470>javac Student.java
D:\VTOP\java-19bce0470>java Student
Default values are:
Regno: NULL
Name: NULL
Course: NULL
CGPA: 0.0
Regno:
19BCE0470
Name:
Ripunjay
Course:
btech
CGPA:
8.4
Regno:
19bce001
Name:
abc
Course:
btech
CGPA:
8.2
Regno:
19bce002
Name:
bcd
Course:
btech
```

length : 1,307   lines : 59   Ln : 59   Col : 2   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

Screenshot 2 — Notepad++ (Student.java) with command prompt

```java
22      }
23      public void display()
24      {
25          System.out.println("Regno: "+Regno);
26          System.out.println("Name: "+Name);
27          System.out.println("Course: "+Course
28          System.out.println("CGPA: "+CGPA);
29      }
30      public static void main(String[] args)
31      {
32          Scanner input=new Scanner(System.in);
33          Student s[]= new Student[10];
34          Student def= new Student();
35          System.out.println("Default values a
36          def.display();
37          for(int i=0;i<10;i++)
38          {
39              System.out.println("Regno: ");
40              String Regno=input.nextLine();
41              System.out.println("Name: ");
42              String Name=input.nextLine();
43              System.out.println("Course: ");
44              String Course=input.nextLine();
45              System.out.println("CGPA: ");
46              float CGPA=input.nextFloat();
47              input.nextLine();
48              s[i]=new Student(Regno,Name,Course,CGPA);
49          }
50          System.out.println("The 9 pointers are: ");
51          for(int i=0;i<10;i++)
52          {
53              if(s[i].CGPA>=9)
```

Command prompt output:
```
C:\Windows\System32\cmd.exe
bcd
Course:
btech
CGPA:
10
Regno:
19BCE0003
Name:
cde
Course:
mtech
CGPA:
8.5
Regno:
18BIT001
Name:
efg
Course:
mba
CGPA:
9.3
Regno:
19BIT0005
Name:
def
Course:
btech
CGPA:
5.4
Regno:
```

length : 1,307   lines : 59   Ln : 59   Col : 2   Sel : 0 | 0   Windows (CR LF)   UTF-8   INS

3.Write a Java program that displays the time in different formats in the form of HH,MM,SS using constructor Overloading

```java
import java.util.Scanner;
public class Time
{
    Time(int H,int M,int S)
    {
        System.out.println("The time in HH:MM:SS format is: "+H+":"+M+":"+S);
    }
    Time(int S)
    {
        System.out.println(S);
    }
}
```

```java
Time(int M,int H)
{
    System.out.print("The time in MM:HH:SS format is: "+M+":"+H+":");
}

public static void main(String[] args)
{
    int s,m,h;
    Scanner x = new Scanner(System.in);
    System.out.println("Enter the Hours");
    h=x.nextInt();
    System.out.println("Enter the Minutes");
    m=x.nextInt();
    System.out.println("Enter the Seconds");
    s=x.nextInt();
    Time j= new Time(h,m,s);
    Time y= new Time(m,h);
    Time z= new Time(s);
}
}
```


4.Write a Java program that displays area of different Figures (Rectangle, Square, Triangle) using the method overloading.

```java
public class MO1{

    float area(float a, float b) // area of rectangle
    {
            System.out.println("The area of the rectangle: ");
        return(a*b);
    }
    float area(float s)   // area of square
    {
```

```java
        System.out.println("The area of the square: ");
            return (s*s);
    }
    double area (float a, float b, float c)
    // area of triangle
    {
        float s= (a+b+c)/2;
        double ar=0;
        ar=Math.sqrt(s*(s-a)*(s-b)*(s-c));
            System.out.println("The area of the triangle: ");
        return ar;
    }
    public static void main(String []args){
        MO1 q= new MO1() ;
        //rectangle
        System.out.println(q.area(10, 20));

        // square
        System.out.println(q.area(10));

        //triangle

        System.out.println(q.area(1, 1,1));
    }
}
```


5.In a school, students of all classes from std I to X appear for the MathPremierLeague examination. Define a class MPL which stores the details of the marks scored by each class. It should contain the following 4 data members: Standard, number of students, marks[] array to store the scores of all the students of the class in MPL exam. Define a parameterized constructor which receives the values

for the first two data members from the main() method. Create a Form within the constructor, read the marks of all students and hence find the first mark. Define a method findBestClass() to display the standard which has secured the highest mark. Overload this method to display the standard with the highest class average. The marks array should be declared dynamically based on the strength of the class.

