VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SHASHANK RAVINDRA KARANAM (1BM23CS312)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019 Sep

2024-Jan 2025

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **SHASHANK RAVINDRA KARANAM(1BM23CS312),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

Sl. No.	Date	Experiment Title	Page No.
1	24/10/2024	Quadratic Equation Execution	
2	24/10/2024	SGPA Calculation of Student	
3	24/10/2024	tostring() method	
4	07/11/2024	Abstract classes	
5	14/11/2024	Inheritance	
6	21/11/2024	Packages	
7	21/11/2024	Exception Handling	
8	28/11/2024	Threads	
9	18/12/2024		
10	18/12/2024		

Include all the 8 programs as instructed in the classroom.

The order to be maintain for every program is

Question

Observation writeup images (complete)

Soft copy of the program

Screenshot of the output

Editable copies of 9^{th} and 10^{th} program s are attached here. Analyze and execute and include both in the lab record pdf giving the same order as above. Explanation for both topics are included in textbook.

Include page numbers from this page onwards

LABORATORY PROGRAM – 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Lab Program 1:

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a,b,c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class quadratic{
    float d;
    Scanner sc = new Scanner(System.in);

    void check()
    {   System.out.println("Enter values of a,b&c:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

      if (a==0){
        System.out.println("Invalid equation");
        }
      else{
            d = b*b - 4*a*c;
            System.out.println(d);
            System.out.println("The solutions are:");
            if(d>0){
              system.out.print("Roots unique");
              double r1 = (-b+Math.sqrt(d))/(2*a);
              double r2 = (-b-Math.sqrt(d))/(2*a);
```

```
            System.out.println(r1+" "+r2);
        }
        if(d==0){
            System.out.println("Roots are equal");
            double r = -b/(2*a);
            System.out.println(r);
        }
        if(d<0){
            System.out.println("No real solutions");
        }
    }
}
}

public class Main{
    public static void main(String[] args){
        quadratic q1 = new quadratic();
        q1.check();
    }
}
```

Output: Enter the values of a, b, and c:
4
4
1
Discriminant: 0.0
The solutions are:
Roots are equal

24/10/24

Soft copy of the program

```
import java.util.Scanner;

class Quad_Eq_cal{
   public static void main(String [] args){
      int y=0;
      Scanner sc=new Scanner(System.in);
      System.out.println("General form  of a quadratic equation is ax^2+bx+c=0");
      do{
         System.out.print("\nEnter value of a=");
         int a=sc.nextInt();
         System.out.print("Enter value of b=");
         int b=sc.nextInt();
         System.out.print("Enter value of c=");
         int c=sc.nextInt();
         float d=(float)(Math.pow(b,2)-4*a*c);
         if(d<0){
            System.out.println("There are no real solutions");
         }
         else if(d==0){
            System.out.println("It has one repeated root(2 equal roots):");
            float r=-b/(2.0f*a);
            System.out.println("x="+r);
         }
         else{
            System.out.println("It has two distinct roots:");
            double r1=((-b+Math.sqrt(d))/(2*a));
            System.out.println("x1="+r1);
            double r2=((-b-Math.sqrt(d))/(2*a));
            System.out.println("x2="+r2);
         }
         System.out.println("\nDo you want to calculate again?(yes=0 and no=1): ");
         y=sc.nextInt();
      }while(y==0);
   }
}
```

Screenshot of the output

