

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

### OBJECT ORIENTED JAVA PROGRAMMING (21CS3PCOOJ)

*Submitted by*

**Tanisha Gotadke  
(1BM21CS229)**

*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**

**October-2022 to 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 “Database Management Systems (22CS3PCDBM)” carried out by **TANISHA GOTADKE (1BM21CS229)** 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. The Lab report has been approved as it satisfies the academic requirements in respect of a Database Management Systems (22CS3PCDBM) work prescribed for the said degree.

**Dr. Nandini Vineeth**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## INDEX

Sl No.	Date	Experiment Title	Page No.
1.	17/11/2022	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.	5
2.	24/11/2022	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.	8
3.	01/12/2022	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.	11
4.	08/12/2022	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.	14
5.	29/12/2022	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 book facility 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-	17

		<p>acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:</p> <p>a) Accept deposit from customer and update the balance.</p> <p>b) Display the balance.</p> <p>c) Compute and deposit interest</p> <p>d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.</p>	
6.	12/01/2023	<p>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.</p>	23
7.	05/01/2023	<p>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 the age and throws the exception WrongAge( ) when the input age&lt;0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is &gt;=father's age.</p>	27
8.	12/01/2023	<p>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</p>	31

## EXPERIMENT-1

**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.*;
import java.lang.Math.*;
public class quadratic
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);
        System.out.println("Enter a");
        double a=in.nextDouble();
        System.out.println("Enter b");
        double b=in.nextDouble();
        System.out.println("Enter c");
        double c=in.nextDouble();

        if(a==0)
        {
            System.out.println("Invalid inputs\n");
        }
        else
        {
            double d=b*b-4*a*c;
            if(d>0.0)
            {
                double r1=(-b+Math.pow(d,0.5))/(2.0*a);
                double r2=(-b+Math.pow(d,0.5))/(2.0*a);
                System.out.println("Roots are real and distinct\n Roots are \n r1="+r1+"and"+r2);
            }
            else if(d==0.0)
            {
                double r1=-b/(2*a);
                System.out.println("Roots are real and equal and each root is equal to" +r1);
            }
            else
            {
                System.out.println("Roots are imaginary and distinct.\n Roots are\n");
                double r1=-b/(2.0*a);
                double r2=(Math.sqrt(Math.abs(d)))/(2.0*a);

                System.out.println("r1="+r1+"+i"+r2+"\n"+"r2="+r1+"-i"+r2);
            }
        }
    }
}
```

## SAMPLE OUTPUTS:

1. Roots are Real and Equal

```
C:\Users\BMSCECSE\Desktop\1BM21CS229>java quadratic
Enter a
2
Enter b
4
Enter c
2
Roots are real and equal and each root is equal to-1.0
```

2. Roots are real and distinct

```
C:\Users\BMSCECSE\Desktop\1BM21CS229>java quadratic
Enter a
1
Enter b
0
Enter c
-1
Roots are real and distinct
Roots are
r1=1.0and1.0
```

3. Roots are imaginary and distinct

```
C:\Users\BMSCECSE\Desktop\1BM21CS229>java quadratic
Enter a
1
Enter b
2
Enter c
3
Roots are imaginary and distinct.
Roots are

r1=-1.0+i1.4142135623730951
r2=-1.0-i1.4142135623730951
```

4. Invalid Outputs

```
Enter a
2
Enter b
4
Enter c
2
Roots are real and equal and each root is equal to-1.0
```

## EXPERIMENT-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.**

```
import java.util.Scanner;

class student{

void display(String name, String usn)

{

System.out.println("USN of the student "+usn);

    System.out.println("Name of the student "+ name);

}

void calculategpa(double[] marks, double[] credits, int number)

{

double gradepoints[]=new double[number];

double sgpa,sum=0,tnum=0;

