

B.MS COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



OBJECT ORIENTED JAVA PROGRAMMING

LAB RECORD

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
Shiv sundar sah

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2023-2024

NAME :- SHIV SUNDAR SAH

SECTION :- 3E

USN :- IBM22CS258

Java Lab Records.

(003)

I N D E X

1BM22CS258

Shiv Sundar Sath

NAME: _____ STD.: _____ SEC.: 3E ROLL NO.: 5258 SUB.: 077(Lab)

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1	12/12	Java program to find quadratic eqn	1-4	(10) 28/12/23
2	19/12	To calculate SGPA.	5-10	(10) 28/12/23
3	26/12	To Create a class book	11-14	(10) 28/12/23
4	02/01	Abstract class	15-18	(10) 28/12/23
5	01/09	Develop the class book	19-25	(10) 28/11/23
	01/09	Software engineer	26	
6	01/03	Passage	26-30	(10) 28/12/23
7	01/30	Exception	31-	(10) 28/12/23
8	02/06	Create threads		(10) 28/12/23
10	03/06	open Ended Exercise		28/12/2024
9	02/20	Division main		28/20/2024

Lap program \rightarrow 1 (1B22CS258)

1. Develop a Java program that prints all real solutions to the quadratic $ax^2 + bx + c = 0$. Read a, b, c and use the quadratic formula. If the ~~distance~~ 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 f(int a, int b, int c)  
    {
```

```
        Scanner s = new Scanner(System.in);  
        System.out.print("Enter the coefficient  
        of a,b,c");
```

~~```
 a = s.nextInt();
```~~~~```
        b = s.nextInt();
```~~~~```
 c = s.nextInt();
```~~

```
 }
 void compute()
{
```

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

{

System.out.println("Not a quadratic eqn");  
 System.out.println("Enter a non zero  
 value for a:");

Scanners = new Scanner(System.in);

a = s.nextInt();

{

d = b \* b - 4 \* a \* c;

if (d == 0)

r1 = (-b) / (2 \* a);

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

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

{

else if (d > 0)

{

r1 = ((-b) + (math.Sqrt(d))) / (double)  
 (2 \* a);

r2 = ((-b) - (math.Sqrt(d))) / (double)(2 \* a);

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

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

{

else if ( $d < 0$ )

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

$r_1 = (-b) / (2 * a);$

$r_2 = \text{Math.sqrt}(-d) / (2 * a);$

System.out.print("Root1 = " + r1 + "\n" +  
 $r_2");$

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

}

}

Output:

class Quadratic main

{

    public static void main String args[]);

        Quadratic q = new Quadratic();

        q.getd()

        q.compute();

}

OUTPUT.

Enter the coefficient of a,b,c.

1 2 3

Roots are Imaginary

$$\text{Root 1} = -1.0 + i 1.41421356$$

$$\text{Root 2} = -1.0 - i 1.41421356$$

Enter the coefficient a,b,c

1 3 1

Roots are real & distinct

$$\text{Root 1} = -0.381966011$$

$$\text{Root 2} = -2.6180339$$

Enter the coefficient a,b,c

5 6 5

Roots are Imaginary

~~$$\text{Root 1} = 0.0 + i 0.8$$~~

~~$$\text{Root 2} = 0.0 - i 0.8$$~~

SC  
12/12/2023

19/12/23

Lab → 2

CLASSMATE

Date \_\_\_\_\_

Page \_\_\_\_\_

5

# Develop a Java program to create a class Student with members USN, name, an array Credits and subject marks. Include methods to except and display details; and a method to calculate SGPA of a student.

```
import java.util.*;
public class main {
 public static void main (String args[]){
 Student S1 = new Student();
 S1.get Student Details();
 S1.get marks();
 S1.computer SGPA();
 }
}
```

```
System.out.println("Name :" + S1.name);
System.out.println("USN :" + S1.USN);
System.out.println("SGPA :" + S1.SGPA);
}
}
```

Class Subject

```
int Subjectmarks;
int Credits;
int grades;
}
```

CLASS Student  
S

Subject Subject [ ];

String name;

String vsn;

double SGPA;

Scanner S;

Student ()

{

Subject = new Subject [ 9 ];

For ( int i = 0; i < 9; i++ )

{

Subject [ i ] = new Subject ();

}

S = new Scanner ( System.in );

void ~~get~~ Student Details ()

System.out.print ("Enter your name: ");

this.name = S.nextLine();

System.out.print ("Enter your vsn: ");

this.vsn = S.next ();

}

Void getStudentDetails()

{  
System.out.print("Enter your name:");  
this.name = s.nextLine();

System.out.print("Enter your USN:");  
this.USN = s.nextLine();

{

Void getMarks()

{  
For (int i=0; i<8; i++)

System.out.print("Enter the marks of  
the " + i + " Subject");

Subject[i].SubjectMarks = s.nextInt();

System.out.print("Enter the credits of  
the " + i + " Subject");

Subject[i].Credits = s.nextInt();

Subject[i].grades = (Subject[i].Subject  
marks / 10);

{ If (Subject[i].grades > 10)

Subject[i].grades = 10;

{

If (Subject[i].grades < 4)

Subject[i].grades = 0;

void computeSGPA()

int sum = 0;

int totalCredits = 0;

for (int i=0; i<9; i++)

Sum = (Subject[i].grades \* Subject[i].credits);

Total Credits = Subject[i].credits;

this.SGPA = (double) Sum / totalCredits;

}

else

else

else

else

QVFT

Enter your VSN:

Lbm21CS258

Enter the mark of 1 Sub - 85

Enter the Credit of 1 Sub - 3

Enter the mark of 1 Subject - 75  
Credits - 3

Enter the marks of 3 Subject - 82  
Credits - 4

Enter the marks of 4 Subject - 71  
Credits - 3

Enter the marks of 5 Subject - 89  
Credits - 4

Enter the marks of 6 Subject - 71  
Credits - 3

Enter the marks of 7 Subject - 75  
Credits - 3

Enter the marks of 8 subjects  $\rightarrow 45$

Credits  $\rightarrow 3$

Name: Shiv

VSN Name: LBm21CS258

SGPA: 7.1636

SC  
19/12/2023

→ Shlv Sundar San (LBm22 CS258) CLASSMATE  
Lab → 3 (To Create a Class Books) Date 26/12  
Page

- # Create a Class Book which contains  
your members. Include a constructor,  
Develop a Java program to Create an  
book objects.

Import java.util.Scanner;

Class Books {

String name;

String author;

int price;

int numPages;

public Books( String name, String  
author, int price, int numPages ) {

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}

public String toString() {

String name, author, price, numPages;

name = "Book name;" + this.name + "\n";

author = "Author name;" + this.author  
+ "\n");

price = "price:" + this.price + "In";

numPages = "Number of pages:" + this.pages + "\n";

return name + author + price + numPages;

}

Class main {

public static void main (String args[]){

Scanner sc = new Scanner (System.in);

int n, price, numPages;

String name, author;

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

n = sc.nextInt();

sc.nextLine();

Books b[] = new Books[n];

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

System.out.print ("Read name of the  
book");

name = sc.nextLine();

System.out.print ("Read author of the  
~~price~~ book");

author = sc.nextLine();

System.out.println ("Read the price  
of the book");

```
price = sc.nextInt();
```

```
System.out.println("Read pg numbers of
the book");
```

```
numPages = sc.nextInt();
```

```
sc.nextLine();
```

```
System.out.printIn(".....");
```

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

{}

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

```
String bookDetails = b[i].to
String();
```

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

{}

{}

#

Output

Enter the number of books

2

Read the name of books,

Chemistry

Read Author of the book  
John.

Read the price of the book  
289

Read the Pg No of the book  
298

-----  
Read name of the book - physics

Read Author of the book  
Raghav

Read the price of the book  
399

Read the Pg No of the book  
250

Book name : Chemistry

Author name : John

Price : 299

USN:18M22CS158

No. of Pages : 150

Book name : Physics

Author name : Raghav

Price : 399

Pg No : 250

26/12/22

- \* 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 contains only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
class InputScanner{
 Scanner s;
 InputScanner(){
 s = new Scanner(System.in);
 }
}
```

~~Abstract class Shape extends InputScanner~~

```
double a;
double b;
```

```
Abstract void getInputs();
Abstract void displayArea();
```

Class Rectangle extends shape {

void get Input() {

InputScanner is = new InputScanner;

System.out.print("Enter the length  
and breadth of the rectangle : ");

a = is.s.nextDouble();

b = is.s.nextDouble();

void display Area() {

System.out.println("The area of  
the rectangle is: " + (a \* b));

}

Class Triangle extends shape {

void get Input() {

InputScanner is = new InputScanner;

System.out.print("Enter ( ) ;  
the base and height of the triangle : ");

a = is.s.nextDouble();

b = is.s.nextDouble();

}

void display Area() {

System.out.println("The area of the  
triangle is: " + (a \* b \* 0.5));

}

{}

Class Circle extends Shape {

    void get Input() {

        Input Scanner is = new Input Scanner;  
        System.out.println("Enter radius of  
                          the circle");

        a = is.nextDouble();

{}

    void display Area() {

        System.out.println("The area  
                          of the circle is: " + (3.14 \* a \* b));

{}

public class Abstract main {

    public static void main(String args  
                                               []){}

        System.out.println("Hello");

        Rectangle rect = new Rectangle();

        rect.get Input();

        rect.display Area();

        Triangle triangle = new Triangle();

        triangle.get Input();

        triangle.display Area();

```
circle circle = new circle();
circle .(get Input());
circle .(display Area());
```

```
System.out.println("Name : shivsundarSah");
System.out.println("USN : LBm22CS258");
```

{}

Output:

Enter the length &amp; breadth of rectangle :

5 6

The area of rectangle is : 30.0

Enter the base &amp; height of triangle : 8 9

The area of triangle is 36.0

Enter the radius of the circle : 4.5

The area of circle is : 63.5850

Name : Shiv Sundar Sah

USN : LBm22CS258

~~Shiv  
USN~~

2024/01/08  
Tuesday  
Lab-5

classmate

Date

## CLASS BANK ACCOUNT

- 5 Develop a Java program which has class Bank account & Savings account which extends Accounts.

import java.util.Scanner;  
Class Account

{

String name;

int accno;

String type;

double balance;

Account(String name, int accno, String

type, double balance)

{

this.name = name;

this.accno = accno;

this.type = type;

this.balance = balance;

}

void deposit(double amount){

balance += amount;

}

```
void withdraw(double amount){
```

```
 if (balance - amount >= 0)
```

```
 balance -= amount;
```

```
 else,
```

```
 System.out.println("no sufficient
```

```
balance, cannot withdraw")
```

```
}
```

```
void display(){}
```

```
System.out.println("Name:" + name)
```

```
"Account No:" + accno + "type:" + type)
```

```
"type" + "balance:" + balance);
```

```
}
```

```
}
```

```
class Savingacc extends account{
```

```
private static double rate = 5;
```

```
Saving acc(string name, int accno,
```

```
double balance){
```

```
super(name, accno, type: "Savings", balance);
```

```
void interest(){}
```

```
balance += balance * (rate / 100);
```

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

