

~~Observation~~

Program I

- Q 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 discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
```

```
class Quadratic
```

```
{
```

```
    int a,b,c;
```

```
    double r1,r2,d;
```

```
    void getd()
```

```
{
```

```
    Scanner s = new Scanner(System.in);
```

```
    System.out.println("Enter the coefficient of a,b,c");
```

```
    a = s.nextInt();
```

```
    b = s.nextInt();
```

```
    c = s.nextInt();
```

```
3
```

```
    void compute()
```

```
{
```

```
    while (a==0)
```

```
{
```

```
        System.out.println("Not a quadratic equation");
```

```
        System.out.println("Enter a non zero value  
for a:");
```

```
    Scanner s = new Scanner(System.in);
```

```
    a = s.nextInt();
```

```
3
```

$d = b^2 - 4ac;$
 if ($d == 0$)
 {

$$r1 = (-b) / (2a);$$

System.out.println ("Roots are real and equal");
 System.out.println ("Root1 = Root2 = " + r1);

3

else if ($d > 0$)

{

$$r1 = ((-b) + (\text{Math.sqrt}(d))) / (2a);$$

$$r2 = ((-b) - (\text{Math.sqrt}(d))) / (2a);$$

System.out.println ("Roots are real and distinct");

System.out.println ("Root1 = " + r1 + "Root2 = " + r2);

3

else if ($d < 0$)

{

System.out.println ("Roots are imaginary");

$$r1 = (-b) / (2a);$$

$$r2 = \text{Math.sqrt}(-d) / (2a);$$

System.out.println ("Root1 = " + r1 + " + " + r2 + "i");

System.out.println ("Root1 = " + r1 + " - " + r2 + "i");

3

3

3

Class QuadraticMain

{

public static void main (String args[])

{

Quadratic q = newQuadratic();

q.getd();

q.compute();

3
})

Output

Enter the coefficients a,b,c

5 10 15

Roots are imaginary

$$\text{Root 1} = -1.0 + i1.11h$$

$$\text{Root 2} = -1.0 - i1.11h$$

- Q 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 Subject {
```

```
    int subjectMarks;
```

```
    int credits;
```

```
    int grade;
```

```
}
```

```
class StudentClass {
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Scanner s;
```

```
    Subject[] subject;
```

```
StudentClass() {
```

```
Subject = new Subject[9];  
for (int i=0; i<9; i++)  
    Subject[i] = new Subject();  
S = new Scanner(System.in);  
3
```

```
Void get Student Details () {  
    System.out.print ("Enter student name");  
    name = S.nextLine();  
    System.out.print ("Enter student usn");  
    usn = S.nextLine();  
    33 (this is part of the program)
```

```
Void get Marks () {  
    for (int i=0; i<9; i++) {  
        System.out.println ("Enter details for  
        Subject" + (i+1));  
        System.out.print ("Enter marks");  
        Subject[i].subjectMarks = S.nextInt();  
        System.out.print ("Enter credits");  
        Subject[i].credits = S.nextInt();  
    }  
}
```

```
if (Subject[i].subjectMarks >= 90) {  
    Subject[i].grade = 10;  
} else if (Subject[i].subjectMarks >= 80) {  
    Subject[i].grade = 9;  
} else if (Subject[i].subjectMarks >= 70) {  
    Subject[i].grade = 8;  
} else if (Subject[i].subjectMarks >= 60) {  
    Subject[i].grade = 7;  
} else if (Subject[i].subjectMarks >= 50) {  
    Subject[i].grade = 6;
```


Ex-6-1

```
s1.getStudentDetails();
```

```
s1.getMarks();
```

```
s1.computeSGPA();
```

```
s1.displayResult();
```

Output

```
Enter Student name: Pranav
```

```
Enter Student USN: IBM22ICOSO
```

```
Enter details for subject: 1
```

```
Enter marks: 12
```

(Similarly for 9 subjects)

