VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SHIKHA SINGH
(1BM21CS202)

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
Oct 2022-Feb 2023

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 lab**" carried out by **SHIKHA SINGH(1BM21CS202),**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 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Lab **- Object oriented java programming (21CS3PCOOJ)** work prescribed for the said degree.

Vikranth Bm
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

Course Outcome

CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyze the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

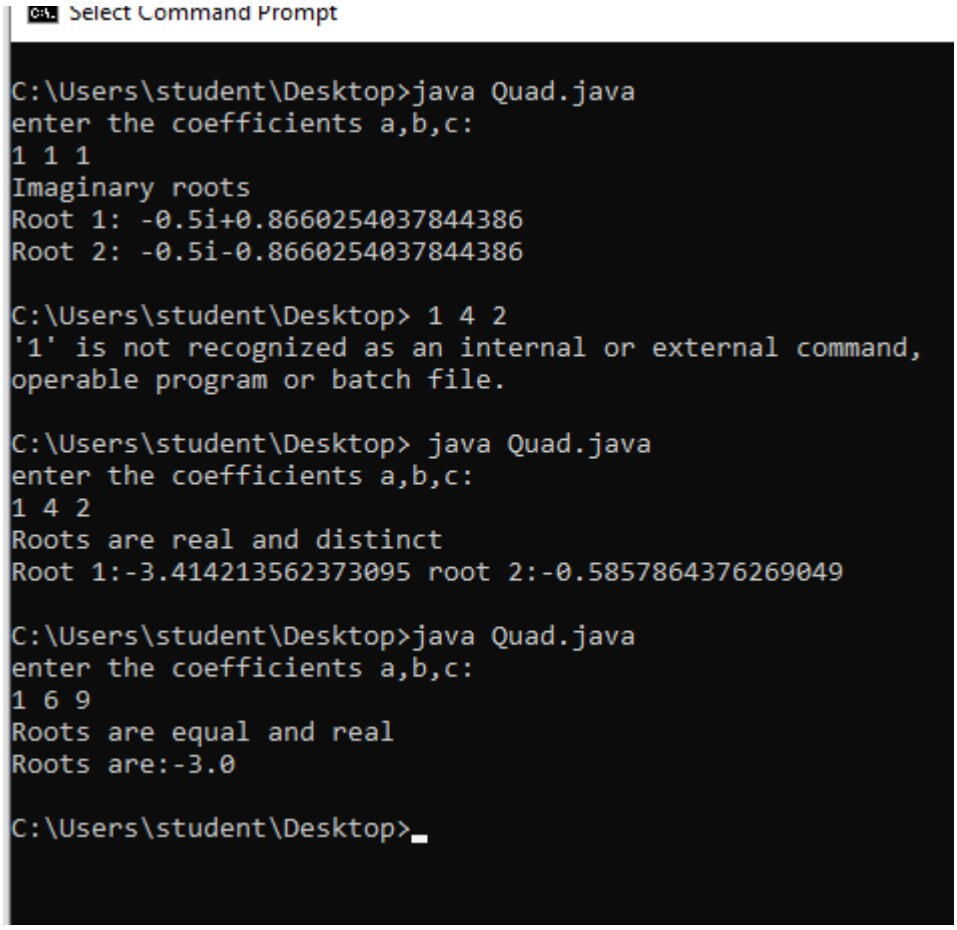
LAB PROGRAM 1: Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

CODE:

```
import java.util.Scanner;
import java.lang.Math;
public class Trial
{
   public static void main(String[] args)
      {
         Scanner s = new Scanner(System.in);
         System.out.println("Enter the coefficients: ");
         float a = s.nextFloat();
         float b = s.nextFloat();
         float c = s.nextFloat();
         double r1,r2;
         float d = (b*b)-(4.0f*a*c);
         if(d>0)
         {
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("Roots are Real");
            System.out.println("Root 1: "+r1+" Root 2: "+r2);
         }
          else if(d==0)
         {
            r1=(-b)/(2*a);
            System.out.println("Roots are Equal");
            System.out.println("Root is: "+r1);


      else
         {
```

```
            double e =(-b)/(2.0f*a);
            double f =(Math.sqrt(-d))/(2*a);
            System.out.println("Roots are imaginary");
            System.out.println("Root 1: "+e+"i+"+f);
            System.out.println("Root 2: "+e+"i-"+f);
        }
    }
}
```

OUTPUT:



LAB PROGRAM 2: Develop a Java program to create a class Student with members usn, name, anarray credits and an array marks. Include methods to accept and display detailsand a method to calculate SGPA of a student.