```
}
```

```
}
```

CLASS CURRENT ACCOUNT EXTENDS CURRENT  
PRIVATE double minbal = 1000;  
PRIVATE double servicecharge = 50;  
CURRENTACCT (string name, int accno,  
double balance)

{

}

void checkmin()

{ if (balance &lt; minBal)

System.out.println("balance  
is less than min balance,"service charges imposed :" + service  
(charges) ;

balance -&gt; service charges;

System.out.println("balance is :" +  
balance");

}

CLASS BANK {

public static void main (String args[])

```
Scanner s = new Scanner (System.in);
System.out.printIn ("enter the name:");
String name = s.next();
System.out.printIn ("enter the type");
String type = s.next();
String type = s.next();
System.out.printIn ("enter the account number:");
int accno = s.nextInt();
System.out.printIn ("enter the initial balance:");
double balance = s.nextDouble();
int ch;
double amount1, amount2;
Account acc = new Account (name,
 type, balance);
Saving acc = new Saving Acc (name,
 accno, balance);
Current Acc = new Current Acc (name,
 accno, balance);
```

while (true)

{ if (acc.type equals ("saving"))

```
System.out.println("1. menu\n2. deposit
3. withdraw 4. compute interest 5.
display");
```

```
System.out.println("enter the choice");
ch = s.nextInt();
switch (ch) {
```

Case 1: System.out.println ("enter  
the amount");

```
amount1 = s.nextInt();
sa.deposit(amount1);
break;
```

Case 2: System.out.println ("enter  
the amount");

```
amount2 = s.nextInt();
```

```
sa.withdraw(amount2);
break;
```

Case 3: sa.Interest();

```
break;
```

Case 4: sa.display();

```
break;
```

Case 5: system.exit(0);

default: System.out.println("Invalid  
input");

```
 } break;
 }
else {
}
```

System.out.println("In menu In 1.deposit  
2.withdraw. 3.display");

System.out.println("enter the choice");  
ch = s.nextInt();  
switch(ch){

Case 1: System.out.println("enter the  
amount");

Amount1 = s.nextInt();  
ca.deposit(amount1);  
break;

Case 2: System.out.println("enter the  
amount");

Amount2 = s.nextInt();  
ca.withdraw(amount2);  
ca.checkmin();  
break;

Case 3: ca.display();  
break;

Case 4: System.exit(0);

```
} } }
```

Output.

Enter the name : Shiv Sundar Seth

Enter the type (Current/Savings);  
Savings.

Enter the account number

1010152055

Enter the initial balance : 8492500.00

Menu.

1. Deposit 2. withdraw 3. display 4.

Compute Interest for Sq. 5. Exit

Enter your choice

1

Enter the deposit amount

15.00

menu.

1. Deposit 2. withdraw 3. compute interest

4. display 5. Exit

9/1/2024  
4

Name : Shivasundar, accno : 1010152055

Type : Savings, balance : 8494000.00

Enter the account type (current or deposit)  
deposit

Enter the amount: 450010345

Enter the initial : 1000

1. Deposit.
2. withdraw
3. Display A/c
4. Exit

Enter the choice: 3

Name: shiv sundar sat

Acount no: 450010345

Type: Deposit.

balance : 1000.0

1. Deposit.
2. withdraw
3. Display A/c.
4. Exit

2.withdraw

Enter the amount:

5200.0

~~Insufficient balance~~

Balance is less than minbalance service

Charge exeeded 100.

balance is RS 800.0

S&F  
16/11/2024

2024/01/23

Tuesday

Lab-6

Package

classmate

Date

Page

26

6

Develop a Java program & create a CSE package consists of Students and Internals classes and SFE package

1)

CSE Student.java

package CSE;

import java.util.\*;

public class Student {

protected String name = new String();

protected String usn = new String();

protected int Sem

public void InputStudentDetails() {

Scanner sc = new Scanner(System  
. in);

System.out.println("Name");

Name = sc.nextLine();

System.out.println("USN");

USN = nextLine();

System.out.println("Sem");

Sem = s.nextInt();

}

public void display() {

System.out.println("Student Details");

System.out.println("Name: " + name);

System.out.println("USN: " + usn);

System.out.println("Sem" + sem);  
CSE - Internals Java.

package CSE;

import java.util.\*;

```
public class Internals extends Student {
 protected int marks[] = new int[5];
 public void Input() {
 Scanner s = new Scanner(System.in);
 System.out.println("Enter the
 marks of 5 students");
 for (int i = 0; i < 5; i++) {
 marks[i] = s.nextInt();
 }
 }
}
```

SCE Internals.java

package SCE;

import CSE.Internals;

import java.util.\*;

public class Internals extend Internals

protected int marks[];

protected int final marks[];

public void Input SCE() {

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

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

```
 System.out.print("Student" + (i+1) + "
marks");
```

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

{

```
public void marks() {
```

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

```
 finalmarks[i] = marks[i]/2 + super.marks[i];
```

{

```
public void displaymarks() {
```

```
 display();
```

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

```
 System.out.println("Subject" +
```

```
 ((i+1) + " " + finalmarks[i]);
```

{

{

```
Studentmain.java
```

```
package CSE;
```

```
import SEF.Externals;
```

```
public class Studentmain {
```

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