~~Student Details:~~

Name: Pranav

USN: IBM22ICOSO

SGPA: 7.266

✓

16/11/24

Lab - 3

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

```
import java.util.Scanner;  
class Book {  
    String name;  
    String author;  
    int price;  
    int numPages;  
  
    public Book(String name, String author, int price, int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
  
    }  
  
    public String toString() {  
        String nameStr = "Book name:" + this.name + "\n";  
        String authorStr = "Author name:" + this.author + "\n";  
        String priceStr = "Price:" + this.price + "\n";  
        String numPagesStr = "Number of pages:" + this.numPages + "\n";  
  
        return nameStr + authorStr + priceStr + numPagesStr;  
    }  
  
    public class Main {
```

four
Include
members.
ils of
od that
. Develop
bjects.

```
public static void main ( String args[] ) {  
    System.out.println ("Enter number of books in record")
```

```
Scanner s = new
```

```
Scanner (System.in);
```

```
int n;
```

```
n = s.nextInt()
```

```
Book b[];
```

```
b = new Book[n];
```

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

```
{
```

```
    System.out.println ("Enter details for book" + (i+1) + ":" );
```

```
    System.out.print ("Name:");
```

```
    String name = s.next();
```

```
    System.out.print ("Author:");
```

```
    String author = s.next();
```

```
    System.out.print ("Price:");
```

```
    int price = s.nextInt();
```

```
    System.out.print ("Number of pages:");
```

```
    int numPages = s.nextInt();
```

```
b[i] = new Book(name, author, price, numPages);
```

```
}
```

```
System.out.println ("In Details of the Books:");
```

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

```
{
```

```
    System.out.println (b[i].toString());
```

```
}
```

```
s.close();
```

```
}
```

Lab - 4

Q 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.

abstract class Shape {

 int dimension1;

 int dimension2;

 public Shape(int dimension1, int dimension2) {

 this.dimension1 = dimension1;

 this.dimension2 = dimension2;

}

 public abstract void printArea();

}

class Rectangle extends Shape {

 public Rectangle(int length, int width) {

 super(length, width);

}

 public void printArea() {

 int area = dimension1 * dimension2;

 System.out.println("Area of rectangle" + area);

 }

 3

class Triangle extends Shape {

 public Triangle(int base, int height) {

 super(base, height);

 }

public void printArea() {
 double area = 0.5 * dimensions1 * dimensions2;
 System.out.println("Area of triangle " + area);
}

3
3

class Circle extends Shape {
 public Circle (int radius) {
 super (radius, 0);
 }
 public void printArea () {
 double area = Math.PI * dimension1 * dimension1;
 System.out.println ("Area of circle :" + area);
 }
}

3
3

public class Main {
 public static void main (String [] args) {
 Rectangle rectangle = new Rectangle (5, 8);
 Triangle triangle = new Triangle (4, 6);
 Circle circle = new Circle (3);
 }
}

rectangle.printArea();

triangle.printArea();

circle.printArea();

3
3

Output :-

Area of Rectangle : 40

Area of triangle : 12.0

Area of Circle : 28.27433

Output of Lab-3

Enter number of books in record

2

Enter details for Book 1:

Name : Inception

Author : Dan

Price : 2000

Number of pages : 600

Enter details for Book 2 :

Name : Carnigan

Author : Steven

Price : 1050

Number of pages : 400

Details of the Books :

Book name : Inception

Author : Dan

Price : 2000

Number of pages : 600

Book name : Carnigan

Author : Steven

Price : 1050

Number of pages : 400

✓ 23/12/20

Lab - 5

- d) Create a class Account that stores customer-name, account number and type of account. From this derive the classes Cur-account and Sav-account to make them more specific to their requirements. Include the necessary methods in order to
- 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

```
import java.util.Scanner;
```

```
class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
```

```
Account (String name, int accNum, String acctype,
        double initialBalance)
```

```
customerName = name;
```

```
accountNumber = accNum;
```

```
accountType = acctype;
```

```
balance = initialBalance;
```

```
void deposit (double amount) {
```

```
    balance += amount;
```

```
    System.out.println ("Balance: $" + balance);
```

```
3
```

```
void withdraw(double amount) {  
    if (amount > balance) {  
        System.out.println("Insufficient funds");  
    } else {  
        balance -= amount;  
        System.out.println("Withdrawal of $" + amount + " successful");  
    }  
}
```

```
void computeInterest() {  
}
```

```
void showDetails() {  
}
```

```
System.out.println("Customer Name:" + customerName);  
System.out.println("Account Number:" + accountNumber);  
System.out.println("Account Type:" + accountType);  
displayBalance();  
}
```

```
System.out.println("Interest Rate:" + interestRate);  
SavAcct(String name, int accNum, String  
accType, double initialBalance, double rate);  
super(name, accNum, accType, initialBalance);  
interestRate = rate;  
}
```

```
void computeInterest() {  
    double interest = balance * interestRate / 100;  
    deposit(interest);  
    System.out.println("Interest computed and deposited");  
    System.out.print(interest);  
}
```

