VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING
(22CS3PCOOJ)

Submitted by

SHRAVANI SHEKAR
(1BM21CS205)

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 OBJECT ORIENTED JAVA PROGRAMMING " (22CS3PCOOJ)" carried out by **SHRAVANI SHEKAR (1BM21CS205)** 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 OBJECT ORIENTED JAVA PROGRAMMING (22CS3PCOOJ) work prescribed for the said degree.

Name of the Lab-Incharge	**Dr. Jyothi S Nayak**
Vikranth B.M	
Assistant Professor	Professor and Head
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

INDEX

		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.	
6.	12/01/2023	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.	23
7.	05/01/2023	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.	26
8.	12/01/2023	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	30

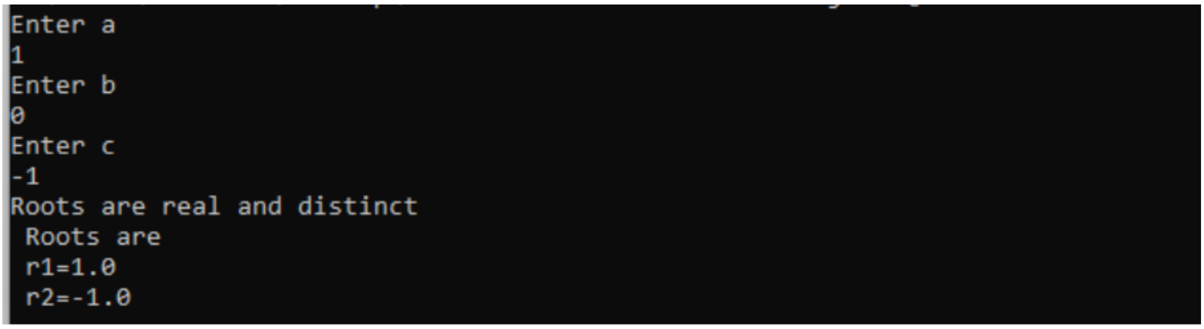
EXPERIMENT-1

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.

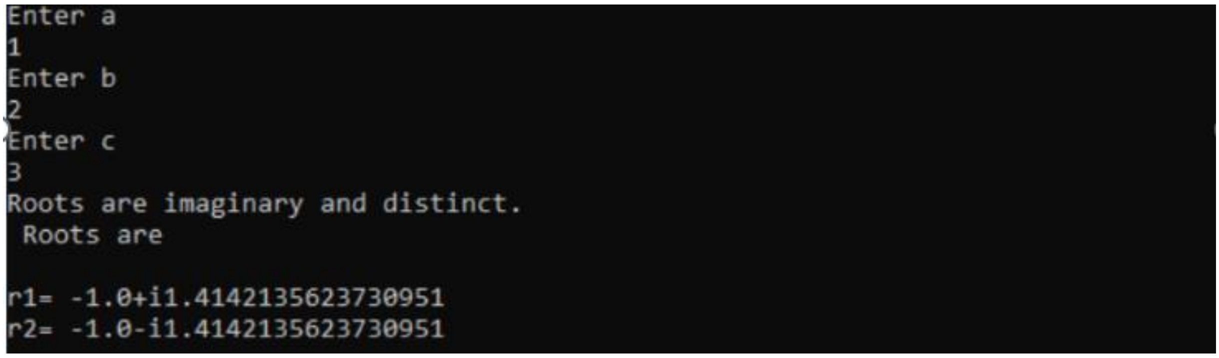
CODE:

```
import java.util.*;
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.sqrt(d)/(2.0*a)));
 double r2=(-b-(Math.sqrt(d)/(2.0*a)));
 System.out.println("Roots are real and distinct \n Roots are \n r1="+r1+"\n r2="+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);
 }
 }
}
}
```