```
Int n = 1;
Externalus finalmarks [] = new
Externalus [n];
For (int i=0; i<n; i++) {
 finalmarks [i] = new Externalus();
 finalmarks [i].inputs student detail
 System.out.printIn ("Enter SEE marks");
 finalmarks [i].inputSee();
 System.out.printIn ("Enter SEE
 marks");
 finalmarks [i].InputSee();
}
System.out.printIn ("Display data");
for (int i=0; i<n; i++) {
 finalmarks [i].ctrmarks();
 finalmarks [i].displaymarks();
}
```

DATA STRUCTURE

QUESTION PAPER

QUESTION PAPER

QUESTION PAPER

QUESTION PAPER

Output:

Name: Shiv Sundar Seth  
USN: LB072205258

SEM: 3

Enter the CSE marks.

50 49 47 46 48

Enter SEE marks.

Subject 1 marks: 98

Sub 2 Marks: 95

Sub 3 Marks 89

Sub 4 Marks 91

Sub 5 Marks 92

Display data.

Name: Shiv Sundar Seth

USN: LB072205258

SEM: 3

Sub 1: 98

Sub 2: 97

Sub 3: 90

Sub 4: 92

Sub 5: 91

S  
30/11/2024

2024/01/30

Tuesday

Lab-7

CLASSMATE

Date

Page

31

## \* Exception Handling

\* pmain

7. WAP that demonstrate handling of exception in inheritance tree. Create a base class called "Father" & derives class called "Son" which extends the base class. Implement a constructor that takes both Father and Son's age and throws an exception if son's age is  $\geq$  Father age.

Import java.util.\*;

Class WrongAge extends Exception {  
    public WrongAge (String s) {  
        super(s);  
    }

Class InputScanner {  
    Scanner sc;

    public InputScanner () {  
        sc = new Scanner (System.in);  
    }

}

Class Father extends InputScanner {  
 int FatherAge;

public Father () throws WrongAge {

    InputScanner ss = new InputScanner();  
 System.out.print("Enter the father age:");  
 FatherAge = ss.sc.nextInt();  
 if (FatherAge < 0) {

        throw new WrongAge("Age cannot be negative");

}

    Void Edisplay() {

        System.out.print("Father age is: " + FatherAge);

}

}

int sonAge;

~~public Son () throws WrongAge {~~

~~InputScanner ss = new InputScanner();~~

~~SonAge = ss.sc.nextInt();~~

~~if (SonAge > FatherAge) {~~

~~throw new WrongAge("Age cannot be greater than father's age");~~

Son's age cannot be greater than  
 father's age);

}

{}

else if (sonAge) {

throw new WrongAge("Son's

age cannot be greater than father's  
age");

{}

else if (sonAge == 0) {

throw new WrongAge("

Age cannot be negative");

{ } } else if (sonAge == fatherAge) {

throw new WrongAge("Age cannot be  
same");

void sdisplay() {

System.out.println("Son's

age is : " + sonAge);

{ }

{ }

public class ExceptionHandling

public static void main (String  
args[]) {

Son p;

try {

p = new Son();

p.sdisplay();

p.sdisplay();

{ }

Causes (Wrong Age. e) {  
System.out.println();  
}

{ }  
{ }

## Output:

Enter the Father age: 12

Enter the Son's age : 25

Wrong Age: Son's age cannot be greater  
than Father's age

Enter the Father age: - 25

Wrong Age: Age cannot be negative

Enter the Father age :- 35

Enter the Son age :- 15

Father age is 35

Son age is 15.

Enter father's age - 12

Enter son's age - 12

Wrong Age: Age cannot be same

8/8/2024

2024/02/06

Tuesday

## Lab. program - 8

classmate

Date

Page

35

- 8 WAP which creates two threads one thread displaying "BMS College of Engineering" once every ten second and another displaying "CSE" once every two second.

# Class Thread implements Runnable

```
Thread t;
public Thread() {
}
```

```
t = new Thread(this, "NThread");
System.out.println("CT:" + t);
t.start();
```

```
}
```

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

try

```
for (int n = 5; n > 0; n--) {
```

```
{
```

System.out.println("CSE" + n);  
Thread.sleep(2000);

Classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
catch (InterruptedException ie) {
 System.out.println("BNSCE Interrupted");
 System.out.println("BNSCE quitting");
}
}
```

```
public class PrintCollege {
 public static void main (String ssc)
{
```

```
 new Thread () {
 System.out.println ("Black in main");
 for (int n=2; n>0; n-)
 {
 System.out.println ("BNSCE"+n);
 Thread.sleep (1000);
 }
 }.start();
}
```

```
 catch (InterruptedException ie) {
 System.out.println ("BNSCE Interrupted");
 }
}
```

```
 System.out.println ("BNSCE quitting");
}
}
```

Output:

Thread [NT Thread, 5, main]

Backing main

CSE 5

Bmscf 2

CSE 4

CSE 3

CSE 2

CSE 1

Bmscf 1

CSE Quitting

Bmscf quitting

28  
6/2/2024

2024/02/20

Tuesday

Lab → 9

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

# WAP that creates a User Interface to perform integer divisions. The user edits two no in the text field num1 & num2. The division of num1 & num2 is displayed in the Result field when Divide button is clicked.

```
Import java.awt.*;
Import java.awt.event.*;
```

```
public class Division extends Frame
implements ActionListener
{
```

```
Text Field num1, num2;
```

```
Button dResult;
```

```
Label rResult;
```

```
String out = " ";
```

```
double resultNum;
```

```
int j10f = 0;
```

```
public Division main()
```

```
{
 setLayout(new FlowLayout());
 JResult = new JButton("RESULT");
```

```
 Label numberL = new Label("Number L:", Label.RIGHT);
```

```
 R1GHT);
```

Label Number 2 = new Label ("Number 2,"