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

{

if(marks[i]>=90)

gradepoints[i]=10;

else if(marks[i]>=80)

gradepoints[i]=9;

else if(marks[i]>=70)

gradepoints[i]=8;

else if(marks[i]>=60)

gradepoints[i]=7;

else if(marks[i]>=50)

gradepoints[i]=6;

else if(marks[i]>=40)

gradepoints[i]=4;

else

gradepoints[i]=0;
```



```

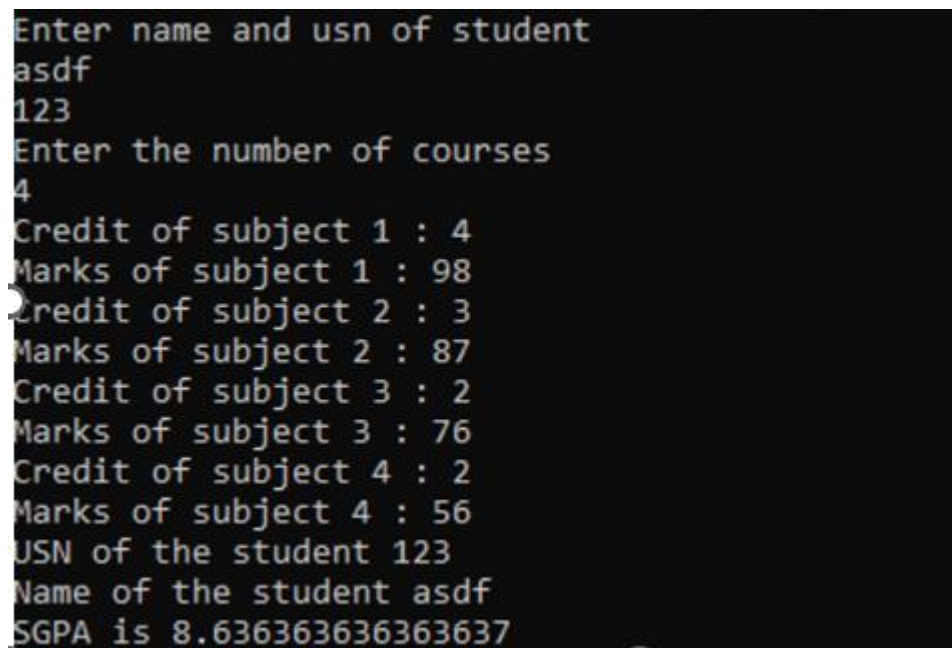
    }
    for(int i=0;i<number;i++)
    {
        sum+=credits[i]*gradepoints[i];
    }
    for(int i=0;i<number;i++)
    {
        tnum+=credits[i];
    }
    sgpa=sum/tnum;
    System.out.println("SGPA is "+sgpa);
}
}

class sgpa{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter name and usn of student");
        String name=s.next();
        String usn=s.next();
        student s1=new student();
        System.out.println("Enter the number of courses");
        int number=s.nextInt();
        double credits[]=new double[number];
        double marks[]=new double[number];
        for(int i=0;i<number;i++)
        {
            System.out.print("Credit of subject "+(i+1) +" : ");
            credits[i]=s.nextDouble();
            System.out.print("Marks of subject "+(i+1) +" : ");

```

```
marks[i]=s.nextDouble();  
}  
s1.display(name,usn);  
s1.calculatesgpa(marks,credits,number);  
}  
}
```

### **SAMPLE OUTPUT:**



```
Enter name and usn of student  
asdf  
123  
Enter the number of courses  
4  
Credit of subject 1 : 4  
Marks of subject 1 : 98  
Credit of subject 2 : 3  
Marks of subject 2 : 87  
Credit of subject 3 : 2  
Marks of subject 3 : 76  
Credit of subject 4 : 2  
Marks of subject 4 : 56  
USN of the student 123  
Name of the student asdf  
SGPA is 8.636363636363637
```

### EXPERIMENT-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.**