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

```
SavAcct Savings = new SavAcct ("John Doe", 123456,
    "Savings", 1000, 3);
```

```
int choice;
do {
    System.out.println ("1. Deposit");
    System.out.println ("2. Withdraw");
    System.out.println ("3. Compute Interest");
    System.out.println ("4. Show Details");
    System.out.println ("5. Exit");
    System.out.print ("Enter your choice:");
    Choice = Scanner . nextInt ();
}
```

~~Case 1 : Enter amount to deposit~~

```
System.out.print ("Enter amount to deposit:");
double depositAmount = scanner.nextDouble();
```

~~Case 2 : Enter amount to withdraw~~

```
System.out.print ("Enter amount to withdraw");
double withdrawAmount = scanner.nextDouble();
Savings . withdraw (withdrawAmount);
break;
```

~~Case 3 : Enter amount to withdraw~~

```
Savings . computeInterest ();
break;
```

~~Case 4 :~~

```
Savings . showDetails ();
break;
```

case 5 :

```
System.out.println ("Exiting . . .");  
break;
```

default :

```
System.out.println ("Invalid choice!");
```

3

```
3 while (choice != 3);
```

```
Scanner.close();
```

3

3

Output

1. Deposit

2. Withdraw

3. Compute Interest

4. Show Details

5. Exit

Enter your choice : 1

Enter amount \$ 1000 to deposit : \$ 1000

Deposit of \$ 1000 successful

~~Deposit~~ Enter your choice : 3

Deposit of \$ 100.0 successful

Interest computed and deposited : \$ 100

Enter your choice : 4

Customer Name : John Doe

Account number : 123456

Account Type : Savings

Balance : \$ 2100.0

Strings - Lab

1. Java
 ~
Java
ABCDE
CDF
2. 3
abcabc
3. Dimensions are 10.0 by 14.0 by 12.0.
Box b: Dimensions are 10.0 by 14.0 by 12.0
4. Bmsc
5. 72
H
6. Bmsee equals Bmsee \rightarrow true
Bmsee equals College \rightarrow false
Bmsee equals BMSCe \rightarrow false
Bmsee equals BMSCe \rightarrow true
7. ✓ Substring is matched
- 8,9 It is true
It is true
10. Hello equals Hello \rightarrow true
Hello == Hello \rightarrow false
11. Watch
apple

ball

cat

dog

ent

free

gun

hen

ice

ing

rite

lift

man

net

orange

parrot

green

ring

star

tree

umbrella

van

xmas

yatch

zee.

12.

1

2

3

4

5

6

7

8

9

10

13. This was a test. Thwas was, too.
 This was a test. Thwas was, too.
 This is a test. Thwas was, too.
 This is a test. This was, too.
 This is a test. This is, too.

14. Hello World

15. Original String : I go to college
 Replaced String : I go to commege

16. Hello World.

17. Enter details for Student 1 :

Registration Number : 50

Full Name : Pranav

Semester : 3

CORPA : 9.8

Similarly we enter details for 3 students.

18. buffer before = Hello

char At(1) before = e

buffer after = H i l k e w o r l d

char At(1) after = i

H.

e

L

Hilke

Hilke h2

2h eRilh. (Contd 1) ~~using h2~~

3 G - aseem - epti /

19. Eagle : A bird having two wings

Eagle flies high in the sky.

Eagle screeches loudly.

20. Circle :

Area : 78.53

Perimeter : 31.45

Triangle :

Area : 6.0

Perimeter : 12.0

Hawk

Hawk soars gracefully through the air

Hawk emits a piercing cry.