CODE:

```
import java.util.Scanner;

class Student

{

        String  USN;

        String name;

        int[] credits = new int[20];

        int[] marks = new int[20];

        void input(int n)

        {

        Scanner s = new Scanner(System.in);

        System.out.print("Enter Student USN: ");

        USN = s.nextLine();

        System.out.print("Enter Student Name: ");

        name = s.nextLine();

        for(int i=0;i<n;i++)

        {

        System.out.print("Enter the Subject "+(i+1)+" marks and credits
respectively: ");
```

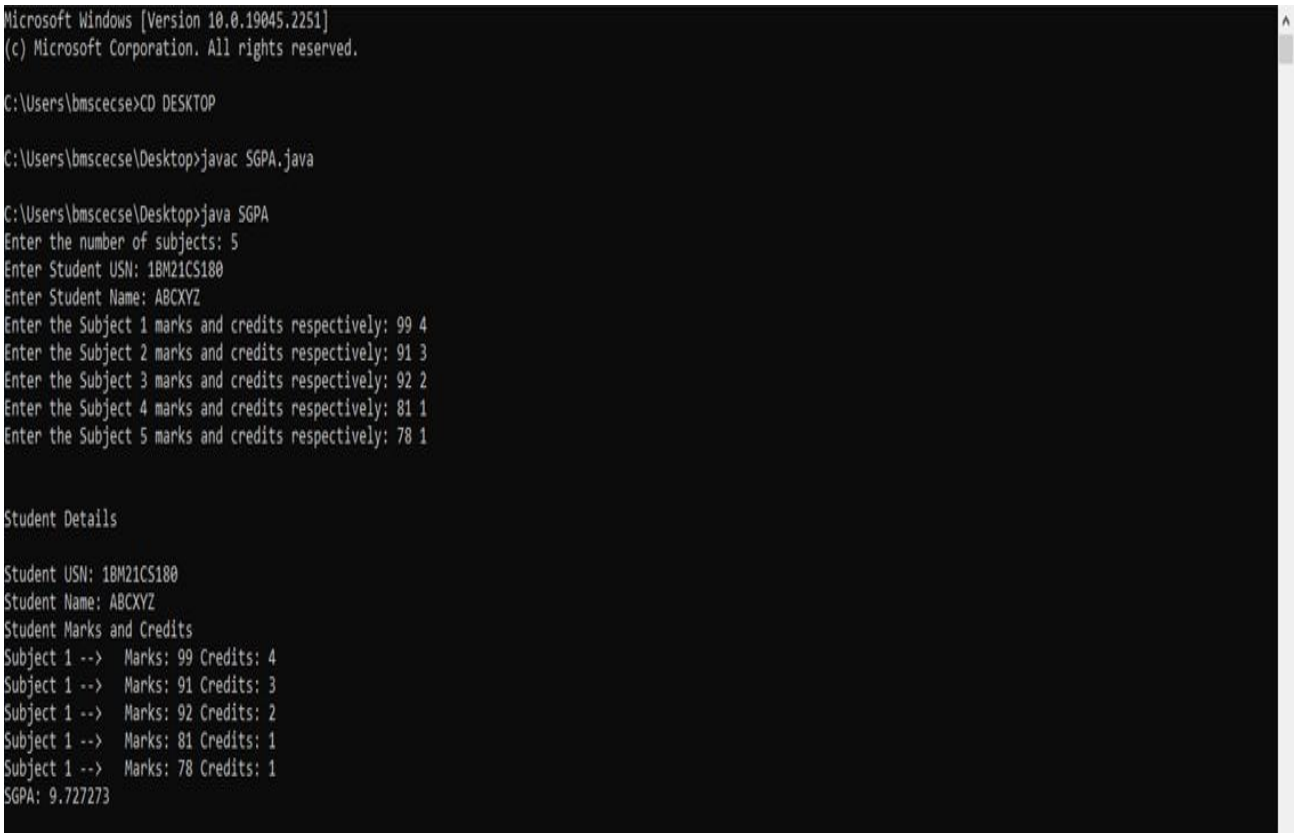
```
marks[i] = s.nextInt();
credits[i] = s.nextInt();
}

}

float calculate(int n)

{

int sum_of_credits = 0;

float result=0.0f;

for(int i=0;i<n;i++)

{

sum_of_credits+=credits[i];

 if(calculate_grade_point(marks[i])==-1)

        return -1.0f;

else

{

        result = result +(float) (calculate_grade_point(marks[i])*credits[i]);

}

}


return (result/sum_of_credits);

}


int calculate_grade_point(int marks)
```

```
{
if(marks>=90)
return 10;
else if ((marks>=80)&&(marks<90))
return 9;
else if ((marks>=70)&&(marks<80))
return 8;
else if ((marks>=60)&&(marks<70))
return 7;
else if ((marks>=50)&&(marks<60))
return 6;
else if ((marks>=40)&&(marks<50))
return 5;
return -1;
}

void display(int n,float result)
{
System.out.println("\n");
System.out.println("Student Details");
System.out.println();
```

```
        System.out.println("Student USN: "+USN);
        System.out.println("Student Name: "+name);
        System.out.println("Student Marks and Credits");
        for(int i=0;i<n;i++)
        {
        System.out.println("Subject 1 -->\tMarks: "+marks[i]+" Credits:
"+credits[i]);
        }
        System.out.println("SGPA: "+result);
        }
}
public class Lab_02_SGPA
{
        public static void main(String[] args)
        {
        Scanner s = new Scanner(System.in);
        Student s1 = new Student();
        System.out.print("Enter the number of subjects: ");
        int n = s.nextInt();
        s1.input(n);
        float result = s1.calculate(n);
        if(result == -1.0f)
```

```
        {
        System.out.println();
        System.out.println("The Student has failed in a subject. SGPA cannot be calculated!");
        System.exit(0);
        }
        s1.display(n,result);
        }
}
```

