VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

NISHANT KUMAR (1BM21CS117)

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 "**OOJ Lab**" carried out by **NISHANT KUMAR(1BM21CS117),** who is a 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 Data structures Lab **- ()** work prescribed for the said degree.

Dr. Rajeshwari BS	Dr. Jyothi S Nayak
Assistant Professor	Professor and HOD
Department of CSE	Department of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

Index Sheet :

	the area of the given shape.	
4	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.	14-17
5	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	18-24

	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.	
6	Exception handling program	25-33
7	User defined exception father and son program	34-40
8	Multithreading program	40-44

Course Outcome :

CO1	Apply the concept of linear and nonlinear data structures.
CO2	Analyze data structure operations for a given problem.
CO3	Design and develop solutions using Data Structure concepts.
CO4	Conduct practical experiments for demonstrating the operations of different data structures.

LAB PROGRAM 1:

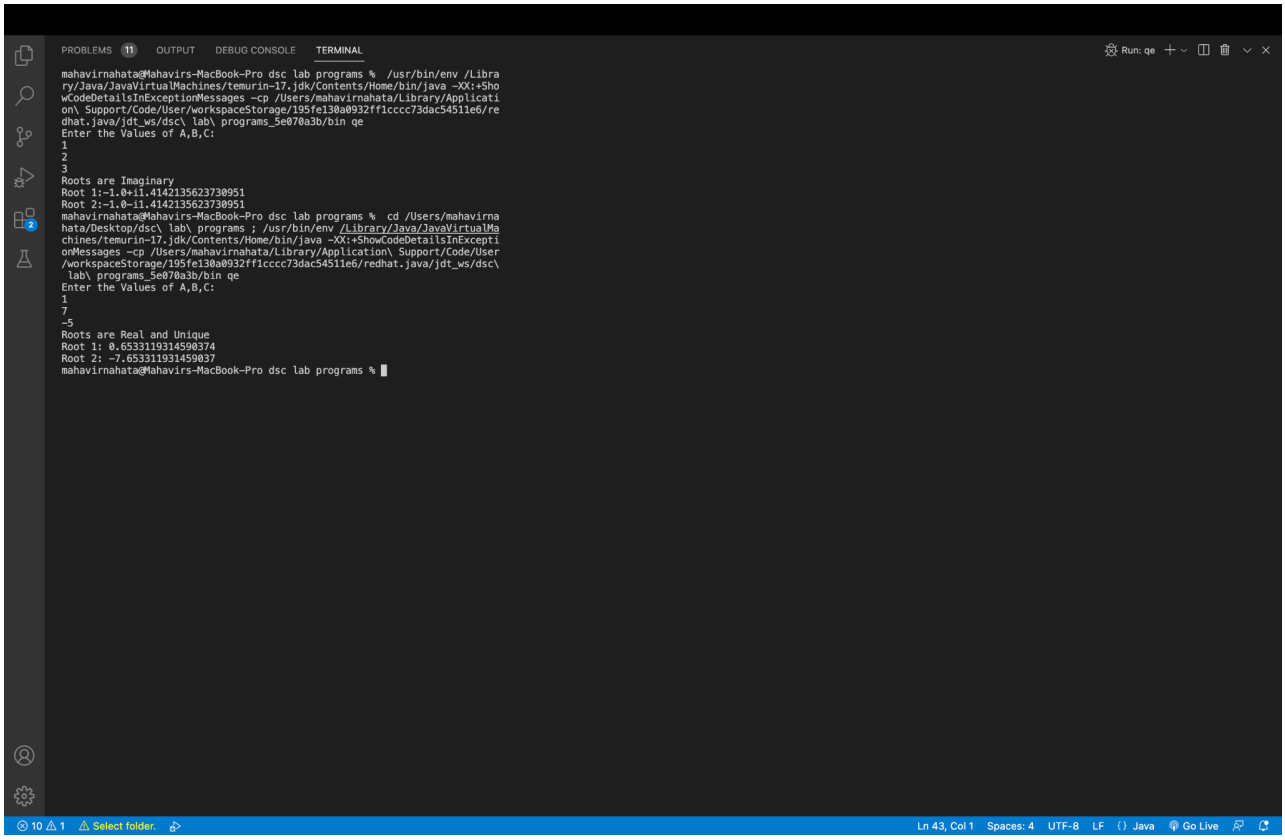
Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions:

Program code:

```
import java.util.Scanner;

class qe {

	public static void main(String args[])

	{

		Scanner sc =new Scanner(System.in);

		System.out.println("Enter the Values of A,B,C:");

		double a=sc.nextDouble();

		double b=sc.nextDouble();

		double c=sc.nextDouble();

		double d;

		d=(b*b)-(4*a*c);

		double r1,r2;

		if(a==0)

		{

			System.out.println("Invaild Input");

		}

		else if(d>0)

		{

			System.out.println("Roots are Real and Unique");

			r1=(-b+Math.sqrt(d))/(2*a);

			r2=(-b-Math.sqrt(d))/(2*a);

			System.out.println("Root 1: "+r1);

			System.out.println("Root 2: "+r2);

		}

		else if(d==0)

		{
```

```
                System.out.println("Roots are Real and Equal");
                r1=(-b/(2*a));
                r2=(-b/(2*a));
                System.out.println("Root 1:"+r1);
                System.out.println("Root 2:"+r2);
            }
            else
            {
                System.out.println("Roots are Imaginary");
                r1=(-b/(2*a));
                r2=(Math.sqrt(-d)/(2*a));
                System.out.println("Root 1:"+r1+"+i"+r2);
                System.out.println("Root 2:"+r1+"-i"+r2);
            }
        }
}
```