```
import java.util.*;
import java.lang.*;
class Book
{
    String name,author; int price,num_pages;

    void getval()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter book name");
        name=sc.next();
        System.out.println("Enter author name");
        author=sc.next();
        System.out.println("Enter price ");
        price=sc.nextInt();
        System.out.println("Enter No. of pages");
        num_pages=sc.nextInt();
    }
    public String toString()
    {
        return name+" "+author+" "+price+" "+num_pages+" ";
    }
}
```

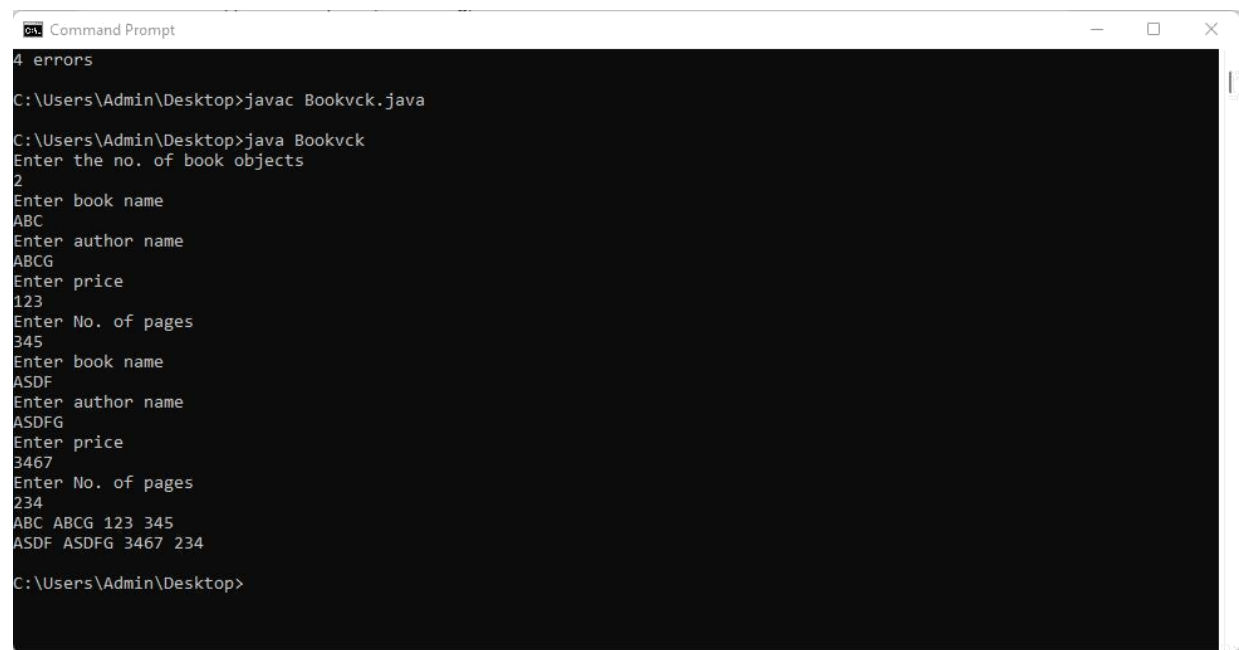
```

    void display()
    {
        System.out.println(this);
    }
}
class Bookvck
{
    public static void main(String args[])
    {
        Scanner in=new Scanner(System.in);

        System.out.println("Enter the no. of book objects");
        int n=in.nextInt();
        Book[] ob=new Book[n];
        for(int i=0;i<n;i++)
            ob[i]=new Book();
        for(int i=0;i<n;i++)
            { ob[i].getval();}
        for(int
            i=0;i<n;i++) {
            ob[i].display();}
        }
    }
}

```

## SAMPLE OUTPUT:



```
4 errors
C:\Users\Admin\Desktop>javac Bookvck.java
C:\Users\Admin\Desktop>java Bookvck
Enter the no. of book objects
2
Enter book name
ABC
Enter author name
ABCG
Enter price
123
Enter No. of pages
345
Enter book name
ASDF
Enter author name
ASDFG
Enter price
3467
Enter No. of pages
234
ABC ABCG 123 345
ASDF ASDFG 3467 234
C:\Users\Admin\Desktop>
```

## EXPERIMENT 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.lang.*;

abstract class Shape{
    int a,b;
    double area;final double pi=3.142;
    Shape(int x,int y) {
        a=x;b=y;area=0;
    }
    abstract void printArea();
}

class Rectangle extends Shape
{
    Rectangle(int x,int y)
    {
        super(x,y);
    }
    void printArea()
    {
        area=a*b;
        System.out.println("Rectangle area="+area);
    }
}

class Triangle extends Shape
```

```

{
    Triangle(int x,int y)
    {
        super(x,y);
    }
    void printArea()
    {
        area=a*b*0.5;
        System.out.println("Triangle area="+area);
    }
}