Label.RIGHT);

num1 = new Text Field (5);

num2 = new Text Field (5);

outResult = new Label ("Result:", like  
RIGHT);

Add (number1)

Add (number2);

Add (number3);

Add (dResult);

Add (outResult);

num1.addActionListener (this);

num2.addActionListener (this);

dResult.addActionListener (this);

Add windowListener (new Window

Adapter ())

{}

public void windowClosing

(WindowEvent we)

System.exit(0);

int a = num1.getText () + num2.getText ();

a = a \* 2; num3.setText (String.valueOf (a));

3. write a program to calculate area of circle.

import java.awt;

public void actionPerformed(ActionEvent e)

{ int n1, n2;

{ try

{ if (ae.getSource() == dResult)

n1 = Integer.parseInt(displayText.getText());

n2 = Integer.parseInt(displayText.getText());

/\* if (n2 == 0)

throw new ArithmeticException();

out = n1 + " / " + n2;

resultNum = n1 / n2;

out += String.valueOf(resultNum);

} repaint();

}

Catch { NumberFormatException e1 }

{

flag = 1;

out += "Number Format Exception! " + "fel";

} repaint();

}

public void paint(Graphics g)

```
if (yaf == 0)
 g.drawString("Out", 0, 0 + result.getFont().getHeight())
 + OUT_RESULT.getFont().getWidth(), 0 + result.getFont().getHeight() + OUT_RESULT + "get Height()").align
else
 g.drawString("Out", 100, 200);
 if (yaf == 0);
```

public static void main(String[] args)

```
Division dm = new Division();
dm.setSize(new Dimension(600, 400));
dm.setTitle("Division of integers");
dm.setVisible(true);
```

Output<sup>o</sup>

Number 1: 10

Number 2: 5

Result: 1052.0

Number 1: 20

Number 2: 2

Result: 20210.0

~~SS  
20210.024~~

# Lap-program-Y20

## Open Ended Exercise.

- \* Demonstrate Inter Process communication and deadlock.

### i) IPC program

Class QF

Int n;

boolean valueset = False;

Synchronized int get()

while (!valueset)

try

System.out.println("In consumer  
waiting\n");

wait();

}

catch (InterruptedException e) {

System.out.println("Interrupted Exception  
Caught");

}

System.out.println("Got : " + n);

valueset = true;

System.out.println("In intimate producer\n");

notify();

```
 return n;
}
```

```
synchronized void put(int n) {
 while (!valueSet)
 try {
```

```
 System.out.println("In producer waiting");
 wait();
```

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

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

```
}
```

```
 this.n = n;
```

```
 valueSet = true;
```

```
 System.out.println("put: " + n);
```

```
 System.out.println("In Intimate consumer
notify");
```

```
}
```

~~class Producer implements Runnable~~

```
Q q;
```

```
producer(Q q) {
```

```
this.Q = q;
```

```
new Thread(this, "producer").start();
```

```
}
```

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

```
int i=0;
while (i<15) {
 q.put(i+1);
 }
 }
}
```

Class consumer implements Runnable

```
(Q q;
Consumer (Q q) {
 this.q=q;
 new Thread (this, "consumer").start();
}
```

```
public void run () {
 int i=0;
 while (i<15) {
 int t=q.get();
 System.out.println ("consumed : " + t);
 q.put(t+1);
 }
}
```

Class Ipc

```
public static void main (String args []) {
 Q q=new Q ();
 new Producer (q);
 new Consumer (q);
}
```

System.out.println("press control - c to  
stop");

?

Output:

Intimate consumer  
Producer waiting

press control - c to stop.

Got: 0

Intimate producer

consumed: 0

pvt: 1

Intimate consumer  
Producer waiting  
Got: 1

Intimate producer

consumer: 1

pvt: 2

Intimate producer

Consumers →

producer waiting  
Got 2

Intimate producer

Consumers: 2

Put: 3

Intimate consumer

producer waiting

Got: 3

Intimate consumer producer

Got consumers: 3

Put: 4

Intimate consumer

producer waiting

Got: 4

Intimate consumer

consumers: 4

Put: 5

## Q. 9) Deadlock program

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(name + " trying to call
B.last()");
```

```
b.last();
```

```
}
```

```
void last() {
```

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

```
}
```

Class B {

```
Synchronized void bar(A a) {
```

```
String name = Thread.currentThread().
```

```
get.Name();
```

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

```
try {
```

```
Thread.sleep(1000);
```

```
} catch (Exception e) {
 System.out.println("B interrupted");
}
```

```
System.out.println("name=" + name + " thing to call
last()");
}
```

```
A.last();
}
```

```
void last() {
```

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

Class Deadlock Implements Runnable

```
Aa = new A();
```

```
B b = new B();
```

```
deadlock();
```

```
Thread.currentThread().setName("main
Thread");
```

~~```
Thread t = new Thread(this, "Racing  
Thread");
```~~

```
t.start();
```

```
a.Foo(); b(); // get lock once in this  
thread.
```

```
public void run() {  
    b.bar(a);  
}
```

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

}
public static void main(String args){
 new Deadlock();

}
Output

Main Thread entered A.last

Racing Thread entered B.back

Main thread 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.