Output:



LAB PROGRAM 2:

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:

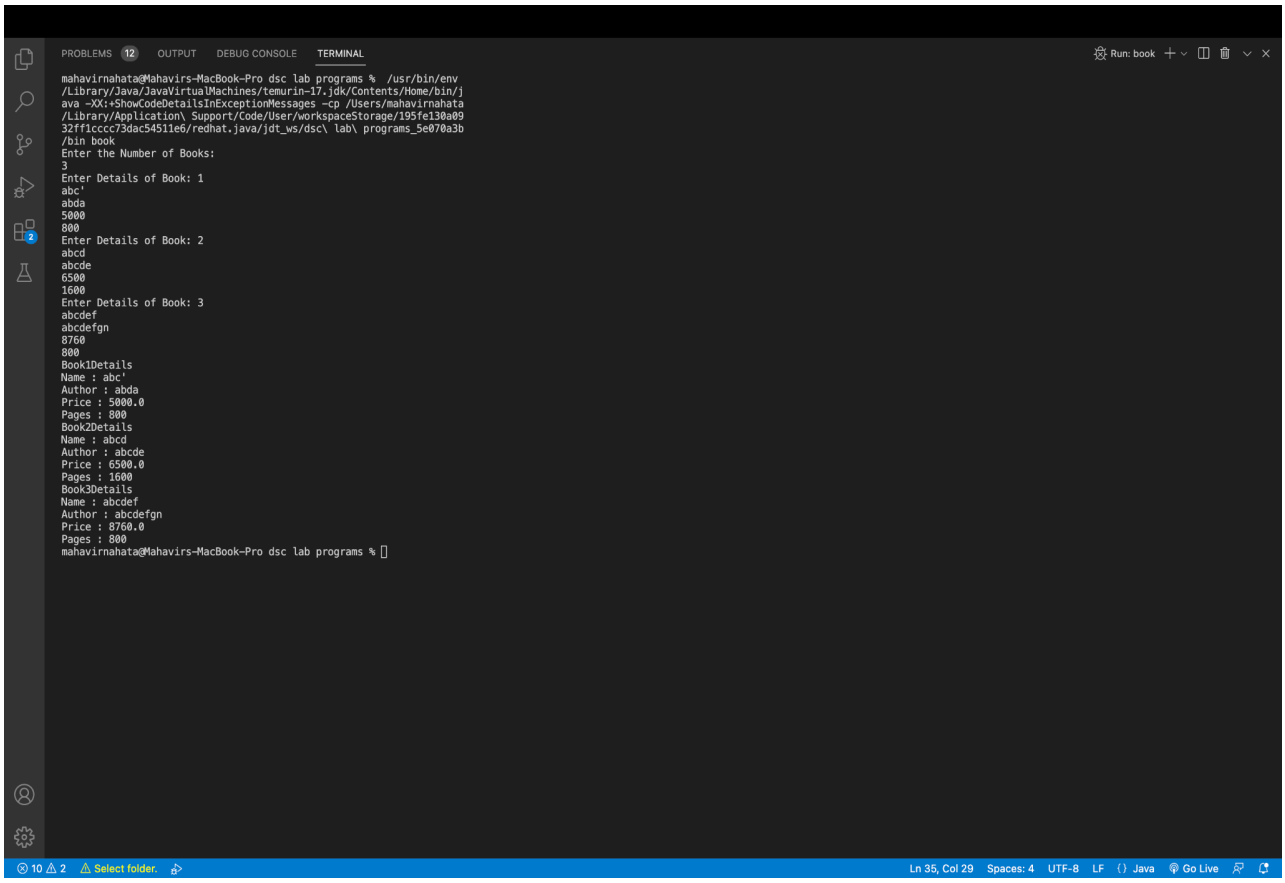
Program code:

```
import java.util.Scanner;

import java.lang.*;


class Book1 {

    String name;

    String author;
```

```
    float price;
    Integer pages;
    Book1(String n,String a,float p,Integer pa) {
        name=n;
        author=a;
        price=p;
        pages=pa;
    }
    void display() {
        System.out.println("Name : "+name);
        System.out.println("Author : "+author);
        String pricee=Float.toString(price);
        System.out.println("Price : "+pricee);
        System.out.println("Pages : "+pages.toString());
    }
}

class book {
    public static void main(String args[])
    {
        int n;
        String name;
        String author;
        float price;
        Integer pages;
```

```
        System.out.println("Enter the Number of Books: ");
        Scanner sc=new Scanner(System.in);
        n=sc.nextInt();
        Book1[] arr;
        arr = new Book1[n];
        for(int i=0;i<n;i++)
        {
           System.out.println("Enter Details of Book: "+(i+1));
           name=sc.next();
           author=sc.next();
           price=sc.nextFloat();
           pages=sc.nextInt();
           arr[i]=new Book1(name,author,price,pages);
        }
        for(int i=0;i<n;i++)
        {
           System.out.println("Book"+(i+1)+"Details");
           arr[i].display();
        }
    }
}
```


Outputs:



LAB PROGRAM 3:

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:

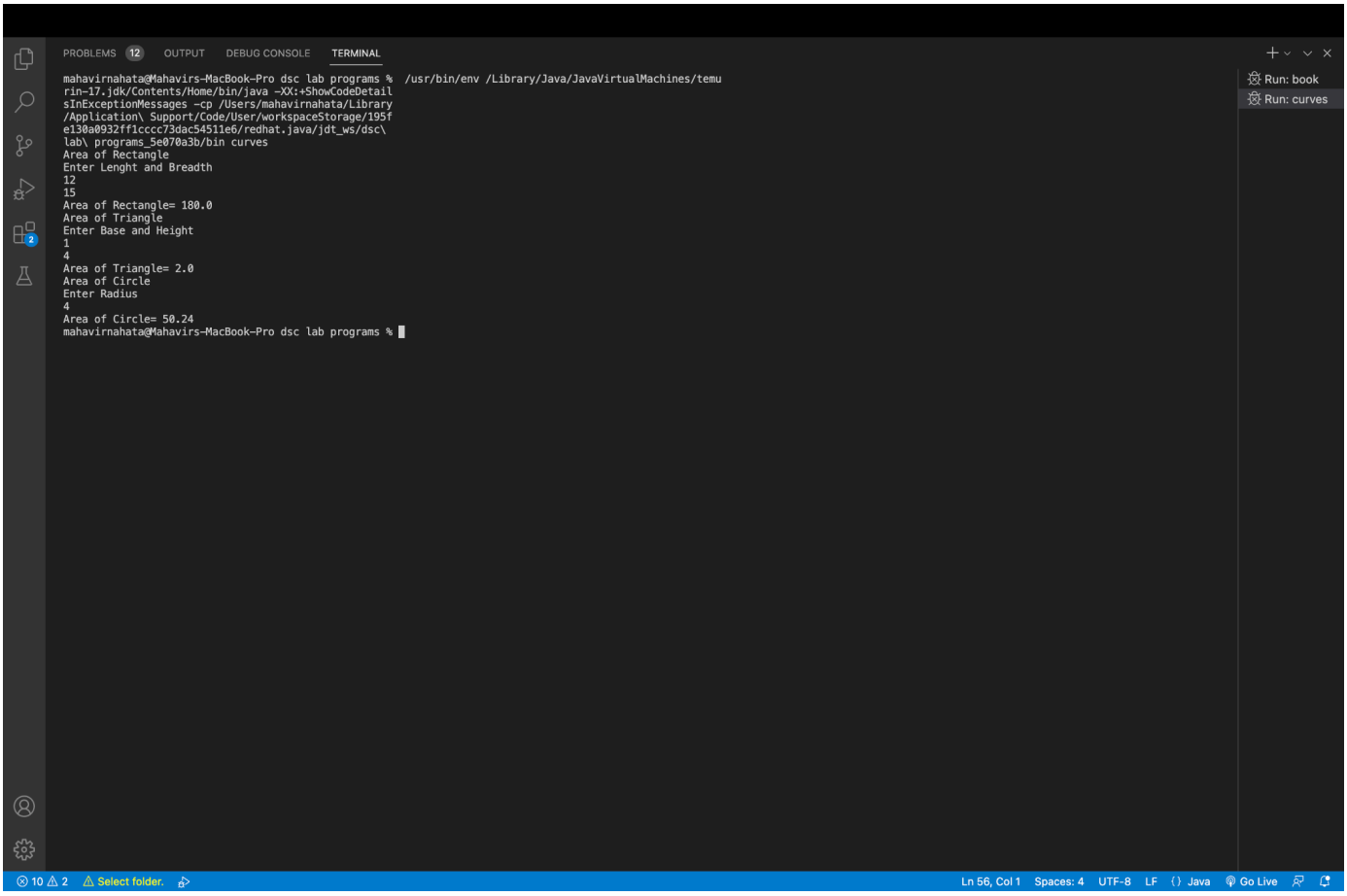
Program code:

```
import java.util.*;

abstract class Shape

{
```

```
    Scanner sc=new Scanner(System.in);
    abstract void printArea();
}
class Rectangle extends Shape
{
    void printArea()
    {
        int l,b;
        System.out.println("Area of Rectangle");
        System.out.println("Enter Lenght and Breadth");
        l=sc.nextInt();
        b=sc.nextInt();
        double area=l*b;
        System.out.println("Area of Rectangle="+" "+area);
    }
}
class Triangle extends Shape
{
    void printArea()
    {
        int b,h;
        System.out.println("Area of Triangle");
        System.out.println("Enter Base and Height");
        b=sc.nextInt();
        h=sc.nextInt();
        double area=(b*h)/2;
        System.out.println("Area of Triangle="+" "+area);
    }
}
class Circle extends Shape
```

```
{
   void printArea()
   {
      int r;
      System.out.println("Area of Circle");
      System.out.println("Enter Radius");
      r=sc.nextInt();
      double area=3.14*r*r;
      System.out.println("Area of Circle="+" "+area);
   }
}
class curves {
   public static void main(String ars[])
   {
      Rectangle rec=new Rectangle();
      rec.printArea();
      Triangle tri=new Triangle();
      tri.printArea();
      Circle cir=new Circle();
      cir.printArea();
   }
}
```

Outputs:



LAB PROGRAM 4:

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:

Program code:

```
import java.util.Scanner;


class Account {
   String customerName;
   String accType;
   int accNum;

   double balance = 1000;

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

   void displayBalance() {
      System.out.println("The balance in the account is : " + balance);
   }
}
```

```
class SavAcct extends Account {
   int n = 4;
   double r = 0.07;

   void interest(double y) {
      double x = balance;
      balance = balance * Math.pow((1 + r / n), (y * n));
      System.out.println("An amount of " + (balance - x) + " has been deposited as interest");
   }

   void withdrawal(int amount) {
      if (balance >= amount) {
         balance -= amount;
      } else {
         System.out.println("You dont have the sufficient balance");
      }
   }
}

class CurrAcct extends Account {
   int minBalance = 1000, penalty = 7;

   void withdrawal(int amount) {
      if (balance <= minBalance) {
         balance -= penalty;
         System.out.println("A penalty of 7 rupees has been imposed!");
      }
```

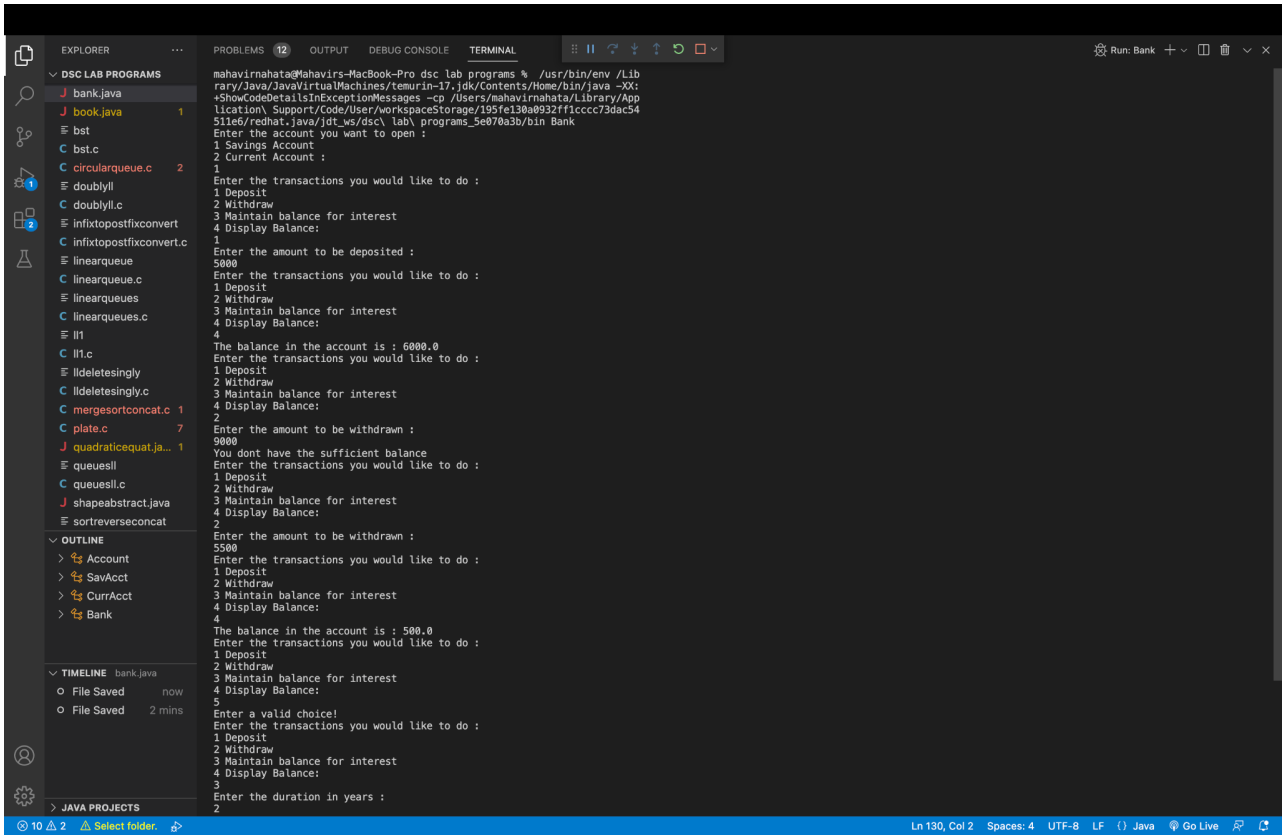
```
      if (balance >= amount) {
         balance -= amount;
      } else {
         System.out.println("You dont have the sufficient balance");
      }
   }
}

class Bank {
   public static void main(String args[]) {
      Scanner sc = new Scanner(System.in);
      System.out.println(
            "Enter the account you want to open :\n1 Savings Account\n2 Current Account : ");
      int choice = sc.nextInt();
      if (choice == 1) {
         SavAcct act = new SavAcct();
         while (true) {
            System.out.println(
                  "Enter the transactions you would like to do :\n1 Deposit\n2 Withdraw\n3 Maintain
balance for interest\n4 Display Balance: ");
            choice = sc.nextInt();
            switch (choice) {
               case 1:
                  System.out.println("Enter the amount to be deposited : ");
                  int amount = sc.nextInt();
                  act.deposit(amount);
                  break;
               case 2:
```