OUTPUT:



LAB 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 tocreate n book objects.

CODE:

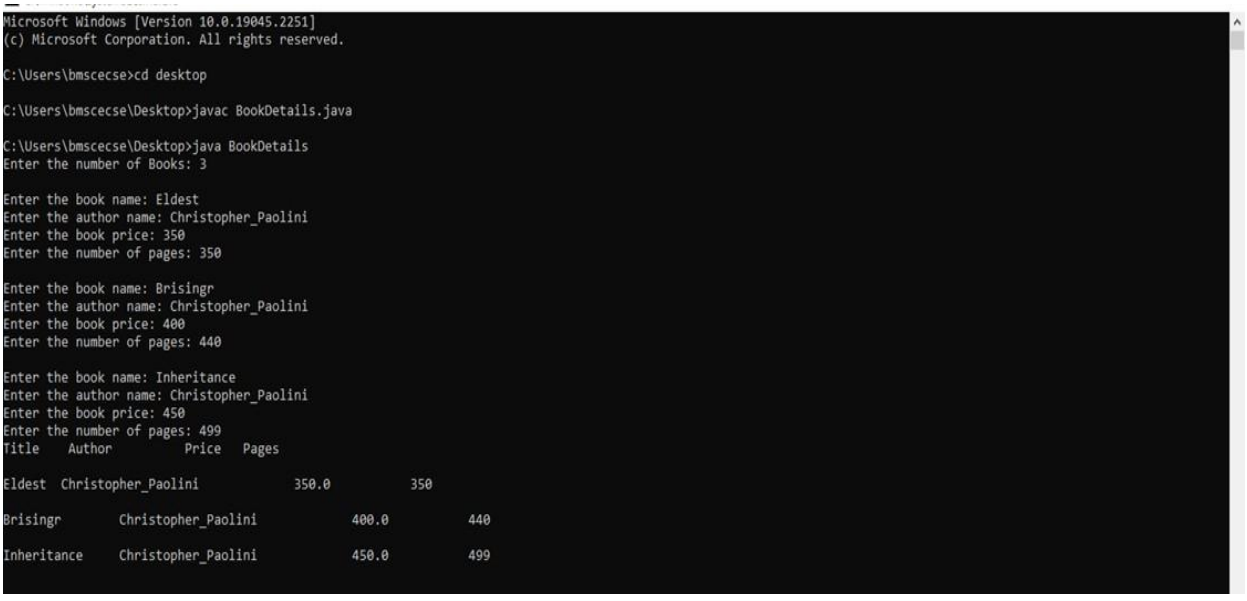
```
import java.util.*;
import java.io.*;

class Book
{
String title,author;
float price;
int num_pages;
Book()
{
title = "Default Value";
author = "Default Value";
price = 0.0f;
num_pages = 0;
}
void setTitle(String title)
{
```

```
this.title=title;
}

void setAuthor(String author)
{
this.author=author;
}

void setPrice(float price)
{
this.price=price;
}

void setPages(int num_pages)
{
this.num_pages = num_pages;
}

public String toString()
{
return title+"\t\t"+author+"\t\t"+price+"\t\t"+num_pages+"\n";
  }
  }
```

```
public class BookDetails

{

public static void main(String args[])

{

String t, a;

float p;

int np,n;

Scanner s = new Scanner(System.in);

System.out.print("Enter the number of Books: ");

n = s.nextInt();

Book[] b = new Book[n];

for(int i=0;i<n;i++)

{

System.out.println();

System.out.print("Enter the book name: ");

t = s.next();

System.out.print("Enter the author name: ");

a = s.next();

System.out.print("Enter the book price: ");

p = s.nextFloat();

System.out.print("Enter the number of pages: ");
```

```
np = s.nextInt();
b[i] = new Book();
b[i].setTitle(t);
b[i].setAuthor(a);
b[i].setPrice(p);
b[i].setPages(np);
}
System.out.println("Title \t\t Author \t\t Price \t\t
Pages\n");for(int i=0; i<n;i++)
{
System.out.println(b[i]);
}
}
}
```

OUTPUT:



LAB PROGRAM 4: Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

CODE:

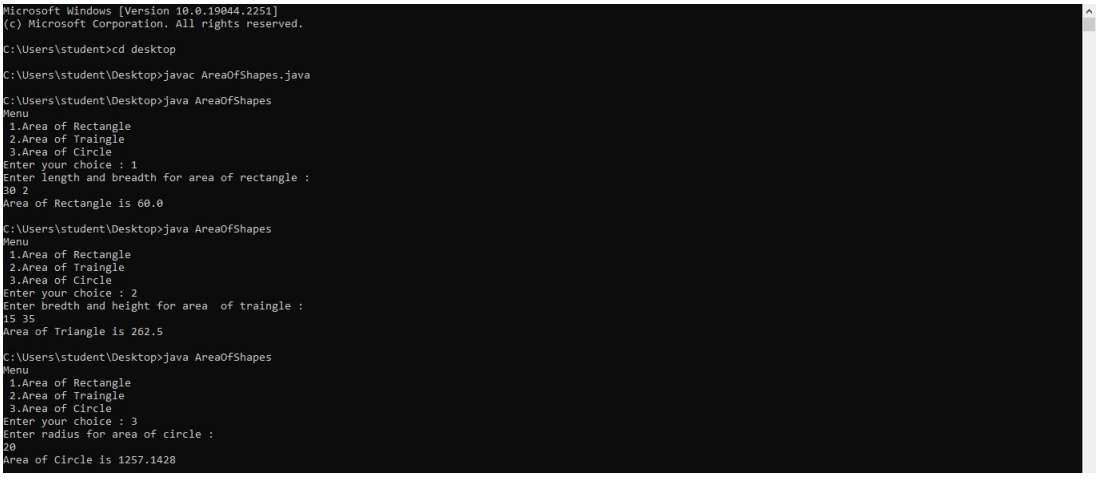
```
import java.util.Scanner;
public class Shape1
{
	public static void main(String args[])
	{
	int choice;
	Scanner s = new Scanner(System.in);