```
General form  of a quadratic equation is ax^2+bx+c=0

Enter value of a=2
Enter value of b=4
Enter value of c=7
There are no real solutions
```

LABORATORY PROGRAM – 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

PAGE NO :
DATE :

Lab Program 2 :

Develop a lab program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.scanner;
public class cgpa{
  public static void main(String[] args){
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of
    subjects:");
    int numSubjects = scanner.nextInt();

    double[] gradePoints = new double[numSubjects];
    int[] credits = new int[numSubjects];
    int total_credits = 0;
    double total = 0;

    for(int i = 0; i<numSubjects; i++){
        System.out.print("Enter grade points"+(i+1)+":");
        gradePoints[i] = scanner.nextDouble();

        System.out.println("Credits"+(i+1)+":");
        credits[i] = Scanner.nextInt();

        total += gradePoints[i]*credits[i];
        total_credits += credits[i];
}
```

```
        double sgpa = total/total_credits;
        System.out.println("Your SGPA is: "+sgpa
    }
}
```

Output:

Enter the number of subjects : 8
Enter grade points for subject 1 : 9
" credits " " " " : 4
" grade " " " 2 : 10
" credits " " " " : 4
" grade " " " 3 : 9
" credits " " " " : 4
" grade " " " 4 : 8
" credits " " " " : 3
" grade " " " 5 : 9
" credits " " " " : 3
" grade " " " 6 : 10
" credits " " " " : 2
" grade " " " 7 : 10
" credits " " " " : 1
" grade " " " 8 : 10
" credits " " " " : 1

Your SGPA is : 9.227272727272

24/10/24

```
import java.util.Scanner;

class Subject {
    int subM;
    int cred;
    int grade;

    void setSubDet(int marks, int cred) {
        this.subM = marks;
        this.cred = cred;

        if (subM >= 90) {
            grade = 10;
        } else if (subM >= 80) {
            grade = 9;
        } else if (subM >= 70) {
            grade = 8;
        } else if (subM >= 60) {
            grade = 7;
        } else if (subM >= 50) {
            grade = 6;
        } else if (subM >= 40) {
            grade = 5;
        } else {
            grade = 0;
        }
    }
}

class Student {

    Scanner s = new Scanner(System.in);
    Subject[] subjects = new Subject[8];

    Student() {
        for (int i = 0; i < subjects.length; i++) {
            subjects[i] = new Subject();
        }
    }

    void getMarks() {
        for (int i = 0; i < subjects.length; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            int marks = s.nextInt();
```

```
            System.out.print("Enter credit for subject " + (i + 1) + ": ");
            int cred = s.nextInt();
            subjects[i].setSubDet(marks, cred);
        }
    }

    double calSGPA() {
        double Score = 0;
        int totalCred = 0;
        double SGPA = 0.0;

        for (Subject subject : subjects) {
            Score += (subject.grade * subject.cred);
            totalCred += subject.cred;
        }

        if (totalCred > 0) {
            SGPA = Score / totalCred;
        } else {
            SGPA = 0;
        }
        return SGPA;
    }
}

public class StudentDetails {

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


        System.out.print("Enter number of semesters: ");
        int numSems = sc.nextInt();

        Student[] students = new Student[numSems];
        double cumulativeSGPA = 0.0;


        System.out.print("Enter USN: ");
        String usn = sc.next();

        System.out.print("Enter Name: ");
        String name = sc.next();
```

```
        for (int i = 0; i < numSems; i++) {
            System.out.println("Enter details for semester " + (i + 1));
            students[i] = new Student();
            students[i].getMarks();
            double semSGPA = students[i].calSGPA();
            cumulativeSGPA += semSGPA;
        }


        for (int i = 0; i < numSems; i++) {
            System.out.println("USN: " + usn);
            System.out.println("Name: " + name);
            System.out.println("SGPA for sem " + (i + 1) + ": " + students[i].calSGPA());
        }


        double CGPA = cumulativeSGPA / numSems;
        System.out.println("CGPA: " + CGPA);
    }
}
```

```
(c) Microsoft Corporation. All rights reserved.

C:\3rd_sem\JAVA\Programs\lab>javac StudentDetails.java

C:\3rd_sem\JAVA\Programs\lab>java StudentDetails
Enter number of semesters: 2
Enter USN: 1bm23cs312
Enter Name: shashank
Enter details for semester 1
Enter marks for subject 1: 9
Enter credit for subject 1: 4
Enter marks for subject 2: 9
Enter credit for subject 2: 4
Enter marks for subject 3: 9
Enter credit for subject 3: 3
Enter marks for subject 4: 8
Enter credit for subject 4: 4
Enter marks for subject 5: 8
Enter credit for subject 5: 4
Enter marks for subject 6: 7
Enter credit for subject 6: 2
Enter marks for subject 7: 7
Enter credit for subject 7: 1
Enter marks for subject 8: 5
Enter credit for subject 8: 1
Enter details for semester 2
Enter marks for subject 1: 7
Enter credit for subject 1: 1
Enter marks for subject 2: 8
Enter credit for subject 2: 4
Enter marks for subject 3: 9
Enter credit for subject 3: 34
Enter marks for subject 4: 8
Enter credit for subject 4: 4
Enter marks for subject 5: 8
Enter credit for subject 5: 4
Enter marks for subject 6: 6
Enter credit for subject 6: 4
Enter marks for subject 7: 9
Enter credit for subject 7: 2
Enter marks for subject 8: 9
Enter credit for subject 8: 2
USN: 1bm23cs312
Name: shashank
SGPA for sem 1: 0.0
USN: 1bm23cs312
Name: shashank
SGPA for sem 2: 0.0
CGPA: 0.0

C:\3rd_sem\JAVA\Programs\lab>
```