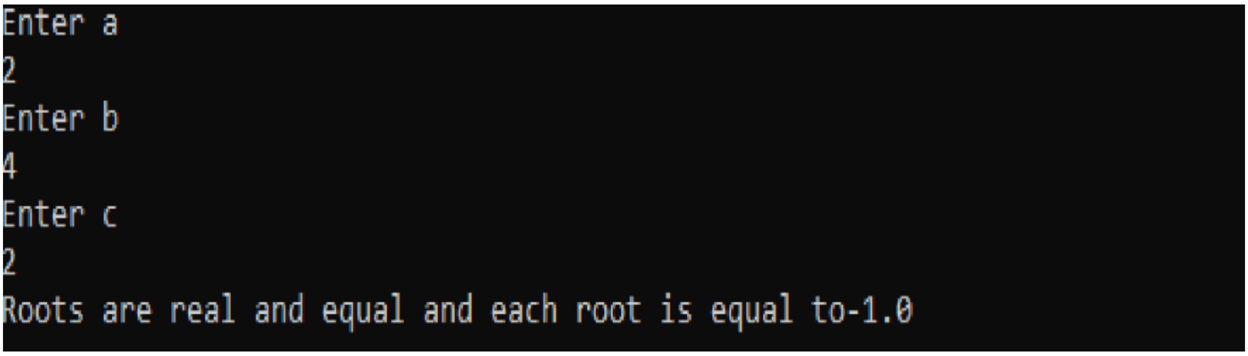
SAMPLE OUTPUTS:



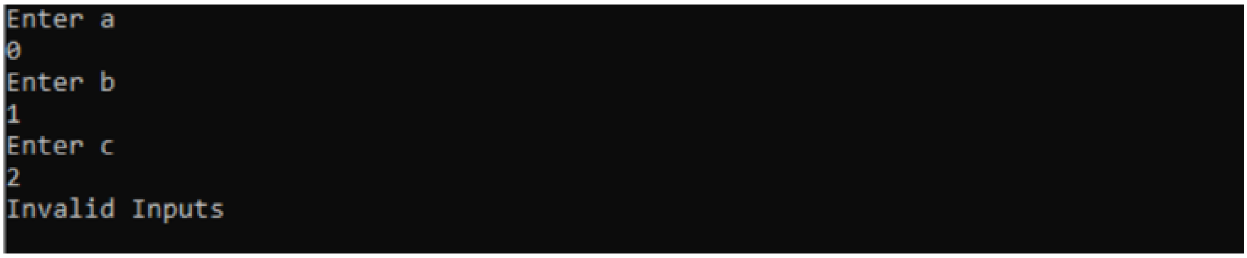
```
Enter a
1
Enter b
0
Enter c
-1
Roots are real and distinct
 Roots are
 r1=1.0
 r2=-1.0
```



```
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
```



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



```
Enter a
0
Enter b
1
Enter c
2
Invalid Inputs
```

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.

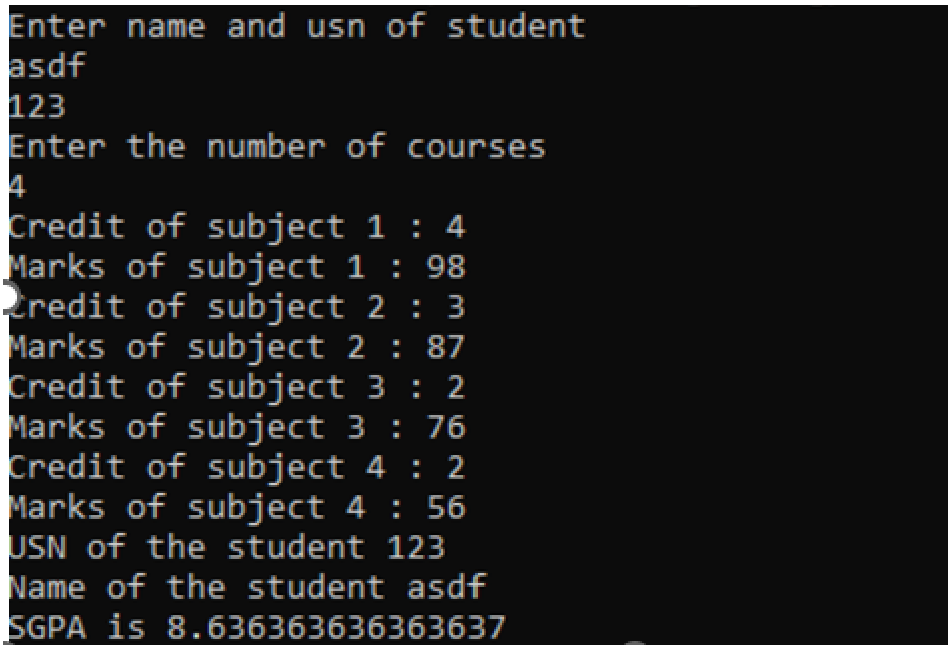
CODE:

```
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 calculatesgpa(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
```