	do
	{
	System.out.println("1. Calculate Area of Rectangle\n2. Calculate Area of Triangle\n3. Calculate Area of " +
		"Circle\n4. Exit the Program\n\nEnter the choice: ");
	choice = s.nextInt();
	switch(choice)
	{
		case 1: Rectangle r = new Rectangle();
		r.printArea();
		break;
```

```
                case 2: Triangle t = new Triangle();
                t.printArea();
                break;
                case 3: Circle c = new Circle();
                c.printArea();
                break;
                case 4: System.out.println("Exiting the program!");
                System.exit(0);
                break;
                default: System.out.println("\nInvalid Choice!\n");
        }
        }while(true);
        }
}

abstract class Shape
{
        int a,b;
        abstract void printArea();
}

class Rectangle extends Shape
{
```

```
	void printArea()
	{
	int area;
	Scanner s = new Scanner(System.in);
	System.out.println("Enter the length and breadth of rectangle: ");
	a = s.nextInt();
	b = s.nextInt();
	area = a*b;
	System.out.println("\nArea of Rectangle: "+area+"\n");
	}
}

class Triangle extends Shape
{
	void printArea()
	{
	float area;
	Scanner s = new Scanner(System.in);
	System.out.println("Enter the base and height of triangle: ");
	a = s.nextInt();
	b = s.nextInt();
	area = 0.5f*a*b;
```

```
        System.out.println("\nArea of triangle: "+area+"\n");
        }
}
class Circle extends Shape
{
        void printArea()
        {
        double area;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the radius of circle: ");
        a = s.nextInt();
        area = Math.PI*a;
        System.out.println("Area of Circle: "+area+"\n");
        }
}
```