Output

80. Popping integers from the stack

50

40

30

20

10

Popping doubles from the stack

5.5

4.5

3.5

2.5

1.5

Class Stack $<T>$

private T[] array;

private int top;

private final int maxSize = 5;

public Stack () {

array = (T[]) new Object[maxSize];

top = -1;

}

public void push (T item) {

if (top == maxSize - 1) {

System.out.println("Stack is full");

else {

array[++top] = item;

}

```

public T pop() {
    if (top == -1) {
        System.out.println("Stack is empty");
        return null;
    } else {
        return array[top];
    }
}

```

```
public class TestStack {

```

```

    public static void main (String [] args) {
        Stack < Integer > intStack = new Stack < Integer > ();
        intStack.push(10);
        intStack.push(20);
        intStack.push(30);
        intStack.push(40);
        intStack.push(50);
    }
}

```

~~System.out.println ("Popping integers from the stack");~~

~~for (i=0; i<26; i++)~~

~~System.out.println (intStack.pop());~~

~~3 (Custom 3) taking b6 2nd day~~

~~int x=10~~

~~Stack < Double > doubleStack = new Stack < Double > ();~~

~~for (i=0; i<6; i++)~~

~~E~~

~~" "+(x)+" "+x+" "+x)~~ System.out.println (doubleStack.push(x));
~~x=x+10;~~

~~3~~

~~for (i=0; i<6; i++)~~

~~E~~

~~System.out.println (doubleStack.pop());~~

~~}~~

Lab - b

Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks in all five courses.

(:/desktop/internals.java)

```
package CIE;
import java.util.Scanner;
```

```
public class Internals extends Student {
    protected int marks[] = new int[5];
```

```
    public void inputCIEmarks() {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.println("Enter CIE marks for 5 subjects");
        for (int i = 0; i < 5; i++) {
            scanner.nextInt();
        }
    }
```

```
i((x) do{System.out.println("Subject" + (i+1) + " marks"));
    marks[i] = scanner.nextInt();
}
```

3
3

((C:\Users\Acer\OneDrive\Desktop\Java\Lab 6\Internals.java))

C:/desktop/ Student.java.

```
package CIE;
import java.util.Scanner;

public class Student {
    protected String usn;
    public String name;
    protected int sem;

    public void inputStudentDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN");
        this.usn = scanner.nextLine();
        System.out.print("Enter Name");
        this.name = scanner.nextLine();
        System.out.print("Enter Semester");
        this.sem = scanner.nextInt();
    }
}
```

```
public void displayStudentDetails() {
    System.out.println("USN: " + this.usn);
    System.out.println("Name: " + this.name);
    System.out.println("Semester: " + this.sem);
}
```

C:/desktop/ externals.java

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
```

```
public class Externals extends Internals
```

```
protected int marks[10];
protected int finalMarks[];
```

public Externals() {

marks = new int[5];

finalMarks = new int[5];

}

public void inputSEEmarks()

{

Scanner scanner = new Scanner (System.in);

System.out.println ("Enter SEE marks for 5 subjects");

for (int i=0; i<5; i++)

System.out.print ("Subject" + (i+1) + ": ");

finalMarks[i] = scanner.nextInt();

Scanner scanner = new Scanner (System.in);

finalMarks[i] = scanner.nextInt();

Scanner scanner = new Scanner (System.in);

public void calculateFinalMarks()

for (int i=0; i<5; i++)

finalMarks[i] = marks[i]/2 + super_marks[i];

Scanner scanner = new Scanner (System.in);

Scanner scanner = new Scanner (System.in);

public void displayFinalMarks()

super.displayStudentDetails();

for (int i=0; i<5; i++)

System.out.println ("Subject" + (i+1) + ":" +

finalMarks[i]);

}

}

}

C:/desktop/ SEE Main.java

import SEE.Externals;

public class Main {

public static void main (String args []) {

int numofStudents = 2;

Externals finalMarks [] = new Externals
[numofStudents];

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

{

finalMarks [i] = new Externals();

finalMarks [i].inputStudentDetails();

System.out.println ("Enter CIE marks for"

+ finalMarks [i].name);

finalMarks [i].inputCIEmarks();

System.out.println ("Enter SEE marks for" +

finalMarks [i].name);

finalMarks [i].inputSEEmarks();

}

System.out.println ("Displaying data : \n");

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

{

finalMarks [i].calculateFinalMarks();

finalMarks [i].displayFinalMarks();

}

Output :-

Displaying data :

USN : 1bm22ic030

Name : pranav.

Semester : 3

Subject 1 : 5

Subject 2 : 12

Subject 3 : 14

Subject 4 : 18

Subject 5 : 14

VSN : 1bm22c051

Name : rahul

Semester : 3

Subject 1 : 16

Subject 2 : 18

Subject 3 : 20

Subject 4 : 22

Subject 5 : 24

(Semester 1) Subject 1 : 16

(Semester 1) Subject 2 : 18

Subject 3 : 20

Subject 4 : 22

Subject 5 : 24

(Semester 1) Subject 1 : 16

(Semester 1) Subject 2 : 18

Subject 3 : 20

Subject 4 : 22

Subject 5 : 24

(Semester 1) Subject 1 : 16

(Semester 1) Subject 2 : 18

(Semester 1) Subject 3 : 20

(Semester 1) Subject 4 : 22

(Semester 1) Subject 5 : 24

Lab -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 takes both father and son's age and throws an exception if son's age is \geq father's age.