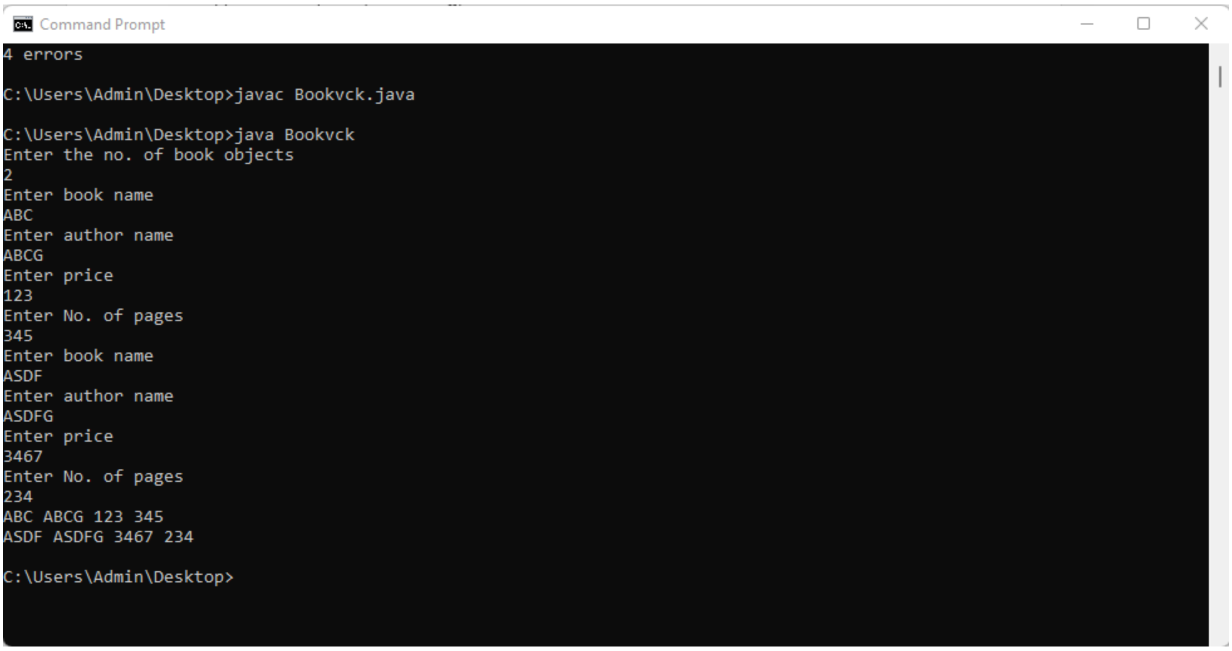
EXPERIMENT-3

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.