```

```

class Circle extends Shape
{
    Circle(int x)
    {
        super(x,-1);
    }
    void printArea()
    {
        area=pi*Math.pow(a,2);
        System.out.println("Circle area="+area);
    }
}

```

```

class demoshape1 {
    public static void main(String args[])
    {
        Rectangle r1=new Rectangle(1,2);
        Triangle t1=new Triangle(1,2);
        Circle c1=new Circle(5);
    }
}

```

```
    Shape ref;  
    ref=r1;ref.printArea();  
    ref=t1;ref.printArea();  
    ref=c1;ref.printArea();  
}  
}
```

SAMPLE OUTPUT:

```
Rectangle area=2.0  
Triangle area=1.0  
Circle area=78.55
```



## EXPERIMENT 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 book facility 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 make them 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 the balance.**

**Code :**

```
import java.util.Scanner;

class account {
    String name;
    int account_num;
    String acc_type;
}

class sav_acct extends account {
    double balance;

    sav_acct(String n, int ac, String actype, Double bl) {
        name = n;
        account_num = ac;
        actype = acc_type;
        balance = bl;
    }

    Scanner sc = new Scanner(System.in);

    void deposit(int val) {
        balance += val;
    }

    void display_bal() {
        System.out.println("Balance is: " + balance);
    }
}
```

```

void deposit_interest() {
    double int_rate = 0.05;
    double time = 0;
    System.out.println("enter the time period");
    time = sc.nextDouble();
    double amount;
    amount = balance * Math.pow((1 + int_rate), time);
    balance = amount;
}

void withdraw(int val) {
    if (val > balance) {
        System.out.println("out of funds, withdraw lesser");
    } else {
        balance -= val;
        System.out.println("withdrawal successful");
        System.out.println("new balance: " + balance);
    }
}

void check_min() {
    Double min_bal = 1000.00;
    Double penalty = 100.00;
    if (balance < min_bal) {
        System.out.println("balance lesser than minimum balance, penalty imposed");
        balance -= penalty;
    }
    else{
        System.out.println("balance higher than minimum balance");
    }
}

class cur_acct extends account {
    double balance;

    cur_acct(String n, int ac, String actype, Double bl) {
        name = n;
        account_num = ac;
        actype = acc_type;
        balance = bl;
    }

    void deposit(int val) {
        balance += val;
    }

    void display_bal() {
        System.out.println("Balance is: " + balance);
    }
}

```

```

    }

    void deposit_interest() {
        System.out.println("Current account doesnt provide any interest");
    }

    void withdraw(int val) {
        System.out.println("Current account doesnt provide withdrawal facility");
    }

    void check_min() {
        double min_bal = 1000.00;
        double penalty = 100.00;
        if (balance < min_bal) {
            System.out.println("balance lesser than minimum balance, penalty imposed");
            balance -= penalty;
        }
        else{
            System.out.println("balance higher than minimum balance");
        }
    }

    void cheque_withdrawal(int val) {
        this.check_min();
        balance -= val;
        System.out.println("withdrawal successful");
        System.out.println("new balance: " + balance);
    }
}

class bank {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter your name, account number, aaccount type(savings/current), balance");
        String name = sc.nextLine();
        int account_num = sc.nextInt();
        String acc_type = sc.next();
        double balance = sc.nextDouble();
        if (acc_type.equals("savings")) {
            sav_acct a1 = new sav_acct(name, account_num, acc_type, balance);
            int choice = 0;
            while (choice != 6) {
                System.out.println(
                    "1.deposit\n2.display balance\n3.compute and deposit interest\n4.withdraw\n5.check for minimum balance\n6.exit");
                choice = sc.nextInt();
                switch (choice) {
                    case (1):
                        System.out.println("enter the value to deposit");