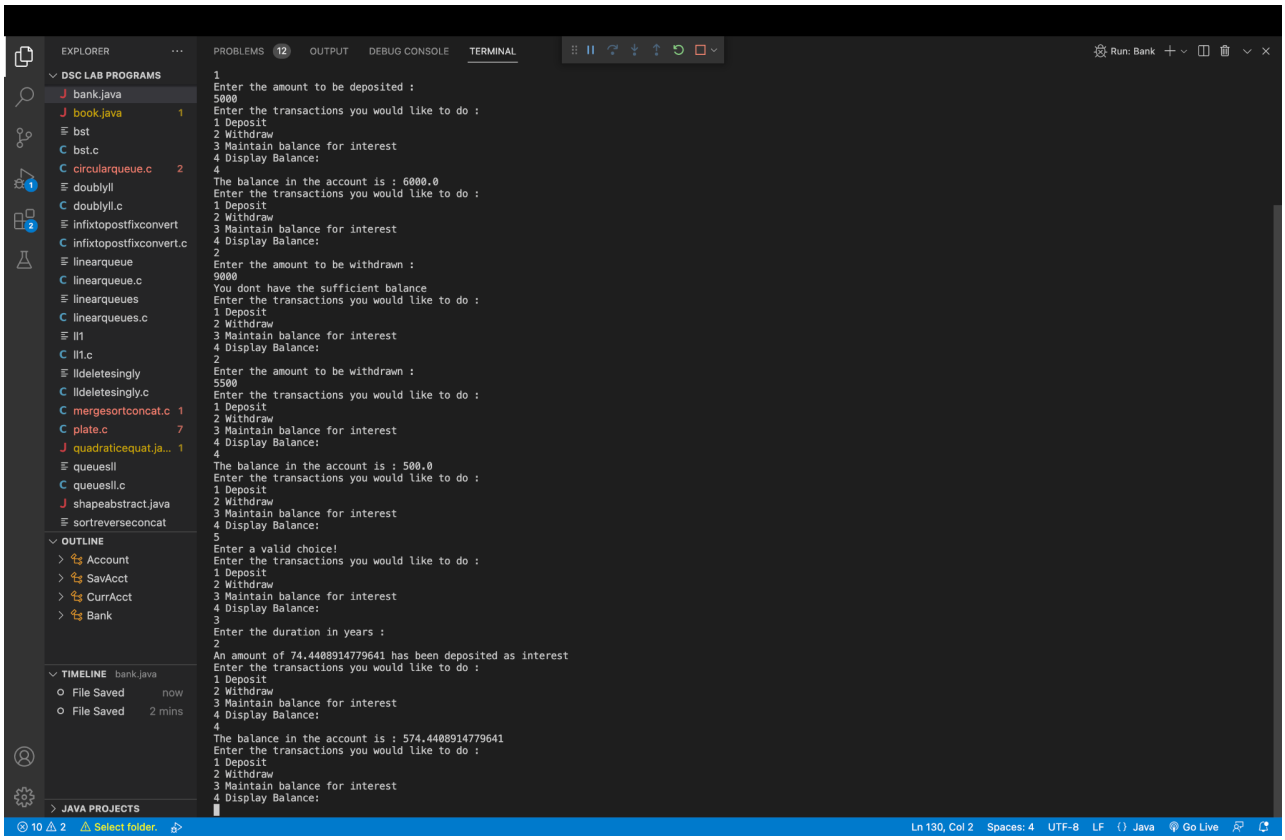
```
                System.out.println("Enter the amount to be withdrawn : ");
                int amt = sc.nextInt();
                act.withdrawal(amt);
                break;
            case 3:
                System.out.println("Enter the duration in years : ");
                double y = sc.nextDouble();
                act.interest(y);
                break;
            case 4:
                act.displayBalance();
                break;
            default:
                System.out.println("Enter a valid choice!");
                break;
          }
        }
      }

      else if (choice == 2) {
        CurrAcct act = new CurrAcct();
        while (true) {
          System.out.println(
              "Enter the transactions you would like to do :\n1 Deposit via chequebook\n2
Withdraw via chequebook\n3 Display Balance: ");
          choice = sc.nextInt();
          switch (choice) {
            case 1:
```