CODE:

```
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

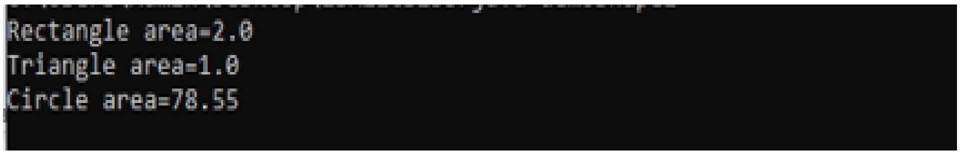
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

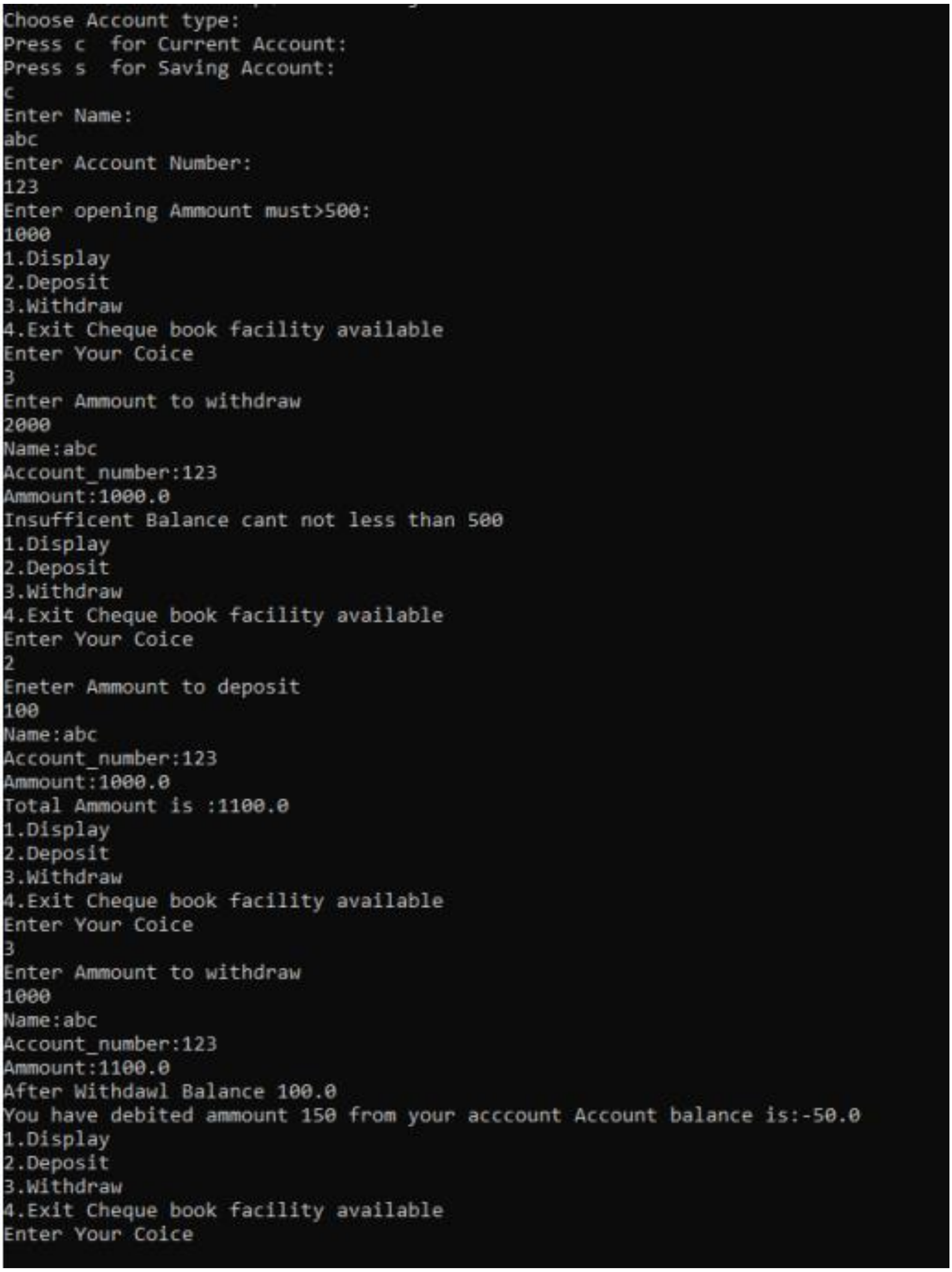
CODE:

```
import java.lang.*;
 import java.util.Scanner;
class Account
{
    public static int min=500;
    String name;
    int Account_num;
    public float o_Price;
    Scanner sc=new Scanner(System.in);
     public void get_info()
    {
        System.out.println("Enter Name:");
        name=sc.nextLine();
       System.out.println("Enter Account Number:");
       Account_num=sc.nextInt();
       System.out.println("Enter opening Ammount must>500:");
```

```
        o_Price=sc.nextFloat();
       if(o_Price <500)
      {
          System.out.println("Enter opening Ammount must>500:");
       }
     }
    public void show()
   {
      System.out.println("Name:"+name);
      System.out.println("Account_number:"+Account_num);
      System.out.println("Ammount:"+o_Price);
  }
}
class Current extends Account
 {
    float deposit,withdraw,penality;
     public void deposit()
    {
       System.out.println("Eneter Ammount to deposit") ;
       deposit =sc.nextFloat();
        show();
        o_Price=o_Price+deposit;
        System.out.println("Total Ammount is :"+o_Price) ;
     }
      public void check_Bal()
      {
        if(o_Price<min);
       o_Price=o_Price-150;
        System.out.println("You have debited ammount 150 from your acccount Account
        balance is:"+o_Price);
      }
      public void withdraw_Bal()
  {
   System.out.println("Enter Ammount to withdraw");
   withdraw=sc.nextFloat();
   show();
   if(withdraw<o_Price)
{
  o_Price=o_Price-withdraw;
  System.out.println("After withdrawal Balance "+o_Price);
  }
else
 {
  System.out.println("Insufficient balance can not be less than 500");
}
check_Bal();
}
}