```

```

        int val = sc.nextInt();
        a1.deposit(val);
        break;
    case (2):
        a1.display_bal();
        break;
    case (3):
        a1.deposit_interest();
        break;
    case (4):
        System.out.println("enter the value to withdraw");
        int withd = sc.nextInt();
        a1.withdraw(withd);
        break;
    case (5):
        a1.check_min();
        break;
    case (6):
        System.out.println("exited");
        break;
    default:
        System.out.println("enter a valid choice");
        break;
    }
}
} else {
    cur_acct a1 = new cur_acct(name, account_num, acc_type, balance);
    int choice = 0;
    while (choice != 6) {
        System.out.println(
            "1.deposit\n2.display balance\n3.compute and deposit interest\n4.withdraw
using cheque\n5.check for minimum balance\n6.exit");
        choice = sc.nextInt();
        switch (choice) {
            case (1):
                System.out.println("enter the value to deposit");
                int val = sc.nextInt();
                a1.deposit(val);
                break;
            case (2):
                a1.display_bal();
                break;
            case (3):
                a1.deposit_interest();
                break;
            case (4):
                System.out.println("enter the value to withdraw");
                int withd = sc.nextInt();
                a1.cheque_withdrawal(withd);
                break;

```

```
        case (5):
            a1.check_min();
            break;
        case (6):
            System.out.println("exited");
            break;
        default:
            System.out.println("enter a valid choice");
            break;
    }
}
}
```

**Sample Output :**

```

Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\Desktop\1BM21CS229

C:\Users\Admin\Desktop\1BM21CS229>set path=C:\Program Files\Java\jdk-19\bin

C:\Users\Admin\Desktop\1BM21CS229>javac classbank.java

C:\Users\Admin\Desktop\1BM21CS229>java bank
enter your name, account number, account type(savings/current), balance
tanisha
45084840
savings
3500
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
1
enter the value to deposit
650
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 4150.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
3
enter the time period
2
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 4575.375
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
4
enter the value to withdraw
575
withdrawal successful
new balance: 4000.375

```

```

new balance: 4000.375
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
4
enter the value to withdraw
678
balance higher than minimum balance
withdrawal successful
new balance: 7926.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
balance higher than minimum balance
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
4
enter the value to withdraw
7892
balance higher than minimum balance
withdrawal successful
new balance: 34.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
balance lesser than minimum balance, penalty imposed
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
6
exited

C:\Users\Admin\Desktop\1BM21CS229>

```

## EXPERIMENT 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 `ArithmeticException`. Display the exception in a message dialog box.**

CODE :

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

public class DivApplet extends JApplet implements
    ActionListener {
    JTextField number1,number2,result;
    JButton divide;

    public void init()
    {
        try
        {
            SwingUtilities.invokeLater(
                new Runnable()
                {
                    public void run()
                    {
                        makeGUI();
                    }
                }
            );
        }
        catch (Exception exc) {
            System.out.println("Can't create because of " + exc);
        }