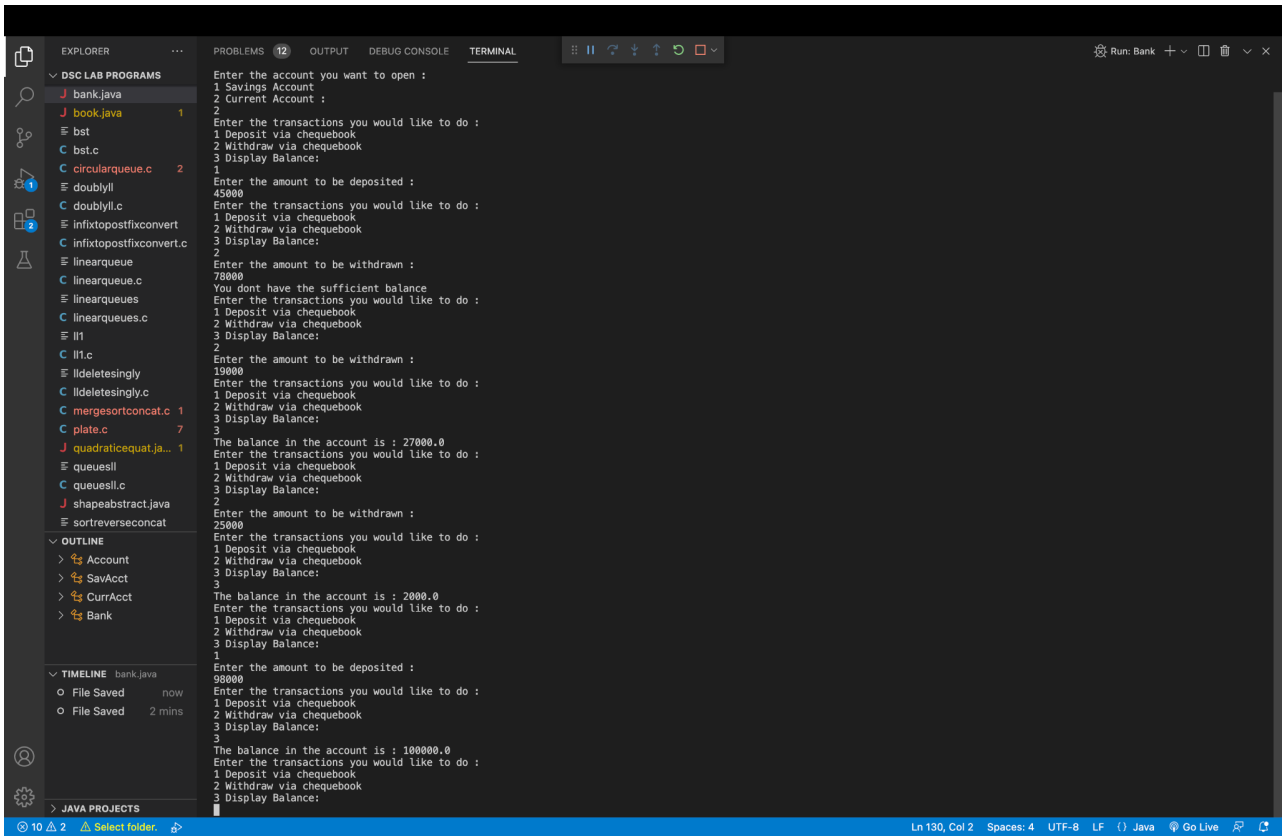
```
            System.out.println("Enter the amount to be deposited : ");
            int amount = sc.nextInt();
            if (amount < act.minBalance) {
               act.balance -= act.penalty;
               System.out.println("A penalty of 7 rupees has been imposed!");
            }
            act.deposit(amount);
            break;
         case 2:
            System.out.println("Enter the amount to be withdrawn : ");
            int amt = sc.nextInt();
            act.withdrawal(amt);
            break;
         case 3:
            act.displayBalance();
            break;
         default:
            System.out.println("Enter a valid choice!");
            break;
      }
   }
}

else {
   System.out.println("Ënter a valid choice!");
   System.exit(0);
}
sc.close();
```

```
    }
}
```


Outputs:


EXPLORER
DSC LAB PROGRAMS
bank.java
book.java
bst
bst.c
circularqueue.c
doublyll
doublyll.c
infixtopostfixconvert
infixtopostfixconvert.c
linearqueue
linearqueue.c
linearqueues
linearqueues.c
ll1
ll1.c
lldeletesingly
lldeletesingly.c
mergesortconcat.c
plate.c
quadraticequat.ja...
queuesll
queuesll.c
shapeabstract.java
sortreverseconcat
OUTLINE
Account
SavAcct
CurrAcct
Bank
TIMELINE bank.java
File Saved now
File Saved 2 mins
JAVA PROJECTS
PROBLEMS 12
OUTPUT
DEBUG CONSOLE
TERMINAL
Run: Bank
mahavirnahata@Mahavirs-MacBook-Pro dsc lab programs % /usr/bin/env /Library/Java/JavaVirtualMachines/temurin-17.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/mahavirnahata/Library/Application\ Support/Code/User/workspaceStorage/195fe130a0932ff1cccc73dac54511e6/redhat.java/jdt_ws/dsc\ lab\ programs_5e070a3b/bin Bank
Enter the account you want to open :
1 Savings Account
2 Current Account :
1
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
1
Enter the amount to be deposited :
5000
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
4
The balance in the account is : 6000.0
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
2
Enter the amount to be withdrawn :
9000
You dont have the sufficient balance
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
2
Enter the amount to be withdrawn :
5500
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
4
The balance in the account is : 500.0
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
5
Enter a valid choice!
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
3
Enter the duration in years :
2
⊗ 10 ⚠ 2 ⚠ Select folder.
Ln 130, Col 2 Spaces: 4 UTF-8 LF { } Java Go Live


EXPLORER
DSC LAB PROGRAMS
bank.java
book.java 1
bst
bst.c
circularqueue.c 2
doublyll
doublyll.c
infixtopostfixconvert
infixtopostfixconvert.c
linearqueue
linearqueue.c
linearqueues
linearqueues.c
ll1
ll1.c
lldeletesingly
lldeletesingly.c
mergesortconcat.c 1
plate.c 7
quadraticequat.ja... 1
queuesll
queuesll.c
shapeabstract.java
sortreverseconcat
OUTLINE
Account
SavAcct
CurrAcct
Bank
TIMELINE bank.java
File Saved now
File Saved 2 mins
JAVA PROJECTS
PROBLEMS 12
OUTPUT
DEBUG CONSOLE
TERMINAL
Run: Bank
1
Enter the amount to be deposited :
5000
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
4
The balance in the account is : 6000.0
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
2
Enter the amount to be withdrawn :
9000
You dont have the sufficient balance
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
2
Enter the amount to be withdrawn :
5500
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
4
The balance in the account is : 500.0
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
5
Enter a valid choice!
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
3
Enter the duration in years :
2
An amount of 74.4408914779641 has been deposited as interest
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
4
The balance in the account is : 574.4408914779641
Enter the transactions you would like to do :
1 Deposit
2 Withdraw
3 Maintain balance for interest
4 Display Balance:
Ln 130, Col 2
Spaces: 4
UTF-8
LF
Java
Go Live