```
import java.util.*
```

```
class FatherAgeException extends Exception
```

```
{
```

```
    public String toString() {
```

```
        return ("Father age is less than 0");
```

```
}
```

```
3 (sonage < age || age < 0) {
```

```
class SonAgeException extends Exception {
```

```
int a;
```

```
SonAgeException (int a) {
```

```
    a = age;
```

```
public String toString() {
```

```
    if (a < 0)
```

```
        return ("Son's age is less than 0");
```

```
else
```

```
        return ("Son's age is more than father's  
age");
```

```
3
```

```
3
```

class Father {

int age;

Scanner in = new Scanner (System.in);

Father () {

System.out.println ("Enters the father's age");

age = in.nextInt();

}

void ex1() throws fatherAgeException

{

if (age < 0)

throw new fatherAgeException();

}

}

class Son extends Father {

int age;

Son () {

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

age = in.nextInt();

}

void ex2 throws sonAgeException {

if (age < 10 || age > super.age) {

throw new sonAgeException (age);

}

}

public class except {

public static void main (String [] args) {

Son s = new Son();

try {

s.ex1();

catch (fatherAgeException e) {

System.out.println (e);

```
try {
```

```
    s.ex20();
```

```
}
```

```
catch (SonAgeException e) {
```

```
    System.out.println(e);
```

```
}
```

```
}
```

Output

```
Enter the father's age : 3
```

```
50 ("50") nothing to print
```

```
Enter the son's age : 3
```

```
35 ("35") printing the message
```

Lab - 3

- Q Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "(SE)" once every two seconds.

```
package javaLabPrograms;
class DnameThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("BMS college of engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println("interrupted");
            }
        }
    }
}
```

```
class DnameThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("BMS college of engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println("interrupted");
            }
        }
    }
}
```

```
class DnameThread extends Thread {
```

```
    public void run() {
```

```
        while (true) {
```

```
            System.out.println("(SE)");
```

```
            try {
```

```
                Thread.sleep(2000);
```

```
}
```

```
        catch (InterruptedException e) {
```

```
            System.out.println("interrupted");
```

```
}
```

```
}
```

```
public class threads {  
    public static void main (String [] args)  
    {  
        System.out.println ("Java");  
        CnameThread t1 = new CnameThread ();  
        DnameThread t2 = new DnameThread ();  
        t1.start ();  
        t2.start ();  
    }  
}
```

Output

java

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

"BMS college of Engineering")

CHENNAI

3 (a) (i) (ii) (iii) (iv) (v)

"BMS college of Engineering")

Lab 10

Q Demonstrate Inter process Communication and deadlock

Class Q E

```
private int n;  
private boolean valuset = false;  
synchronized int get() E  
while (valuset) E  
try E
```

```
System.out.println ("Consumer Waiting");  
wait();
```

3 catch (InterruptedException e) E

```
System.out.println ("Interrupted Exception caught")
```

3

```
System.out.println ("Got" + n);  
valueSet = false;
```

```
System.out.println ("Notifying producer")
```

```
notify();
```

```
return n;
```

3

Synchronized void put (int n) E

```
while (valueSet) E
```

```
try E
```

```
System.out.println ("In Producer waiting")
```

```
wait();
```

3

catch (InterruptedException e) E

```
System.out.println ("Interrupted Exception")
```

3

3

this.n=n

```
valueset = true;
```

```
System.out.println("Put : " + r);
System.out.println("Notifying consumer");
    notify();
}
}
```

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

~~Public class PCLFixed~~

~~public static void main (String args[])~~