S
13/2/2024

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 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 coefficients of a,b,c");
        a=s.nextInt();
        b=s.nextInt();
        c=s.nextInt();
    }
    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();
        }
        d=b*b-4*a*c;
        if(d==0)
        {
            r1=(-b)/(2*a);
            System.out.println("Roots are real and equal");
        }
    }
}
```

```

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

    }

    else if(d>0)

    {

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

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

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

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

    }

    else if(d<0)

    {

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

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

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

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

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

    }

}

}

class QuadraticMain

{

    public static void main(String[] args)

    {

        Quadratic q=new Quadratic();

        q.getd();

        q.compute();

        System.out.println("Shiv sundar sah1BM22CS258");

    }

}

```

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 Subject{  
    int subjectMarks;  
    int credits;  
    int grades;  
}
```

```
class Student{  
    Subject subject[];  
    String name;  
    String usn;  
    double SGPA;  
    Scanner s;  
  
    Student(){  
        subject = new Subject[9];
```

```
        for(int i = 0;i<9;i++){  
            subject[i] = new Subject();  
        }  
        s= new Scanner(System.in);  
    }
```

```
    void getStudentDetails(){
```

```

        System.out.println("Enter your name: ");
        this.name = s.nextLine();

        System.out.println("Enter your usn: ");
        this.usn = s.next();

    }

void getMarks(){
    for(int i = 0;i<8;i++){
        System.out.println("Enter the marks of the "+(i+1)+" subject");
        subject[i].subjectMarks = s.nextInt();

        System.out.println("Enter the credits of the "+(i+1)+" subject");
        subject[i].credits = s.nextInt();
        subject[i].grades = (subject[i].subjectMarks/10)+1;

        if(subject[i].grades >10){
            subject[i].grades = 10;
        }
        if(subject[i].grades <4){
            subject[i].grades = 0;
        }
    }
}

void computeSGPA(){
    int sum=0;
    int totalCredits = 0;
    for(int i = 0;i<9;i++){
        sum+=(subject[i].grades * subject[i].credits);
        totalCredits += subject[i].credits;
    }
}

```

```

        }

        this.SGPA = (double) sum/totalCredits;

    }

}

public class MainSGPA{

    public static void main(String args[]){

        Student s1 = new Student();

        s1.getStudentDetails();

        s1.getMarks();

        s1.computeSGPA();

        System.out.println("Name: "+s1.name);

        System.out.println("Usn: "+s1.usn);

        System.out.println("SGPA: "+s1.SGPA);

        System.out.println("Shivsundar sah 1BM22CS258");

    }

}

```

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.Scanner;
```

```
class Books{
```

```
String name;  
String author;  
int price;  
int numPages;  
  
public Books(String name, String author, int price, int numPages){  
    this.name=name;  
    this.author=author;  
    this.price=price;  
    this.numPages=numPages;  
}  
  
public String toString(){  
    String name,author,price,numPages;  
    name="Book name:" + this.name + "\n";  
    author="Author name:" + this.author + "\n";  
    price="Price :" +this.price+ "\n";  
    numPages="No of Pages : "+this.numPages+"\n";  
    return name + author + price + numPages;  
}  
  
String getName(){  
    this.name=name;  
}  
  
String getAuthor(){  
    this.author=author;  
}  
  
int getPrice(){  
    this.price=price;
```

```
}

int getNumPages(){
    this.numPages=numPages;
}

}

class MainB{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n,price,numPages;
        String name,author;

        System.out.println("Enter no of books :");
        n= s.nextInt();
        s.nextLine();

        Books b[];
        b= new Books[n];

        for(int i=0;i<n;i++){
            System.out.println("Enter the name of book"+(i+1)+":");
            name=s.nextLine();
            System.out.println("Enter Author of book"+(i+1)+":");
            author=s.nextLine();
            System.out.println("Enter price of book"+(i+1)+":");
            price=s.nextInt();
            System.out.println("Enter no of pages of book"+(i+1)+":");
            numPages=s.nextInt();
            s.nextLine();
            b[i]=new Books(name,author,price,numPages);
        }
    }
}
```

```

    }

    for(i=0;i<n;i++){
        String bookDetails=b[i].toString();
        System.out.println(bookDetails);
    }

    for(int i=0;i<n;i++){
        System.out.println("Book "+(i+1)+":");
        System.out.println("Name :" + b[i].getName());
        System.out.println("Author :" + b[i].getAuthor());
        System.out.println("Price :" + b[i].getPrice());
        System.out.println("No of pages :" + b[i].getNumPages());
        System.out.println("*****");
    }

    System.out.println("Shivsundar sah \n 1BM22CS258");

}

}

```

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.

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

```
class InputScanner{
```

```
    Scanner s;
```

```
InputScanner() {  
    s = new Scanner(System.in);  
}  
}  
  
abstract class Shape extends InputScanner{  
    double a;  
    double b;  
    abstract void getInput();  
    abstract void displayArea();  
}  
  
class Rectangle extends Shape{  
    void getInput(){  
        InputScanner sc=new InputScanner();  
        System.out.println("Enter the length and breadth of rectangle");  
        a=sc.s.nextInt();  
        b=sc.s.nextInt();  
    }  
    void displayArea(){  
        double area_rect=a*b;  
        System.out.println("Area of retangle is : "+area_rect);  
    }  
}  
  
class Triangle extends Shape{  
    void getInput(){  
        InputScanner sc=new InputScanner();  
        System.out.println("Enter the base and height of triangle:");  
        a=sc.s.nextInt();  
        b=sc.s.nextInt();  
    }  
}
```

```

        }

void displayArea(){

    double area_tri=a*b/2;

    System.out.println("Area of tritangle is : "+area_tri);

}

}

class Circle extends Shape{

void getInput(){

    InputScanner sc=new InputScanner();

    System.out.println("Enter the radius of circle:");

    a=sc.s.nextInt();




}

void displayArea(){

    double area_circle=3.14*a*a;

    System.out.println("Area of circle is : "+area_circle);