EXPLORER
DSC LAB PROGRAMS
bank.java
book.java 1
bst
bst.c
circularqueue.c 2
doublyll
doublyll.c
infixtopostfixconvert
infixtopostfixconvert.c
linearqueue
linearqueue.c
linearqueues
linearqueues.c
ll1
ll1.c
lldeletesingly
lldeletesingly.c
mergesortconcat.c 1
plate.c 7
quadraticequat.ja... 1
queuesll
queuesll.c
shapeabstract.java
sortreverseconcat
OUTLINE
Account
SavAcct
CurrAcct
Bank
TIMELINE bank.java
File Saved now
File Saved 2 mins
JAVA PROJECTS
PROBLEMS 12
OUTPUT
DEBUG CONSOLE
TERMINAL
Run: Bank
Enter the account you want to open :
1 Savings Account
2 Current Account :
2
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
1
Enter the amount to be deposited :
45000
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
2
Enter the amount to be withdrawn :
78000
You dont have the sufficient balance
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
2
Enter the amount to be withdrawn :
19000
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
3
The balance in the account is : 27000.0
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
2
Enter the amount to be withdrawn :
25000
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
3
The balance in the account is : 2000.0
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
1
Enter the amount to be deposited :
98000
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
3
The balance in the account is : 100000.0
Enter the transactions you would like to do :
1 Deposit via chequebook
2 Withdraw via chequebook
3 Display Balance:
Ln 130, Col 2
Spaces: 4
UTF-8
LF
Java
Go Live

LAB PROGRAM 5:

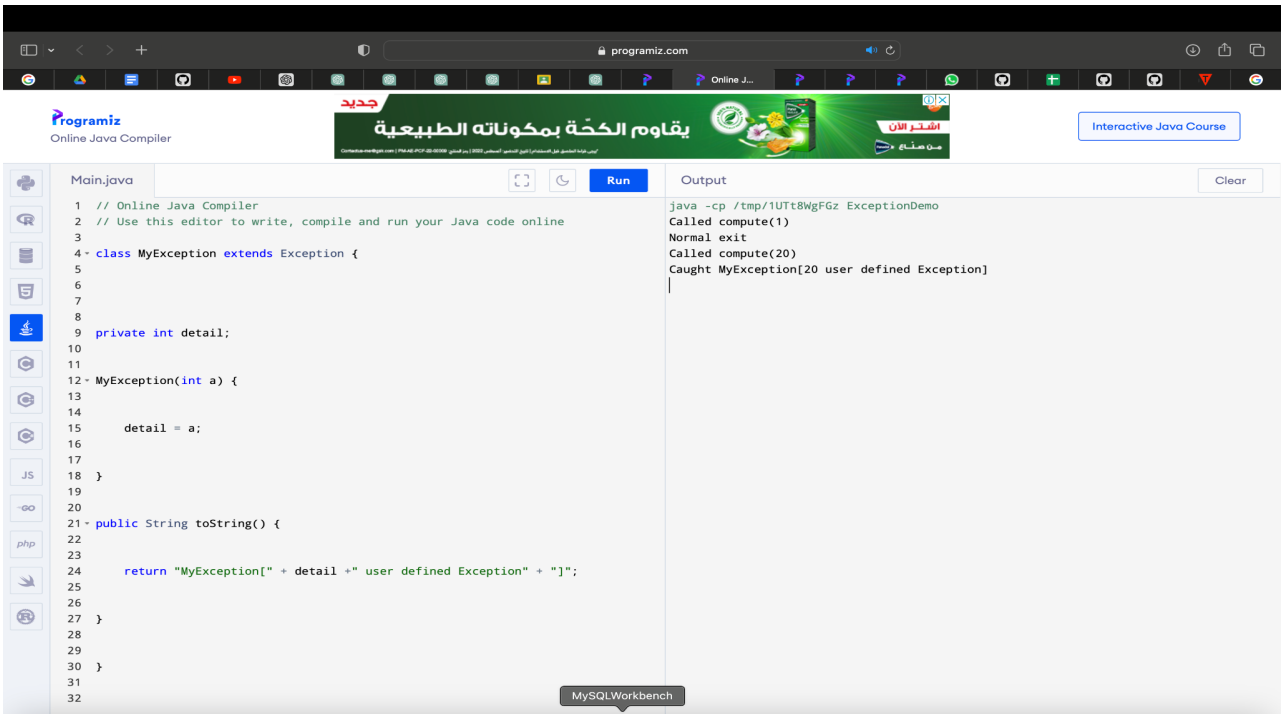
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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

Program code:

```
class MyException
extends Exception {

                    private int detail;

                    MyException(int a) {

                         detail = a;

                    }

                    public String toString() {

                         return "MyException[" + detail +"
                    user defined Exception" + "]";

                    }
```

```
}

class ExceptionDemo {

static void compute(int a) throws
MyException

{

    System.out.println("Called
compute(" + a + ") ");

    if(a > 10)

    throw new MyException(a);

    System.out.println("Normal exit");

}

public static void main(String args[])

{

    try {

    compute(1);
```

```
        compute(20);

        }

    catch (MyException e)

    {

    System.out.println("Caught "+e);

    }

    }

    }
```