class Saving extends Account
 {
 float deposit,withdraw,intr;
 public void deposit()
  {
   System.out.println("Eneter Ammount to deposit") ;
   deposit =sc.nextFloat();
   show();
   o_Price=o_Price+deposit;
    System.out.println("Total Ammount is :"+o_Price) ;
  }
 public void check_intrest()
 {
 intr=(o_Price*2)/100;
 o_Price=o_Price+intr;
 System.out.println("Total Ammount with intrest is :"+o_Price) ;
 }
 public void withdraw_Bal()
 {
  System.out.println("Enter Ammount to withdraw:");
  withdraw=sc.nextFloat();
  show();
  if(withdraw<o_Price)
{
 o_Price=o_Price-withdraw;
System.out.println("After withdrawal Balance "+o_Price);
```

```
}
else{
System.out.println("Insufficient Balance!");
}
}
}

public class Bank
{
 static String ch;
 public static void main(String[] args)
{
 int count=0;
 Scanner sc=new Scanner(System.in);
 Current cu=new Current ();
 Saving sav=new Saving ();
 System.out.println("Choose Account type:");
 System.out.println("Press c for Current Account:");
 System.out.println("Press s for Saving Account:");
 ch=sc.nextLine();
 if(ch.equalsIgnoreCase("c"))
  {
     cu.get_info();
     cu.check_Bal();
     while(count!=4)
  {
      System.out.println("1.Display\n2.Deposit\n3.Withdraw\n4.Exit");
            System.out.println("Enter Your Coice");
            int cho=sc.nextInt();
            switch(cho)
       {
         case 1: cu.show();
              break;
         case 2: cu.deposit();
                 break;
         case 3: cu.withdraw_Bal();
                 break;
         case 4: System.exit(0);
                 break;
        default:System.out.println("Wrong Choce!");
      }
    }
   }

else if(ch.equalsIgnoreCase("s"))
 {
   sav.get_info();
    while(count!=5)
      {
      System.out.println("1.Display\n2.Deposit\n3.Withdraw\n4Intrest\n5.Exit");
         System.out.println("Enter Your Coice");
         int cho=sc.nextInt();
         switch(cho)
    {
  case 1: sav.show();
           break;
  case 2: sav.deposit();
                       break;
  case 3: sav.withdraw_Bal();
                       break;
      case 4: sav.check_intrest();
                       break;
      case 5: System.exit(0);
                       break;
              default:System.out.println("Wrong Choce!");
           }
       }
  }
else
 {
  System.out.println("Wrong choice!");
 }
 }
}
```

SAMPLE OUTPUT:



```
Choose Account type:
Press c  for Current Account:
Press s  for Saving Account:
c
Enter Name:
abc
Enter Account Number:
123
Enter opening Ammount must>500:
1000
1.Display
2.Deposit
3.Withdraw
4.Exit Cheque book facility available
Enter Your Coice
3
Enter Ammount to withdraw
2000
Name:abc
Account_number:123
Ammount:1000.0
Insufficent Balance cant not less than 500
1.Display
2.Deposit
3.Withdraw
4.Exit Cheque book facility available
Enter Your Coice
2
Eneter Ammount to deposit
100
Name:abc
Account_number:123
Ammount:1000.0
Total Ammount is :1100.0
1.Display
2.Deposit
3.Withdraw
4.Exit Cheque book facility available
Enter Your Coice
3
Enter Ammount to withdraw
1000
Name:abc
Account_number:123
Ammount:1100.0
After Withdawl Balance 100.0
You have debited ammount 150 from your acccount Account balance is:-50.0
1.Display
2.Deposit
3.Withdraw
4.Exit Cheque book facility available
Enter Your Coice
```

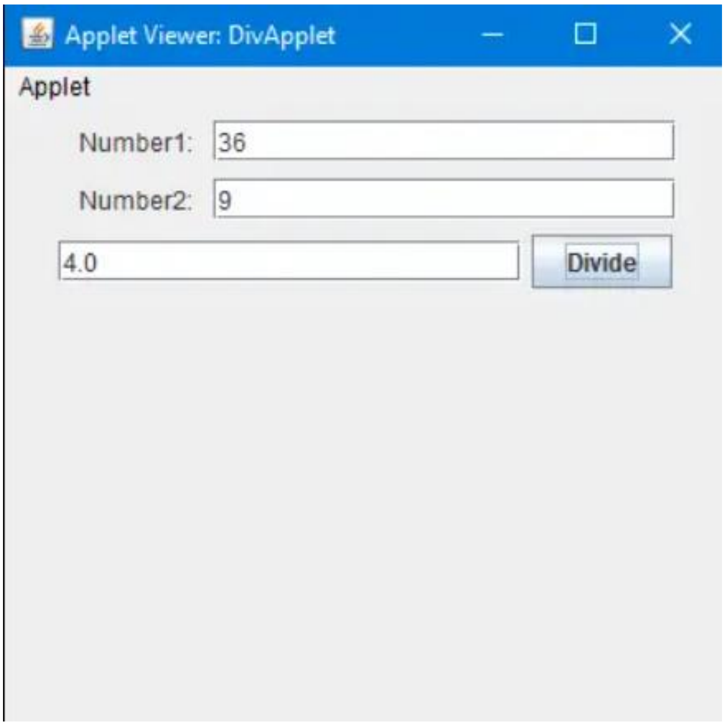
EXPERIMENT 6

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 Arithmetic Exception 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.invokeAndWait(

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");
 }
  }
}
```