}

}

public class AbstractMain{

public static void main(String args[]){

    Rectangle a = new Rectangle();

    a.getInput();

    a.displayArea();



    Triangle b = new Triangle();

    b.getInput();

    b.displayArea();



}

```

```

        Circle c = new Circle();
        c.getInput();
        c.displayArea();

        System.out.println("Shivsundar sah 1BM22CS258");

    }
}

```

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.

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

```

class Account{
    String name;
    int accNo;
    String type;
    double balance;
}

```

```

Account(String name,int accNo,String type,double balance){

    this.name=name;
    this.accNo=accNo;
    this.type=type;
    this.balance=balance;
}

void deposit(double amt){

    balance=balance+amt;
}

void withdraw(double amt){

    if(balance<amt){
        System.out.println("Insufficient Balance");
    }
    else{
        balance=balance-amt;
    }
}

void display(){

    System.out.println("Name:"+name+"\tAccount No:"+accNo+"\tAccount
Type:"+type+"\tBalance"+balance);
}

class Savings_acc extends Account{

    private static double rate= 3.5;

    Savings_acc(String name,int accNo,double balance){

        super(name,accNo,"savings",balance);
    }
}

```

```

    }

void callInt(){
    double interest=(balance*rate)/100;
    System.out.println("Interest is "+interest);
}

}

class Current_acc extends Account{
    private double minBal=500;
    double s_charges=50;

    Current_acc(String name,int accNo,double balance){
        super(name,accNo,"current",balance);
    }

    void check_bal(){
        if(balance<minBal){
            System.out.println("Insufficient Balance");
            balance=balance-s_charges;

        }
        System.out.println("Balance =" +balance);
    }

}

public class Bank{
    public static void main(String args[]){
        String name;
        int AccNo;
        String Type;
    }
}

```

```
double init_bal;

Scanner s=new Scanner(System.in);

System.out.println("Enter Customer Name:");

name=s.nextLine();

System.out.println("Enter Account No:");

AccNo=s.nextInt();

System.out.println("Enter Account Type:");

Type=s.next();

System.out.println("Enter Initial Balance:");

init_bal=s.nextDouble();

double amt;

Account a=new Account(name,AccNo,Type,init_bal);

Savings_acc sv=new Savings_acc(name,AccNo,init_bal);

Current_acc ca=new Current_acc(name,AccNo,init_bal);

while(true){

if(Type.equalsIgnoreCase("savings")){

    System.out.println("----MENU----");

    System.out.println("Enter 1:Deposit 2:Withdraw 3:Interest

4:Display Details 5:Exit");

    int ch=s.nextInt();

    switch(ch){

        case 1:

            System.out.println("Enter The Amount:");

            amt=s.nextDouble();

            a.deposit(amt);

            break;

        case 2:

            System.out.println("Enter the withdrawing

amount");

            amt=s.nextDouble();
```

```

        a.withdraw(amt);

        break;

    case 3:

        sv.callInt();

        break;

    case 4:

        a.display();

        break;

    case 5:

        System.exit(0);

    default:

        System.out.println("Invalid Choice");

    }

}

else{

    System.out.println("----MENU---");

    System.out.println("Enter 1:Deposit 2:Withdraw 3:Display Details
4:Exit");

    int ch=s.nextInt();

    switch(ch){

        case 1:

            System.out.println("Enter The Amount:");

            amt=s.nextDouble();

            a.deposit(amt);

            break;

        case 2:

            System.out.println("Enter the withdrawing
amount");

            amt=s.nextDouble();

            a.withdraw(amt);
    }
}

```

```

        ca.check_bal();
        break;
    case 3:
        a.display();
        break;
    case 4:
        System.exit(0);
    default:
        System.out.println("Invalid Choice");
    }
}

System.out.println("Shiv sundar sah 1BM22CS258");
}
}

}
}

```

6)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 of n students in all five courses.

```

package CIE;
import java.util.*;

```

```
public class Student{  
    protected String usn=new String();  
    protected String name =new String();  
    protected int sem;  
  
    public void inputStudentDetails(){  
        Scanner s=new Scanner(System.in);  
        this.usn=s.nextLine();  
        this.name=s.nextLine();  
        this.sem=s.nextInt();  
    }  
  
    public void displayStudentDetails(){  
        System.out.println(this.usn+" "+this.name+" "+this.sem);  
    }  
}
```

```
package CIE;  
import java.util.Scanner;  
public class Internals extends Student{  
    protected int marks[]=new int[5];  
    public void inputCIEmarks(){  
        Scanner s=new Scanner(System.in);  
        for(int i=0;i<5;i++){  
            marks[i]=s.nextInt();  
        }  
    }  
}  
  
package SEE;
```

```

import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals{
    protected int marks[];
    protected int finalMarks[];

    public Externals(){
        marks =new int[5];
        finalMarks=new int[5];
    }

    public void inputSEEmarks(){
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5;i++){
            System.out.print("Subject "+(i+1)+" marks: ");
            marks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for(int i=0;i<5;i++)
            finalMarks[i] = marks[i]/2 + super.marks[i];
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        for(int i=0;i<5;i++)
            System.out.println("Subject " + (i+1) + ":" + finalMarks[i]);
    }
}

```

```

}

import SEE.*;

class Main1{
    public static void main(String args[]){
        int num=2;

        Externals finalMarks[] = new Externals[num];

        for(int i=0;i<num;i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }

        System.out.println("Displaying Data:\n");
        for(int i=0;i<num;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }

        System.out.println("shiv sundar sah1BM22CS258");
    }
}

```

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.