LAB PROGRAM 6:

```
import
java.uti
l.Scanne
r;
```

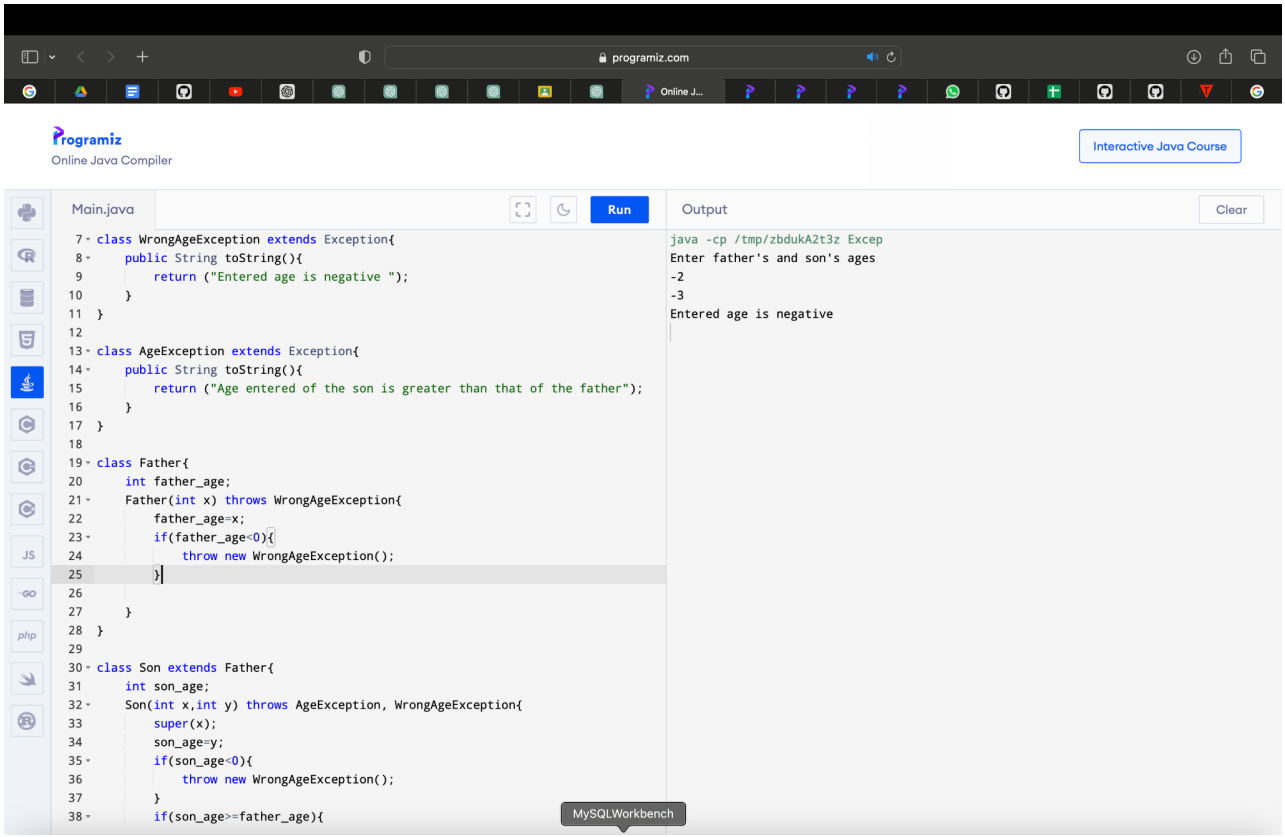
```
class IllogicalAgeException extends Exception {

private String message;

IllogicalAgeException(String error_text) {
```

```
        this.message = error_text;
    }


    @Override
    public String toString() {
        return "IllogicalAgeException: " +
this.message;
    }
}


class Father {
    private int age;
    Father(int age) throws IllogicalAgeException
{
        if(age < 0)
            throw new IllogicalAgeException("Age
is negative");
        this.age = age;
    }
    int getAge() {return this.age;}
```

```
}



class Son extends Father{

   private int age;

   Son(int age_son, int age_father) throws
IllogicalAgeException {

       super(age_father);

       if(age_son < 0)

           throw new IllogicalAgeException("Age
is negative");

       if(age_son >= age_father)

           throw new IllogicalAgeException("Age
of son is greater than or equal to father");

       this.age = age_son;

   }

   int getAge() {return this.age;}

   int getParentAge() {return super.getAge();}

}



class Main {
```

```
   public static void main(String[] args)
throws IllogicalAgeException {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the age of son
and father:");

        Son son = new Son(sc.nextInt(),
sc.nextInt());



        System.out.println("Age of father is: "
+ son.getParentAge());

        System.out.println("Age of son is: " +
son.getAge());

        sc.close();

    }

}
```

Output::




```
7  class WrongAgeException extends Exception{
8      public String toString(){
9          return ("Entered age is negative ");
10     }
11 }
12
13 class AgeException extends Exception{
14     public String toString(){
15         return ("Age entered of the son is greater than that of the father");
16     }
17 }
18
19 class Father{
20     int father_age;
21     Father(int x) throws WrongAgeException{
22         father_age=x;
23         if(father_age<0){
24             throw new WrongAgeException();
25         }
26
27     }
28 }
29
30 class Son extends Father{
31     int son_age;
32     Son(int x,int y) throws AgeException, WrongAgeException{
33         super(x);
34         son_age=y;
35         if(son_age<0){
36             throw new WrongAgeException();
37         }
38         if(son_age>=father_age){
```

LAB PROGRAM 7:

```
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 xx[])
```

```
{

     new Call("BMS College of
Enginnering",10000,2);

     new Call("CSE",2000,10);

}

}
```