SAMPLE OUTPUT;



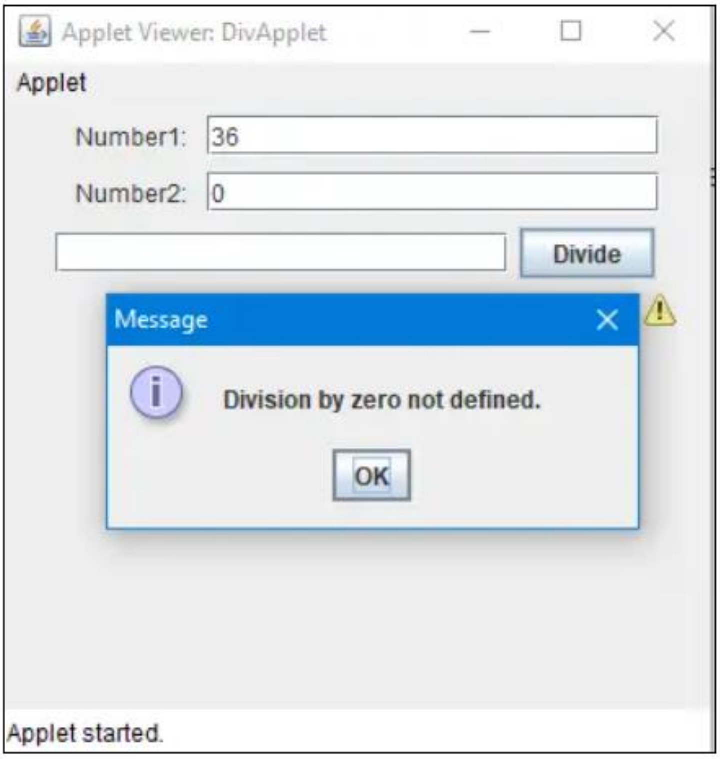
Applet Viewer: DivApplet

Applet

Number1: 36

Number2: 9

4.0



Applet Viewer: DivApplet

Applet

Number1: 36

Number2: 0

Message

Division by zero not defined.

Applet started.

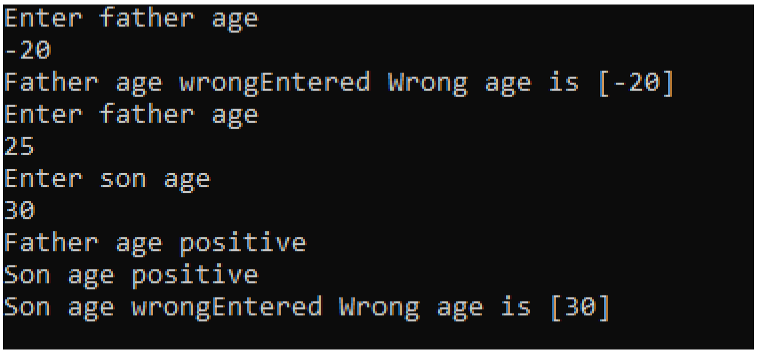
EXPERIMENT 7

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.