```java
import java.util.Scanner;
public class MPL{
    int Standard;
    int No_of_Students;
    int marks[];
    int bestmark;
    int classavg;
    MPL(int Standard, int No_of_Students)
    {
        this.Standard=Standard;
        this.No_of_Students=No_of_Students;
        Scanner input= new Scanner(System.in);
        System.out.println("Form to fill up all the students marks: ");
        this.marks=new int[No_of_Students];
        int a=-1,sum=0;
        for(int i=0;i<No_of_Students;i++)
        {
            marks[i]=input.nextInt();
            sum=sum+marks[i];
            if(marks[i]>a)
            {
                a=marks[i];
            }
        }
        this.classavg=sum/No_of_Students;
        this.bestmark=a;
        System.out.println("First mark is: "+bestmark);
    }
    void findBestClass(MPL m[],int n)
    {
        int a=-1;
        for(int i=0;i<n;i++)
        {
            if(m[i].bestmark>a)
            {
                a=m[i].bestmark;
            }
        }
        for(int i=0;i<No_of_Students;i++)
        {
            if(m[i].bestmark==a)
            {
                System.out.println("Class with highest mark is: "+m[i].Standard);
                break;
            }
        }
    }
```

```java
void findBestClass(MPL m[])
{
    int a=-1;
    for(int i=0;i<No_of_Students;i++)
    {
        if(m[i].classavg>a)
        {
            a=m[i].classavg;
        }
    }
    for(int i=0;i<No_of_Students;i++)
    {
        if(m[i].classavg==a)
        {
            System.out.println("Class with highest average is: "+m[i].Standard);
            break;
        }
    }
}

public static void main(String args[])
{
    MPL m= new MPL('A',5);
    Scanner input = new Scanner(System.in);
    System.out.println("Enter no.of classes");
    int n= input.nextInt();
    MPL arr[]=new MPL[n];
    for(int i=0;i<n;i++)
    {
        System.out.println("Enter standard");
        int std= input.nextInt();
        System.out.println("Enter No.of Students");
        int num= input.nextInt();
        arr[i]=new MPL(std,num);
    }
    arr[0].findBestClass(arr,n);
    arr[0].findBestClass(arr);
}
}
```

6.Read the following details of 'n' students using Scanner class methods and display the same. –
Registration number (String) – Name (String that may contain first name, middle name and last
name) – CGPA (Floating point number) – Programme Name(String) – School Name (String with
multiple words) – Proctor Name (String that may contain first, middle and last names)

```java
import java.util.Scanner;
public class StudentDetails
{
        String RegNo;
        String Name;
        float CGPA;
        String Program;
        String School;
        String Proctor;

        void display()
        {
                System.out.println("RegNo: "+this.RegNo);
                System.out.println("Name: "+this.Name);
                System.out.println("CGPA: "+this.CGPA);
                System.out.println("Program: "+this.Program);
                System.out.println("School: "+this.School);
                System.out.println("Proctor: "+this.Proctor);
        }
        StudentDetails()
        {
                Scanner input= new Scanner(System.in);
                System.out.println("Enter Regno");
                this.RegNo=input.nextLine();
                System.out.println("Enter Name");
                this.Name=input.nextLine();
                System.out.println("Enter CGPA");
                this.CGPA=input.nextFloat();
                input.nextLine();
```

```java
            System.out.println("Enter Program");
            this.Program=input.nextLine();
            System.out.println("Enter School");
            this.School=input.nextLine();
            System.out.println("Enter Proctor");
            this.Proctor=input.nextLine();
    }

    public static void main(String args[])
    {
            Scanner input= new Scanner(System.in);
            System.out.println("Enter no.of students");
            int n= input.nextInt();
            StudentDetails s[]=new StudentDetails[n];
            for(int i=0;i<n;i++)
            {
                    s[i]=new StudentDetails();
            }
            for(int i=0;i<n;i++)
            {
                    s[i].display();
            }
    }
}
```


7.A training centre conducts a total of 7 tests for its students. Students are allowed to skip few tests. Let there be 25 students in the batch. So in the main class for every student, read the number of tests taken and the marks scored in each test. A class 'TestDetails' should be defined with a 2D array of float type to store the marks of all the students. Define a method 'storeMarks()' that will receive the following details for every student from the main class and create in the 2D array, those many columns equal to the number of tests, so as to store the marks. There is no need to store the number of tests. Define another method 'displayMarks()' to print the details. Also the training centre wishes to keep those students in notice period who have taken < 3 tests and those who have

not scored ≥ 50 in at least 3 tests. Derive another class 'NoticePeriod' from 'TestDetails' that includes a method to count and print the number of students in bench. Also it should print the ID of those students assuming the row index of the array to be their ID. While checking do not proceed to check the marks in all tests, if the student has already scored more than 50 in 3 tests. Instantiate this class from the main class and do the required processing.