```
import java.util.Scanner;

class WrongAge extends Exception{
    WrongAge(String s){
        super(s);
    }
}

class InputScanner{
    Scanner sc;
    InputScanner(){
        sc=new Scanner(System.in);
    }
}

class Father extends InputScanner{
    int fatherAge;
    public Father() throws WrongAge{
        InputScanner sf=new InputScanner();
        fatherAge=sf.sc.nextInt();
        if(fatherAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void Fdisplay(){
        System.out.println("Father's Age: "+fatherAge);
    }
}

class Son extends Father{
```

```

int sonAge;

public Son() throws WrongAge{
    InputScanner ss=new InputScanner();
    sonAge=ss.sc.nextInt();
    if(sonAge>=fatherAge){
        throw new WrongAge("Son's age cannot be greater than father's age");
    }
    else if(sonAge<0){
        throw new WrongAge("Age cannot be negative");
    }
}

void Sdisplay(){
    System.out.println("Son's Age: "+sonAge);
}
}

public class AgeCheck{
    public static void main(String args[]){
        Son a;
        try{
            a=new Son();
            a.Fdisplay();
            a.Sdisplay();
        }
        catch(WrongAge e){
            System.out.println(e);
        }
        System.out.println("Shiv sundar sah 1BM22CS258");
    }
}

```

```
}
```

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 Thread1 implements Runnable{
```

```
    Thread t;
```

```
    public Thread1()
```

```
{
```

```
    t=new Thread(this, "NThread");
```

```
    System.out.println("CT:"+t);
```

```
    t.start();
```

```
}
```

```
    public void run()
```

```
{
```

```
    try
```

```
{
```

```
        for(int n=5;n>0;n--)
```

```
{
```

```
        System.out.println("CSE "+n);
        Thread.sleep(2000);

    }

}

catch(InterruptedException ie) {
    System.out.println("CSE Interrupted");
    System.out.println("CSE quitting");

}

}

public class PrintColleg {

    public static void main(String ss[])
    {
        new Thread1();
        System.out.println("Back in main");
        try
        {
            for(int n=2;n>0;n--)
            {
                System.out.println("BMSCE "+n);
                Thread.sleep(10000);
            }
        }
    }
}
```

```
    }

    catch(InterruptedException ie){

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

    }

    System.out.println("BMSCE quitting");

    System.out.println("shiv sundar sah 1BM22CS258");

}

}
```

10) Demonstrate Inter process Communication and deadlock

a) Inter process Communication

```
class Q {

    int n;

    boolean valueSet = false;

    synchronized int get() {

        while(!valueSet)

            try {

                System.out.println("\nConsumer waiting\n");

```

```
    wait();

} catch(InterruptedException e) {

    System.out.println("InterruptedException caught");

}

System.out.println("Got: " + n);

valueSet = false;

System.out.println("\nIntimate Producer\n");

notify();

return n;

}

synchronized void put(int n) {

while(valueSet)

try {

    System.out.println("\nProducer waiting\n");

    wait();

} catch(InterruptedException e) {

    System.out.println("InterruptedException caught");
```

```
    }

    this.n = n;

    valueSet = true;

    System.out.println("Put: " + n);

    System.out.println("\nIntimate Consumer\n");

    notify();

}

}

class Producer implements Runnable {

    Q q;

    Producer(Q q) {

        this.q = q;

        new Thread(this, "Producer").start();

    }

    public void run() {

        int i = 0;
```

```
while(i<4) {  
  
    q.put(i++);  
  
}  
  
}  
  
}  
  
class Consumer implements Runnable {  
  
    Q q;  
  
    Consumer(Q q) {  
  
        this.q = q;  
  
        new Thread(this, "Consumer").start();  
  
    }  
  
    public void run() {  
  
        int i=0;  
  
        while(i<4) {  
  
            int r=q.get();  
  
            System.out.println("consumed:"+r);  
        }  
    }  
}
```

```
i++;

}

}

}

class PCFixed {

public static void main(String args[]) {

    Q q = new Q();

    new Producer(q);

    new Consumer(q);

    System.out.println("Press Control-C to stop.");

    System.out.println("shivsundar sah 1BM22CS258p");

}

}
```

b)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();

}

void last() {

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

}

}

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();  
  
}  
  
void last() {  
  
    System.out.println("Inside A.last");  
  
}  
  
}  
  
}  
  
class Deadlock implements Runnable  
{  
  
    A a = new A();
```

```
B b = new B();

Deadlock() {
    Thread.currentThread().setName("MainThread");

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

    t.start();

    a.foo(b); // get lock on a in thisthread.

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

}

public void run() {

    b.bar(a); // get lock on b in otherthread.

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

}

public static void main(String args[]) {

    new Deadlock()
    System.out.println("shivsundarsah 1BM22CS258");
}

}
```

9)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.

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

class SwingDemo1 {
    SwingDemo1(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and           divident:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");
```

```

// labels

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();

// add in order :)

jfrm.add(err); // to display error bois

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

ActionListener l = new ActionListener() {

    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text
                           field");
    }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);

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("\nA = " + a);
        blab.setText("\nB = " + b);
        anslab.setText("\nAns = " + ans);
    }

    catch(NumberFormatException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");

        err.setText("Enter Only Integers!");
    }

    catch(ArithmetricException e){
        alab.setText("");
        blab.setText("");
        anslab.setText("");

        err.setText("B should be NON zero!");
    }

}

});

// display frame
jfrm.setVisible(true);

}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){

        public void run(){
            new SwingDemo1();
        }
    });
}

```

```
    }  
});  
System.out.println("shivsundar sah 1BM22CS258");  
}  
}
```