OUTPUT:



LAB PROGRAM 5: Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque bookfacility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to makethem more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update thebalance.

CODE:

```
import java.util.Scanner;
class Account
{
   String customer_name;
   long acc_no;
   float bal;
   Scanner s = new Scanner(System.in);
   public void input()
   {
       System.out.print("\nEnter the Customer Name: ");
       customer_name = s.nextLine();
```

```
        System.out.print("\nEnter the Account Number:");
        acc_no = s.nextLong();
    System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000): ");

        bal = s.nextFloat();
        if(bal<5000f)
        {
            System.out.println("\nAccount Balance cannot be less than 5000.0 \n");
            System.exit(0);
        }
    }

          public void display()
    {
        System.out.println("\nCustomer Name: "+customer_name);
        System.out.println("Account Number: "+acc_no);
        System.out.println("Amount: "+bal);
    }
}

class Savings extends Account
{
    Scanner s = new Scanner(System.in);
    float deposit,withdraw,interest;
    public void deposit()
    {
```

```
        System.out.print("\nEnter the amount to be deposited: ");
        deposit = s.nextFloat();
        bal+=deposit;
        System.out.println("\nBalance: "+bal);
    }
    public void withdraw()
    {
        System.out.print("\nEnter the amount to be withdrawn: ");
        withdraw = s.nextFloat();
        if(bal<5000)
        {
            System.out.println("\nInsufficient Balance");
        }
        else
        {
            bal-=withdraw;
            System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance:
"+bal);
        }
    }
```

```
  public void check_Bal()
   {
      if(bal<5000)
      {
         System.out.println("\nInsufficient Balance!!\nBalance: "+bal);
      }
      else
      {
         System.out.println("\nBalance: "+bal);
      }
   }
   public void interest()
   {
      interest=(bal*6)/100;
      bal+=interest;
      System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;
   }
}
class Current extends Account
{
   float deposit, withdraw, penalty;
   public void deposit()
   {
      System.out.print("\nEnter Amount to be deposited: ");
```

```
        deposit = s.nextFloat();
        bal += deposit;
        System.out.println("Balance: " + bal);
    }

    public void check_Bal()
    {
        if (bal < 5000)
        {
            penalty = (0.1f * bal);
            System.out.println("\nInitial Account Balance: "+bal);
            bal = bal-penalty;
            System.out.println("\nLow balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal);
        }
        else
            {
            System.out.println("\n Balance: " + bal);
        }
    }

     public boolean check_Bal_part_2()
    {
        if (bal < 5000)
        {
            penalty = (0.1f * bal);
```

```
            System.out.println("\nInitial Account Balance: "+bal);
            bal = bal-penalty;
            System.out.println("\nLow Balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal);
            return false;
        }
        return true;
    }

    public void withdraw()
    {
        System.out.print("\nEnter Amount to withdraw: ");
        withdraw = s.nextFloat();
        if(check_Bal_part_2())
        {
            bal-=withdraw;
                    System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance:
"+bal);
        }
    }
            public void chequebook()
    {
        System.out.println("\nCheque Book has been Issued!");
    }
```

```
public class Bank
{
   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
      String ch;
      int n;
      Current c = new Current();
      Savings sa = new Savings();
      System.out.print("\nEnter the Account Type (S for Savings , C for Current) :
");
      ch = s.next();

      switch(ch.toLowerCase())
```

```
        {
            case "s" :  sa.input();
                        do
                        {
        System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4.
Check Interest"+"\n5. Show Account Details \n6. Exit Transaction\n\nEnter
        your choice: ");
                            n = s.nextInt();
                            switch(n)
                            {
                               case 1 : sa.deposit();
                                        break;
                               case 2 : sa.withdraw();
                                        break;
                               case 3 : sa.check_Bal();
                                        break;
                               case 4 : sa.interest();
                                        break;
                               case 5 : sa.display();
                                        break;
                               case 6 : System.out.println("\nExiting Transaction!");
                                        System.exit(0);
                                        break;
                               default : System.out.println("\nInvalid Operation");
                            }
```

```
                }while(true);
            case "c" : c.input();
                do {
                    System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check
Balance \n4. Issue Cheque Book"
                        + "\n5. Show Account Details \n6. Exit Transaction\n\nEnter
your choice: ");
                    n = s.nextInt();
                    switch (n) {
                        case 1:
                            c.deposit();
                            break;
                        case 2:
                            c.withdraw();
                            break;
                        case 3:
                            c.check_Bal();
                            break;
                        case 4:
                            c.chequebook();
                            break;
                        case 5:
                            c.display();
                            break;
                        case 6:
                            System.out.println("\nExiting Transaction!");
```

```
                System.exit(0);
                        break;
                    default:
                        System.out.println("\nInvalid Operation");
                }
            }while(true);
        default : System.out.println("\nInvalid Choice");
                break;
        }
    }
}
```