```java
import java.util.Scanner;
class TestDetails
{
    static float storearr[][];
    static void storeMarks(float arr[][])
    {
        storearr=new float[25][];
        for(int i=0;i<25;i++)
        {
            storearr[i]=new float[arr[i].length];
            storearr[i]=arr[i];
        }
    }
    static void displayMarks()
    {
        for(int i=0;i<25;i++)
        {
            for(int j=0;j<storearr[i].length;j++)
            {
                System.out.print(storearr[i][j]+" ");
            }
            System.out.println();
        }
    }
}
class NoticePeriod extends TestDetails
{
    int index[]=new int[25];
    int k=0;
    void benchedStudents(float storearr[][])
    {
        for(int i=0;i<storearr.length;i++)
        {
            if(storearr[i].length<3)
            {
                index[k]=i;
                k++;
            }
            else
            {
                int count=0;
                for(int j=0;j<storearr[i].length;j++)
                {
                    if(storearr[i][j]>50f)
                    {
                        count++;
                        if(count>=3)
```

```java
                    {
                        break;
                    }
                }
            }
            if(count<3)
            {
                index[k]=i;
                k++;
            }
        }
    }
    System.out.println("The indices of the benched students: ");
    for(int i=0;i<k;i++)
    {
        System.out.print(index[i]+" ");
    }
    System.out.println();
    }
}
public class MainStudentTest extends TestDetails
{
    public static void main(String args[])
    {
        Scanner input= new Scanner(System.in);
        float arr[][]=new float[25][];
        for(int i=0;i<25;i++)
        {
            System.out.println("No.of tests");
            int n=input.nextInt();
            System.out.println("Enter the marks of student for attempted tests: ");
            arr[i]=new float[n];
            for(int j=0;j<n;j++)
            {
                arr[i][j]=input.nextFloat();
            }
        }
        storeMarks(arr);
        displayMarks();
        NoticePeriod np=new NoticePeriod();
        np.benchedStudents(arr);

    }
}
```

```java
53              if(count<3)
54              {
55                  index[k]=i;
56                  k++;
57              }
58          }
59      }
60      System.out.println("The indices of the ben
61      for(int i=0;i<k;i++)
62      {
63          System.out.print(index[i]+" ");
64      }
65      System.out.println();
66  }
67 }
68 public class MainTest extends TestDetails
69 {
70     public static void main(String args[])
71     {
72         Scanner input= new Scanner(System.in);
73         float arr[][]=new float[25][];
74         for(int i=0;i<25;i++)
75         {
76             System.out.println("No.of tests");
77             int n=input.nextInt();
78             System.out.println("Enter the marks of
79             arr[i]=new float[n];
80             for(int j=0;j<n;j++)
81             {
82                 arr[i][j]=input.nextFloat();
83             }
84         }
85         storeMarks(arr);
```

Java source file

C:\Windows\System32\cmd.exe

```
No.of tests
5
Enter the marks of student for attempted tests:
5
Enter the marks of student for attempted tests:
1
66
No.of tests
1
55
No.of tests
1
25
No.of tests
1
66
No.of tests
1
88
No.of tests
1
44
No.of tests
2
55
66
No.of tests
3
44
55
66
No.of tests
1
77
No.of tests
1
33
No.of tests
1
```

```java
53              if(count<3)
54              {
55                  index[k]=i;
56                  k++;
57              }
58          }
59      }
60      System.out.println("The indices of the ben
61      for(int i=0;i<k;i++)
62      {
63          System.out.print(index[i]+" ");
64      }
65      System.out.println();
66  }
67 }
68 public class MainTest extends TestDetails
69 {
70     public static void main(String args[])
71     {
72         Scanner input= new Scanner(System.in);
73         float arr[][]=new float[25][];
74         for(int i=0;i<25;i++)
75         {
76             System.out.println("No.of tests");
77             int n=input.nextInt();
78             System.out.println("Enter the marks of
79             arr[i]=new float[n];
80             for(int j=0;j<n;j++)
81             {
82                 arr[i][j]=input.nextFloat();
83             }
84         }
85         storeMarks(arr);
```

Java source file

C:\Windows\System32\cmd.exe