LABORATORY PROGRAM – 3

Create a class Book which contains four members: name,author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

LABORATORY PROGRAM - 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
import
java.awt.*;
import
java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
        TextField
        num1,num2;
        Button dResult;
        Label
        outResult;
        String
        out="";
        double
        resultNum;
        int flag=0;

        public DivisionMain1()
        {
                setLayout(new FlowLayout());

                dResult = new Button("RESULT");
                Label number1 = new Label("Number
                1:",Label.RIGHT); Label number2 = new
                Label("Number          2:",Label.RIGHT);
                num1=new TextField(5);
                num2=new TextField(5);
                outResult = new Label("Result:",Label.RIGHT);

                add(number1
                );
                add(num1);
```

```
        add(number2
        );
        add(num2);
        add(dResult);
        add(outResul
        t);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new
        WindowAdapter()
        {
                public void windowClosing(WindowEvent we)
                {
                        System.exit(0);
                }
        });
}
public void actionPerformed(ActionEvent ae)
{
        int
        n1,n2;
        try
        {
                if (ae.getSource() == dResult)
                {
                        n1=Integer.parseInt(num1.getText());
                        n2=Integer.parseInt(num2.getText());

                        /*if(n2==0)
                                throw new
                        ArithmeticException();*/ out=n1+"
                        "+n2+" ";
                        resultNum=n1/n2;
                        out+=String.valueOf(result
                        Num); repaint();

                }
        }
```

```
        catch(NumberFormatException e1)
        {
                flag=1;
                out="Number Format Exception!
                "+e1; repaint();
        }
        catch(ArithmeticException e2)
        {
                flag=1;
                out="Divide by 0 Exception!
                "+e2; repaint();
        }

}
public void paint(Graphics g)
{
        if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outRes
        ult.getY()+outResult. getHeight()-8);
        else
        g.drawString(out,10
        0,200); flag=0;
}
```

Demonstrate Interprocess communication and deadlock

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
```