OUTPUT:

```
Insufficient Balance!!
Balance: 4500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 4

Interest Credited: 270.0
Balance :4770.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 5

Customer Name: Rashtri km
Account Number: 12345678
Amount: 4770.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 6

Exiting Transaction!

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : c

Enter the Customer Name: rashtri km
```

```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd desktop

C:\Users\student\Desktop>javac Bank.java

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : s

Enter the Customer Name: Rashtri km

Enter the Account Number: 12345678

Enter the Starting Amount (Minimum Amount = 5000): 5500

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 1000

Invalid Operation

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 1

Enter the amount to be deposited: 1000

Balance: 6500.0

1. Deposit
2. Withdrawal
3. Check Balance
```

```
1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 3

 Balance: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 4

Cheque Book has been Issued!

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 5

Customer Name: rashtri km
Account Number: 123456789
Amount: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 6
```

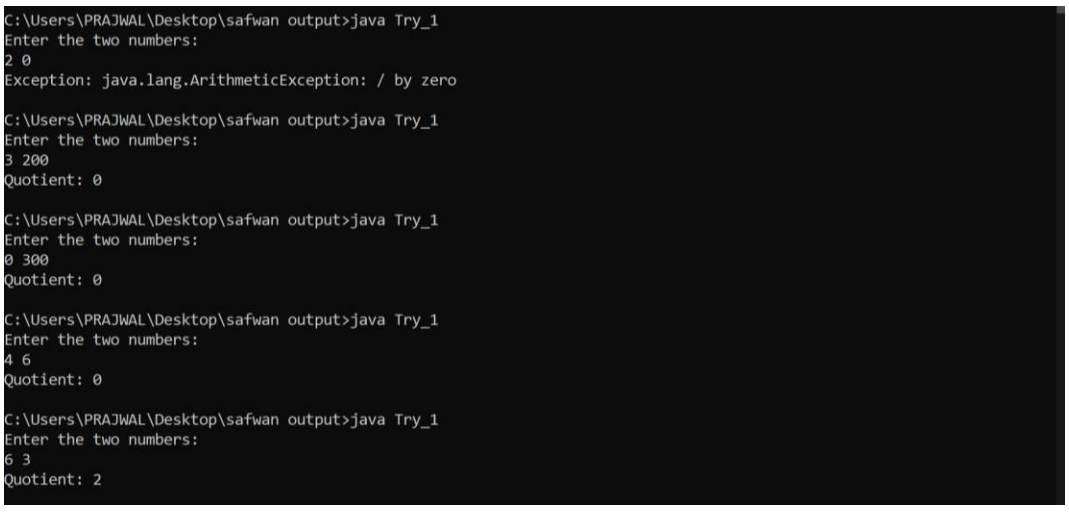
LAB PROGRAM 6: Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