```
No.of tests
Enter the marks of student for attempted tests:
77
No.of tests
1
77
No.of tests
1
33
No.of tests
1
22
No.of tests
1
44
No.of tests
1
77
No.of tests
1
88
No.of tests
1
99
No.of tests
1
100
No.of tests
1
65
No.of tests
1
45
No.of tests
1
35
No.of tests
1
18
No.of tests
```

8.Create an inheritance hierarchy in java using following information given below that a bank might use to represent customers' bank accounts. Base class Account should include one data member of type double to represent account balance. The class should provide constructor that receives an initial balance and uses it to initialize the data member.The constructor should validate the initial balance to ensure that it is greater than or equal to 0.If not the balance is set to 0.0 and the constructor should display an error message,indicating that the initial balance was invalid. The class also provides three member functions credit, debit(debit amount should not exceed the account balance) and enquiry.Derived class SavingsAccount should inherit the functionality of an Account, but also include data member of type double indicating the interest rate assigned to the Account. SavingsAccount constructor should receive the initial balance, as well as an initial value for SavingsAccount's interest rate. SavingsAccount should provide public member function calculateInterest that returns double indicating the amount of interest earned by an account.The method calculateInterest should determine this amount by multiplying the interest rate by the account balance. SavingsAccount function should inherit member functions credit,debit and enquiry without redefining them. Derived class CheckingAccount should inherit the functionality of an Account, but also include data member of type double that represents the fee charged per transaction. CheckingAccount constructor should receive the initial balance, as well as parameter indicating fee amount. class CheckingAccount should redefine credit and debit function so that they subtract the fee from account balance whenever either transaction is performed. CheckingAccount's debit function should charge a fee only if the money is actually withdrawn (debit amount should not exceed the account balance).After defining the class hierarchy, write program that creates object of each class and tests their member functions. Add interest to SavingAccount object by first invoking its calculateInterest function, then passing the returned interest amount to object's credit function.

```java
import java.util.Scanner;
class Account
{
    double accbalance;
    Account(double accbalance)
    {
        this.accbalance=accbalance;
        if(this.accbalance<0)
```

```java
                {
                        this.accbalance=0.0;
                        System.out.println("The Initial Balance is Invalid");
                }
        }
        void credit(double amount)
        {
                this.accbalance+=amount;
                System.out.println("Amount credited: "+amount);
                System.out.println("New Account Balance: "+this.accbalance);
        }
        void debit(double amount)
        {
                if(this.accbalance<amount)
                {
                        System.out.println("Insufficient Funds");
                }
                else
                {
                        this.accbalance-=amount;
                        System.out.println("Amount debited: "+amount);
                        System.out.println("New Account Balance: "+this.accbalance);
                }
        }
        void enquiry()
        {
                System.out.println("Current Account Balance: "+this.accbalance);
        }
}
class SavingsAccount extends Account
{
        double interest;
        SavingsAccount(double amount, double interest)
        {
                super(amount);
                this.interest=interest;
        }
        double calculateInterest()
        {
                return interest*accbalance;
        }

}

public class CheckingAccount extends Account
{
        double fee;
        CheckingAccount(double amount,double fee)
        {
                super(amount);
                this.fee=fee;
        }
        void credit(double amount)
```

```
{
    super.accbalance=super.accbalance+amount-this.fee;
    System.out.println("Amount credited: "+amount);
    System.out.println("New Account Balance: "+super.accbalance);
}
void debit(double amount)
{
    if(super.accbalance<amount)
    {
        System.out.println("Insufficient Funds");
    }
    else
    {
        super.accbalance=super.accbalance+amount-this.fee;
        System.out.println("Amount debited: "+amount);
        System.out.println("New Account Balance: "+super.accbalance);
    }
}
public static void main(String args[])
{
    Account a= new Account(12345.56);
    a.credit(33.51);
    a.debit(38);
    a.enquiry();
    SavingsAccount sa= new SavingsAccount(12345.56,57);
    double interest= sa.calculateInterest();
    sa.credit(interest);
    CheckingAccount ca=new CheckingAccount(43251.22,66);
    ca.credit(72);
    ca.debit(2589.54);
}
}
```

9. Write an interface called Exam with a method Pass ( ) that returns the total marks. Write another interface called Classify with a method Average (int total) which returns a string. Write a Class called Result which implements both Exam and Classify. The Pass method should get the marks from the user and finds the total marks and return. The Division method calculate the average marks and return "First" if the average is 60 or more, "SECOND" when average is 50 or more but below 60, "NO DIVISION" when average is less than 50