        private void makeGUI(){
            setLayout(new FlowLayout());
```



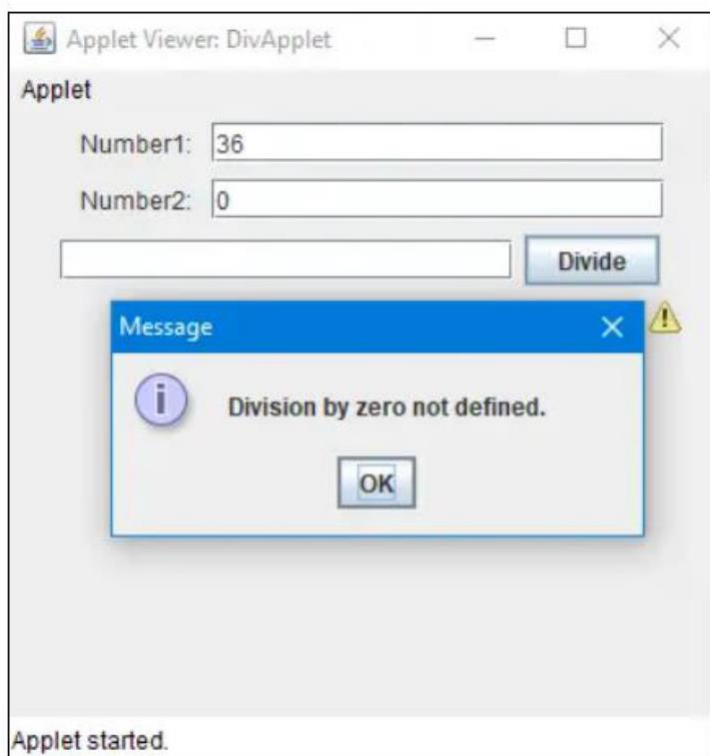
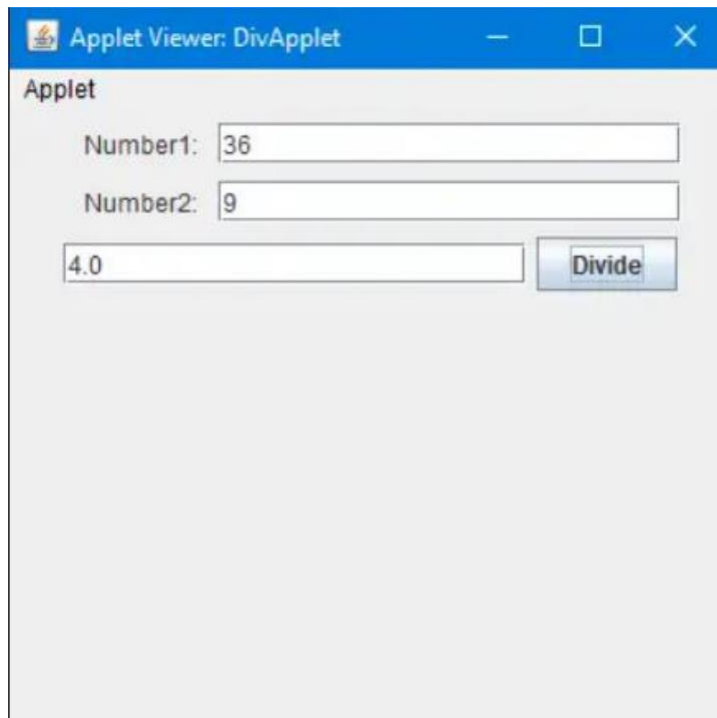
```

Label number1p = new Label("Number1:
",Label.RIGHT); Label number2p = new
Label("Number2: ",Label.RIGHT); number1= new
JTextField(20); number2 = new JTextField(20);
result = new JTextField(20);
divide = new JButton("Divide");
add(number1p);
add(number1);
add(number2p);
add(number2);
add(result);
add(divide);
divide.addActionListener(this);
}

public void actionPerformed(ActionEvent e){
String snumber1,snumber2;
snumber1 = number1.getText();
snumber2 = number2.getText();
try{
int number1 = Integer.parseInt(snumber1);
int number2 = Integer.parseInt(snumber2);
if(number2==0)
JOptionPane.showMessageDialog(null, "Division by zero not
defined."); else{
double r = (double)number1/number2;
result.setText(((Double)r).toString());
}}
catch(NumberFormatException ne)
{
JOptionPane.showMessageDialog(null,"Enter a number");
}}
}

```

OUTPUT;



## EXPERIMENT 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 the age and throws the exception WrongAge() when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.**

CODE:

```
import java.util.*;

class Wrongage extends Exception
{
    int detail;
    Wrongage(int d)
    {
        detail=d;
    }
    public String toString()
    {
        return "Entered Wrong age is [" +detail+"]";
    }
}

class Father
{
    int f;
    Scanner in=new Scanner(System.in);
    Father()
    {
        System.out.println("Enter father age ");
        f=in.nextInt();
    }
    void checkage() throws Wrongage
    {
```