```
Q q = new Q();
new Producer(q);
new consumer(q);
System.out.println("Press control to stop");
}
}
```

Output

Notifying consumer
producer waiting
Gwt: 0

Notifying producer

Put 1

Notifying consumer

producer waiting

consumed : 1

Put 2

Notifying consumer

Get 2

Notifying producer

Consumed : 2.

Deadlock

class A {

 synchronized void foo(B b) {

 String name = Thread.currentThread().getName();

 System.out.println(name + " entered A.foo");

 try {

 Thread.sleep(1000);

 }

 catch (Exception e) {

 System.out.println("A interrupted");

 System.out.println(name + " trying to call B.last()");

 b.last();

}

 synchronized void last() {

 System.out.println("Inside A.last");

 }

}

✓ 12/12/2021

class B {

 synchronized void bar(A a) {

 String name = Thread.currentThread().getName();

 System.out.println(name + " entered B.bar");

 try {

 Thread.sleep(1000);

 }

 catch (Exception e) {

 System.out.println("B Interrupted");

 }

 System.out.println(name + " trying to call A.last()");

 a.last();

 }

 synchronized void last() {

 System.out.println("Inside B.last");

 }

 }

class Deadlock implements Runnable {

 A a = new A();

 B b = new B();

};

Deadlock() {

 Thread.currentThread().setName("Main Thread");

 Thread t = new Thread(this, "RacingThread");

 t.start();

 .a.foo(b);

 System.out.println("Back in Main Thread");

};

 public void run() {

 b.bar(a);

 System.out.println("Back in main thread");

};

 public static void main(String args[]) {

 new Deadlock();

};

};

Output

MainThread entered A.foo

Racing Thread entered B.bar

MainThread trying to call B.last()

Inside A.last

Back in main thread

Racing Thread trying to call A.last()

Inside A.last

Back in other thread,

Lab-9

Q 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 an ~~Arithmatic~~ ^{NumberFormatException} Exception. Display the exception in a message dialog box, and if Num2 were zero, the program would throw an ~~Arithmatic~~ ^{ArithmeticException} Exception. Display.

```

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

class SwingDemo {
    JFrame jfrm = new JFrame("Divide App");
    jfrm.setSize(275, 180);
    jfrm.setLayout(new FlowLayout());
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JLabel jlab = new JLabel("Enter the divider and");
    jlab = new JLabel("divident");
    JTextField aJtf = new JTextField(8);
    JTextField bJtf = new JTextField(8);
    JButton button = new JButton("Calculate");
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslabel = new JLabel();
}

```

```

jfrm.add (err);
jfrm.add (ilab);
jfrm.add (ajtf);
jfrm.add (bjtf);
jfrm.add (button);
jfrm.add (alab);
jfrm.add (blab);
jfrm.add (anslab);

```

Action Listener I = new ActionListener () {

```
public void actionPerformed (ActionEvent evt)
```

```
System.out.println ("Action event from a text field");
```

}

;

```
ajtf.addActionListener (I);
bjtf.addActionListener (I);
```

;

button.addActionListener (new ActionListener ()) {

```
public void actionPerformed (ActionEvent evt) {
```

try {

```
int a = Integer.parseInt (ajtf.getText ());
int b = Integer.parseInt (bjtf.getText ());
```

int ans = a/b;

```
alab.setText ("In A = " + a);
```

```
blab.setText ("In B = " + b);
```

```
anslab.setText ("In Ans = " + ans);
```

;

catch (NumberFormatException e) {

```
aLab.setText ("");
```

```
blab.setText ("");
```

```
anslab.setText ("In Ans = " + ans);
```

err.setText ("Enter only Integers");

;

```

    catch (ArithmeticException e) {
        alab.setText(" ");
        blab.setText(" ");
        anslab.setText(" ");
        err.setText("B should be non-zero!");
    }
}

jfrm.setVisible(true);
}

public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}

}

```

Output

Divider App		- <input type="checkbox"/> X
Enter the divisor and dividend :		
500	6	
<input type="button" value="Calculate"/> A = 500 B = 6 Ans = 83		

~~Signature~~