```java
import java.util.Scanner;
interface Exam
{
     int Pass();
}

interface Classify
{
     String Average(int total);
}

public class Result implements Exam,Classify
{
     int n;
     public int Pass()
     {
          Scanner input= new Scanner(System.in);
          System.out.println("Enter the no.of students");
          this.n= input.nextInt();
          int s[]=new int[n];
          int total=0;
          System.out.println("Enter the marks of the students");
          for(int i=0;i<n;i++)
          {
               s[i]=input.nextInt();
               total+=s[i];
          }
          return total;
     }
     public String Average(int total)
     {
          return String.valueOf(total/n);
     }
     String Division(String average)
     {
          int avg= Integer.parseInt(average);
          if(avg>=60)
          {
               return "FIRST";
          }
          else if(avg>=50)
          {
               return "SECOND";
          }
          else
          {
```
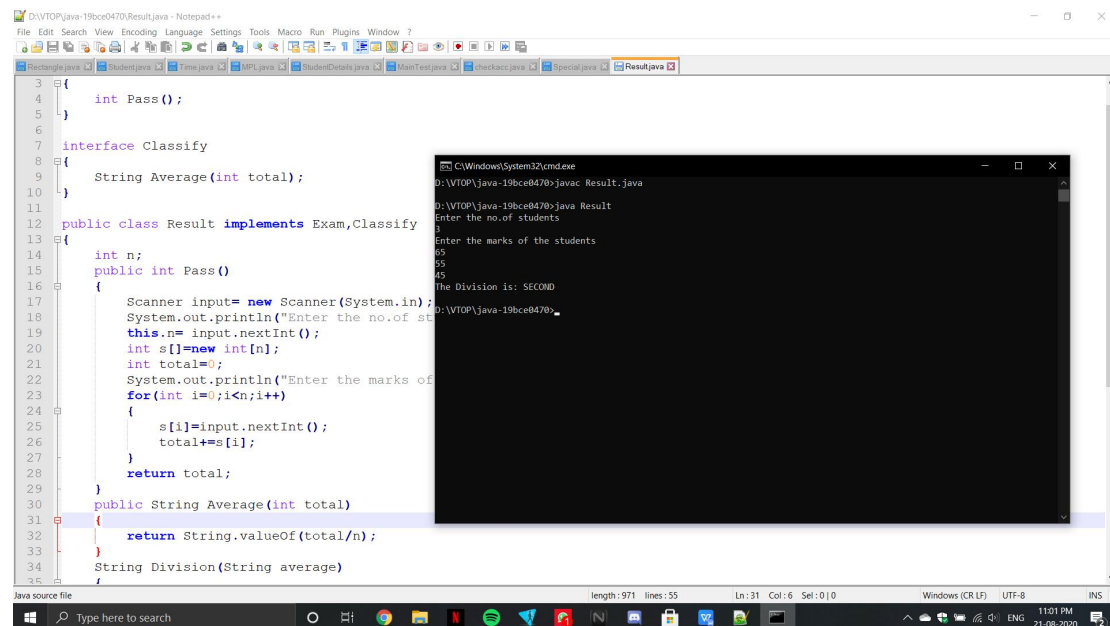
```
            return "NO DIVISION";
        }
    }
    public static void main(String args[])
    {
        Result r=new Result();
        System.out.println("The Division is: "+r.Division(r.Average(r.Pass())));
    }
}
```


10. Write an abstract class special with an abstract method double Process (double P,double R). Create a subclass Discount and implement the Process() method with the following formula: net=P−P∗R/100. Return the Process() method with the following formula: total=P+P∗R/100. Return the total.

```
import java.util.Scanner;
abstract class Special
{
    abstract double Process(double P,double R);
}
class Discount extends Special
    {
        double Process(double P,double R)
        {
            double net=P−P∗(R/100);
            return net;
        }
    }
public class SpecialMain
{
    public static void main(String args[])
    {
        Discount d=new Discount();
        System.out.printf("Total %f ",d.Process(600.00,10.00));
```

```
        }
}
```



```java
import java.util.Scanner;
abstract class Special
{
    abstract double Process(double P,double R);
}
class Discount extends Special
    {
        double Process(double P,double R)
        {
            double net=P-P*(R/100);
            return net;
        }
    }
public class SpecialMain
{
    public static void main(String args[])
    {
        Discount d=new Discount();
        System.out.printf("Total %f ",d.Process(790.00,55.00));
    }
}
```

```
D:\VTOP\java-19bce0470>javac SpecialMain.java

D:\VTOP\java-19bce0470>java SpecialMain
Total 355.500000
D:\VTOP\java-19bce0470>
```