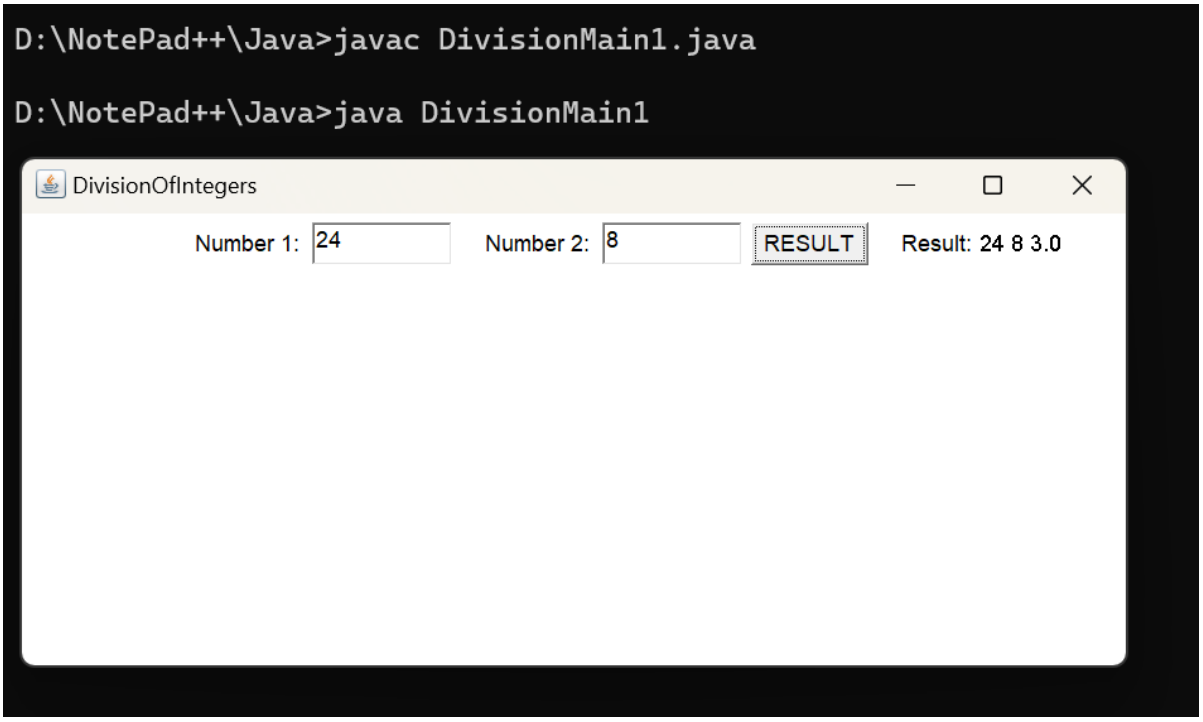
```
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
        int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

OUTPUT



ii. Demonstration of deadlock

```
class A
{
  synchronized void foo(B b)
   { String name = Thread.currentThread().getName();
     System.out.println(name + " entered A.foo");
     try { Thread.sleep(1000); }
     catch(Exception e) { System.out.println("A Interrupted"); }
     System.out.println(name + " trying to call B.last()"); b.last(); }
     synchronized void last() { System.out.println("Inside A.last"); }
}

 class B {
   synchronized void bar(A a) {
   String name = Thread.currentThread().getName();
   System.out.println(name + " entered B.bar");
   try { Thread.sleep(1000); }
catch(Exception e) { System.out.println("B Interrupted"); }
System.out.println(name + " trying to call A.last()"); a.last(); }
synchronized void last() { System.out.println("Inside A.last"); }

}


class Deadlock implements Runnable
 {
  A a = new A(); B b = new B();
  Deadlock( ) {
    Thread.currentThread().setName("MainThread");
    Thread t = new Thread(this, "RacingThread");
     t.start(); a.foo(b); // get lock on a in this thread.
     System.out.println("Back in main thread");
```

```
  }
 public void run() { b.bar(a); // get lock on b in other thread.
  System.out.println("Back in other thread");
  }
public static void main(String args[]) { new Deadlock(); }
}
```

```
        public static void main(String[] args)
        {
                DivisionMain1 dm=new
                DivisionMain1(); dm.setSize(new
                Dimension(800,400));
                dm.setTitle("DivisionOfIntegers");
                dm.setVisible(true);
        }
}
```