```

        if(f<0)
        {
            throw new Wrongage(f);
        }
        System.out.println("Father age positive");
    }
}

class Son extends Father
{
    int s;
    Scanner in=new Scanner(System.in);
    Son()
    {
        super();
        System.out.println("Enter son age "); s=in.nextInt();
    }
    void checkages() throws Wrongage
    {
        super.checkage();
        if(s<0)
        {
            throw new Wrongage(f);
        }
        System.out.println("Son age positive");
    }
    void checkage() throws Wrongage
    {
        if(s>=f)
        {
            throw new Wrongage(s);

```

```

    }
    System.out.println("Father-Son age correct");
}
}
class Newdemo
{
    public static void main(String args[])
    {
        int f,s;
        Father fath=new Father();
        Father r;
        r=fath;
        try
        {
            r.checkage();
        }
        catch(Wrongage e)
        {
            System.out.println("Father age wrong"+e);
        }
        Son sn=new Son();
        r=sn;
        try
        {
            sn.checkages();
            r.checkage();
        }
        catch(Wrongage e)
        {
            System.out.println("Son age wrong"+e);
        }
    }
}

```

```
    }  
    }  
}
```

#### SAMPLE OUTPUT:

```
Enter father age  
-20  
Father age wrongEntered Wrong age is [-20]  
Enter father age  
25  
Enter son age  
30  
Father age positive  
Son age positive  
Son age wrongEntered Wrong age is [30]
```

```
Enter father age  
40  
Father age positive  
Enter father age  
45  
Enter son age  
30  
Father age positive  
Son age positive  
Father-Son age correct
```

```
Enter father age  
12  
Father age positive  
Enter father age  
12  
Enter son age  
12  
Father age positive  
Son age positive  
Son age wrongEntered Wrong age is [12]
```

```
Enter father age  
-12  
Father age wrongEntered Wrong age is [-12]  
Enter father age  
-89  
Enter son age  
-56  
Son age wrongEntered Wrong age is [-89]
```

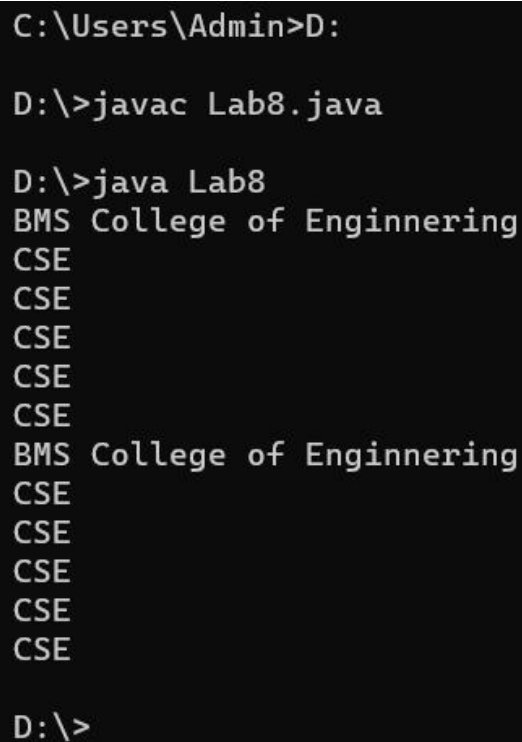
## EXPERIMENT 8

**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.**

```
class Call implements Runnable
{
String a;
int x,time;
Thread t;
Call(String tn,int ti,int ex)
{
a=tn;
x=ex;
time=ti;
t=new Thread(this,a);
t.start();
}
public void run()
{
try{
for(int i=0;i<x ;i++)
{
System.out.println(a);
Thread.sleep(time);
}
}
catch(InterruptedException ie)
{
System.out.println("Inturrupted ");
}
}
}
```

```
class Lab8
{
public static void main(String args[])
{
new Call("BMS College of
Enginnering",10000,2); new Call("CSE",2000,10);
}
}
```

SAMPLE OUTPUT:



```
C:\Users\Admin>D:
D:\>javac Lab8.java
D:\>java Lab8
BMS College of Enginnering
CSE
CSE
CSE
CSE
CSE
CSE College of Enginnering
CSE
CSE
CSE
CSE
CSE
D:\>
```