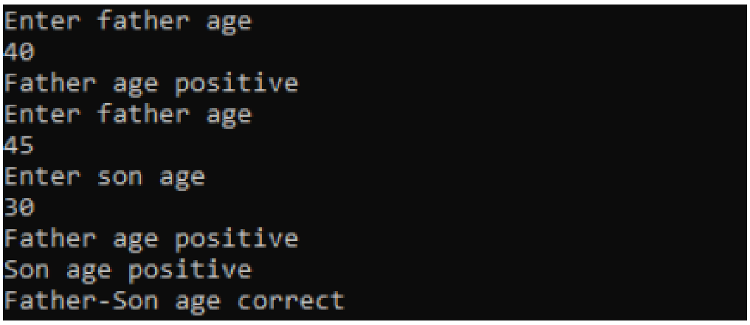
CODE:

```
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);
        }
    }
 }
```

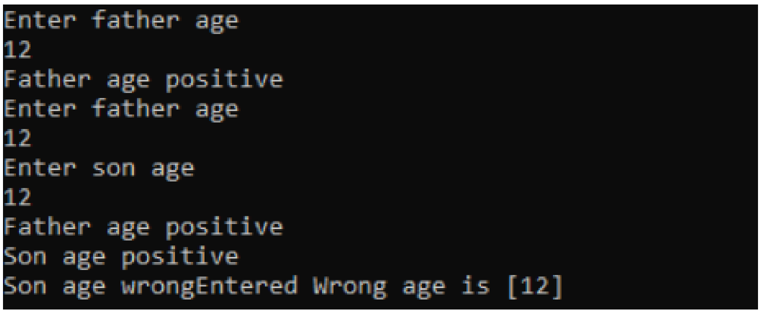
SAMPLE OUTPUT:



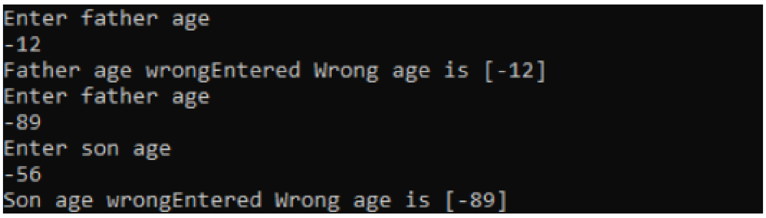
```
Enter father age
-20
Father age wrongEntered Wrong age is [-20]
Enter father age
25
Enter son age
30
Father age positive
Son age positive
Son age wrongEntered Wrong age is [30]
```



```
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]
```

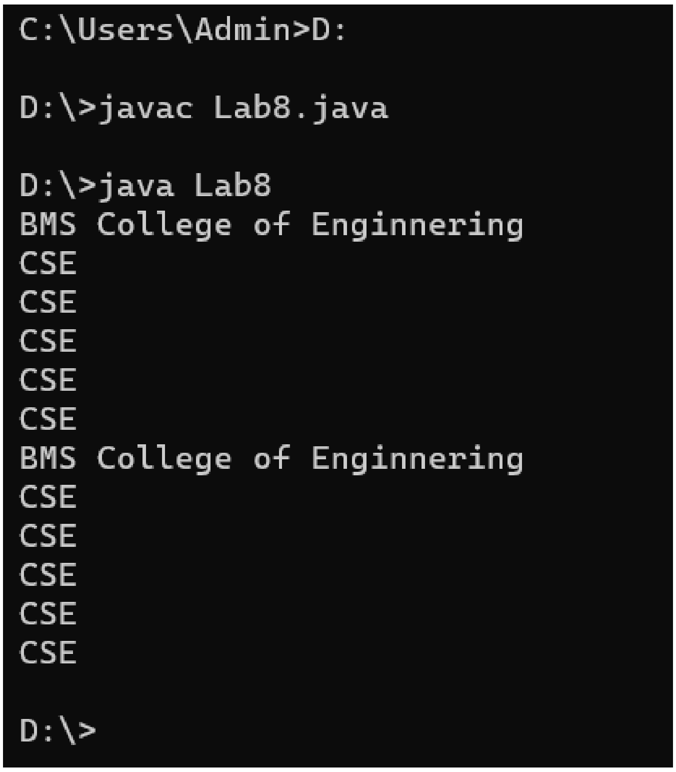
EXPERIMENT 8

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.

CODE:

```
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
BMS College of Enginnering
CSE
CSE
CSE
CSE
CSE

D:\>
```