CODE:

```
import java.util.InputMismatchException;
import java.util.Scanner;
interface Z
{
   public int calc(int a,int b);
}
class Y implements Z
{
   public int calc(int a, int b)
   {
      int c = a/b;
      return c;
   }
}

public class Try_1
{
   public static void main(String[] args)
   {
      Scanner s = new Scanner(System.in);
```

```
        Y o = new Y();
        int num1,num2;
        try
        {
            System.out.println("Enter the two numbers: ");
            num1 = s.nextInt();
            num2 = s.nextInt();
            int c = o.calc(num1,num2);
            System.out.println("Quotient: "+c);
        }
        catch(ArithmeticException | InputMismatchException e1)
        {
            System.out.println("Exception: "+e1);
        }
    }
}
```

OUTPUT:



LAB PROGRAM 7: Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes theage and throws the exception WrongAge( ) when the input age=father’s age.

CODE:

```
import java.util.Scanner;


public class Age
{
	public static void main(String[] args) throws WrongAge,InvalidAge
	{
	new Son();
	}
}
class WrongAge extends Exception
{
	public String getMessage()
	{
	return "Age Cannot Be Negative";
	}
}
class InvalidAge extends Exception
```

```
{
        public String getMessage()
        {
        return "Son's Age cannot be greater than Father's!";
        }
}
class Father
{
        Scanner s = new Scanner(System.in);
        int f;
        Father() throws WrongAge
        {
        System.out.print("Enter the Father's Age: ");
        f = s.nextInt();
        try
        {
        if(f<0)
        throw new WrongAge();
        }
        catch(WrongAge e1)
        {
          System.out.println(e1.getMessage());
```

```
        System.exit(0);
        }
        }
}

class Son extends Father
{
        int son;
        Son() throws WrongAge,InvalidAge
        {
        super();
        System.out.print("Enter the Son's Age: ");
        son = s.nextInt();
        try
        {
        if(son<0)
        throw new WrongAge();
        }
        catch(WrongAge e2)
        {
          System.out.println(e2.getMessage());
            System.exit(0);
```

```
        }
        try
        {
        if(son>f)
        throw new InvalidAge();
        }
        catch(InvalidAge e3)
        {
          System.out.println(e3.getMessage());
            System.exit(0);
        }
        System.out.println("Ages are appropriate");
        }
}
```

OUTPUT:

LAB PROGRAM 8: Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

CODE:

```
class MyThread extends Thread
{
        long time;
        private volatile boolean running = true;
        MyThread(){
        System.out.println("Default");
}
MyThread(String name, long time)
{
        super(name);
        this.time = time;
}
public void pause()
{
        running = false;
}
public void run()
{
        try
        {
                while(running)
                {
```

```
                System.out.println(this.getName());
                Thread.sleep(time*1000);
            }
        }
        catch(InterruptedException ie)
        {
            System.out.println("Exception caught in method");
        }

}
}

class Main
{
    public static void main(String [] args)
    {
        MyThread mt1 = new MyThread("BMS", 10);
        MyThread mt2 = new MyThread("CSE", 2);
        mt1.start();
        mt2.start();
        Try
        {
            Thread.sleep(20*1000);
            mt1.pause();
            mt2.pause();
        }
```

```
            catch(InterruptedException ie)
            {
                System.out.println("Exception caught in main");
            }
        }
}
```

OUTPUT: