

# B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



## LAB

## REPORT

## 23CS3PCOOJ

Submitted in partial fulfillment of the requirements for Lab  
Bachelor of Engineering  
in  
Computer Science and Engineering

Submitted by:  
PRANAV SRINIVAS  
(1BM22CS203)

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

# **INDEX**

<b>Sl. No.</b>	<b>Title</b>	<b>Date</b>
1	Complete scanned observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

## LABORATORY - 1

5TH DECEMBER, 2023

TUESDAY.

- 1) Write a program in JAVA to display your name.

```
import java.util.*;
class PranavSrinivas
{
    public static void main (String args[])
    {
        System.out.println ("PRANAV SRINIVAS");
    }
}
```

PRANAV SRINIVAS

- 2) Write a program in JAVA to accept an integer and check whether it's an even or an odd number.

```
import java.util.*;
class evenodd
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter any integer: ");
        int n = s.nextInt();
        if (n % 2 == 0)
            System.out.println (n + " is an even number.");
        else
            System.out.println (n + " is an odd number.");
    }
}
```

Enter any integer:

41 is an odd number

- 3) Write a program in JAVA to accept a positive integer 'x' and find the  $n^{\text{th}}$  root of the given positive integer.

```

import java.util.*;
class Pranav
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter a positive integer: ");
        int x = s.nextInt();
        while (x > 0)
        {
            System.out.println ("Enter the root: ");
            int n = s.nextInt();
            float a = pow (x, (1/n));
            break;
        }
        System.out.println ("x+" + "n" + "1/" + n + "=" + a);
    }
}

```

- 4) Write a program in JAVA to accept the number of units consumed in 'units' and to print the electricity bill using the following table:

Units consumed	Price per unit
Less than 100	₹ 1.20
For next 200 units	₹ 2.00
For next 100 units	₹ 3.00
Greater than 400	₹ 3.50

```
import java.util.*;
class ElectricityBill
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter number of units consumed:");
        float units = s.nextFloat();
        if (units < 100.0)
        {
            float billamount = (1.00 * units);
            System.out.println ("Bill Amount: " + billamount);
        }
        else if (units > 100.0 & units <= 300.0)
        {
            float billamount = (1.20 * 100) + ((units - 100) * 2.00);
            System.out.println ("Bill Amount: " + billamount);
        }
        else if (units > 300.0 & units <= 400.0)
        {
            float billamount = ((1.20 * 100) + (200.0 * 2.00) +
                (units - 300.0) * 3.00);
            System.out.println ("Bill Amount: " + billamount);
        }
        else
        {
            float billamount = ((1.20 * 100) + (200.0 * 2.00) + (100 * 1.50) +
                (units - 400) * 3.50);
            System.out.println ("Bill Amount: " + billamount);
        }
    }
}
```

Enter the number of units consumed: 472

Bill Amount: 1072.00.

- 5) Write a program in JAVA to accept the co-efficients a, b, c for a quadratic equation  $ax^2 + bx + c = 0$  and hence find the roots of the given quadratic equation.

```

import java.util.*;
class QE
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Quadratic equation: ax^2 + bx + c = 0");
        System.out.println("Enter the value of a: ");
        double a = s.nextDouble();
        System.out.println("Enter the value of b: ");
        double b = s.nextDouble();
        System.out.println("Enter the value of c: ");
        double c = s.nextDouble();
        double d = b * b - (4.0 * a * c);
        if (d > 0.0)
        {
            double r1 = ((-b + Math.sqrt(d)) / (2.0 * a));
            double r2 = ((-b - Math.sqrt(d)) / (2.0 * a));
            System.out.println("The roots are " + r1 + " and " + r2);
        }
        else if (d == 0.0)
        {
            double r1 = -b / (2.0 * a);
            System.out.println("The roots are real and equal: " + r1);
        }
        else
        {
            System.out.println("The roots are imaginary");
            double r1 = (-b / (2.0 * a));
            double r2 = ((d, 0.5) / (2.0 * a));
        }
    }
}

```

```
System.out.println("Roots: " + r1 + " and " + r1 + " - " + r2 + " i");  
}
```

{ Quadratic Equation:  $ax^2 + bx + c = 0$

{ Enter the value of a: 0

{ Not a quadratic equation. Enter non-zero value of a!!!

Quadratic Equation:  $ax^2 + bx + c = 0$

Enter the value of a: 2.3

Enter the value of b: 4

Enter the value of c: 5.6

The roots are imaginary

Roots:  $-0.87 + 1.30i$  and  $-0.87 - 1.30i$

Quadratic Equation:  $ax^2 + bx + c = 0$

Enter the value of a: 1

Enter the value of b: -6

Enter the value of c: 4

The roots are real and equal; 2

Quadratic Equation:  $ax^2 + bx + c = 0$

Enter the value of a: 1

Enter the value of b: -9

Enter the value of c: 10

The roots are -1 and 10

6) ~~Composite a program in JAVA to find the factorial of a number given positive integer~~

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

```
class Factorial
```

{

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

{

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

```
int f = 1, n, i;
```

```
System.out.println("Enter a positive integer: ");
```

```
if (sc.hasNextInt());
```

```
if (n==0)
```

```
f = 1;
```

```
else if (n<0)
```

```
System.out.println("Enter a positive integer!!!");
```

```
else
```

```
{
```

```
for (i=1; i<=n; i++)
```

```
f*=i;
```

```
}
```

```
System.out.println("The factorial of "+n+" is "+f);
```

```
}
```

Enter a positive integer: 5

The factorial of 5 is 120.

Enter a positive integer: -7

Enter a positive integer!!!

Enter a positive integer: 0

The factorial of 0 is 1.

7)

Write a program in JAVA to find the area of a rectangle and verify the same with same various inputs of length and breadth.

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

```
public class Rectangulararea
```

```
{
```

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

```
{
```

```
int l, b;
```

```
l = Integer.parseInt(args[0]);
```

```

b = Integer.parseInt(args[1]);
int area = l * b;
System.out.println("length of rectangle = " + l + " units");
System.out.println("breadth of rectangle = " + b + " units");
System.out.println("Area of rectangle = " + area + " sq units");
}
}

```

C:\> javac RectangleArea.java

C:\> java RectangleArea 20 40

length of rectangle = 20 units

breadth of rectangle = 40 units

area of rectangle = 800 sq units

- 8) Write a program in JAVA to find the sum of digits for a given number using a function / method.

```

import java.util.*;
class Sumofdigits
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println("Enter a number: \t");
        int n = s.nextInt();
        System.out.println("Sum of digits of " + n + " is " + getSum(n));
    }

    static int getSum (int n)
    {
        int sum = 0;
        while (n != 0)
        {
            sum += n % 10;
        }
    }
}

```

CLASSEmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
n1 = 10;  
{  
    return sum;  
}  
}  
}
```

Enter a number: 95914

Sum of digits of 95914 is 28.

- 9) Write a program in JAVA to check whether a given number is palindrome or not using user defined functions

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

```
class Palindrome
```

```
{
```

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

```
{
```

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

```
System.out.println ("Enter any positive integer: ");
```

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

```
int rev = reverse (n);
```

```
System.out.println ("Reversing " + n + " = " + rev);
```

```
if (n == rev)
```

```
System.out.println ("Palindrome Number");
```

```
else
```

```
System.out.println ("Not a palindrome Number");
```

```
}
```

```
static int reverse (int n)
```

```
{
```

```
int rev = 0;
```

```
while (n > 0)
```

```
{
```

```
    rev = rev * 10 + n % 10;
```

{

```
return rev;
```

{

Enter any positive integer: 6954

Reverse of 6954 is 4596

Not a palindrome Number

Enter any positive integer: 696

Reverse of 696 = 696

Palindrome Number.

~~8/12~~

- Addition of snippet to program 5)

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

{

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

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

```
System.out.println("There exists no solution");
```

{

LABORATORY - 3

19<sup>TH</sup> DECEMBER, 2023

TUESDAY

- Q: Develop a program in JAVA 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 the SGPA of a student.

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

```
class Subject
```

```
int subject;
```

```
int credits;
```

```
int grade;
```

```
public Subject()
```

```
{
```

```
    this.credits = 0;
```

```
    this.subject = 0;
```

```
    this.grade = 0;
```

```
}
```

```
class Student
```

```
String name;
```

```
String blrno;
```

```
double SGPA;
```

```
Scanner s;
```

```
Subject[] subj;
```

```
Student()
```

```
{
```

```
    int i;
```

```
    Subject s = new Subject();
```

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

```
        subj[i] = s;
```

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

```
}
```

```
public void getall()
```

```
{
```

```
    System.out.println("Name = " + name);
```

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

```
class Subject {
```

```
    int subjectMarks;
```

```
    int credits;
```

```
    int grade;
```

```
    public Subject ()
```

```
    {
```

```
        this.credits = 0;
```

```
        this.subjectMarks = 0;
```

```
        this.grade = 0;
```

```
}
```

```
}
```

```
class Student {
```

```
    String name;
```

```
    String blank;
```

```
    double SGPA;
```

```
    Scanner s;
```

```
    Subject[] subjects;
```

```
    Student ()
```

```
    {
```

```
        int i;
```

```
        subjects = new Subject[9];
```

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

```
            subjects[i] = new Subject();
```

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

```
}
```

```
    public void getStudentDetails ()
```

```
    {
```

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

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

```
System.out.print("Enter USN: ");
usn = s.nextInt();

}

public void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.print("Enter marks for subject " + (i + 1));
        subjects[i].SubjectMarks = s.nextInt();
        System.out.print("Enter credits for subject " + (i + 1));
        subjects[i].Credits = s.nextInt();
        if (subjects[i].SubjectMarks >= 90)
            subjects[i].grade = 10;
        else if (subjects[i].SubjectMarks >= 80)
            subjects[i].grade = 9;
        else if (subjects[i].SubjectMarks >= 70)
            subjects[i].grade = 8;
        else if (subjects[i].SubjectMarks >= 60)
            subjects[i].grade = 7;
        else if (subjects[i].SubjectMarks >= 50)
            subjects[i].grade = 6;
        else if (subjects[i].SubjectMarks >= 40)
            subjects[i].grade = 5;
        else
            subjects[i].grade = 0;
    }
}

public void computeSGPA() {
    double totalCredits = 0.0;
    double weightedSum = 0.0;

    for (int i = 0; i < 8; i++)
```

totalCredits += subjects[i].credits;

weightedSum += subjects[i].grade \* subject[i].credits;

{

SGPA = weightedSum / totalCredits;

{

{

public class Main {

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);

{

{

Enter Name: NAVANEETH

Enter USN: IBM22CS171

Enter Marks for Subject 1: 100

Enter credits for Subject 1: 4

Enter Marks for Subject 2: 86

Enter credits for Subject 2: 4

Enter Marks for Subject 3: 95

Enter credits for Subject 3: 3

Enter Marks for Subject 4: 78

Enter credits for Subject 4: 3

Enter Marks for Subject 5: 100

Enter credits for Subject 5: 3

Enter Marks for Subject 6: 65

Enter credits for Subject 6: 3

Enter Marks for Subject 7: 100

Enter credits for subject 7: 1

Enter marks for subject 8: 99

Enter marks for subject 9: 1

Student Details:

Name: NAVANEETH

USN: IBM22CS171

SGPA: 9.136363636363637.

19/12/23

Q: Create a class Book which contains four members: name, author, price, num-page. 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 Book
```

```
{
```

```
String name, author;
```

```
int price;
```

```
int num-page;
```

```
Book (String name, String author, int price, int num-page)
```

```
{
```

```
    this.name = name;
```

```
    this.author = author;
```

```
    this.num-page = num-page;
```

```
}
```

```
public String toString()
```

```
{
```

```
String n, a, p, N;
```

```
n = "\n" + "Name of Book:" + name + "\n";
```

```
a = "Author of Book:" + author + "\n";
```

```
p = "Price of Book:" + price + "\n";
```

```
N = "Number of pages:" + num-page + "\n";
```

```
return n + a + p + N;
```

```
{
```

```
{
```

class Main

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of books: ");

int n = sc.nextInt();

Book b[] = new Book[n];

String name, author;

int price, num;

sc.nextLine();

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

{

System.out.println("Enter name of book: ");

name = sc.nextLine();

System.out.println("Enter author's name: ");

author = sc.nextLine();

System.out.println("Enter price: ");

price = sc.nextLine();

System.out.println("Enter number of pages: ");

num = sc.nextInt();

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

{

System.out.println("Details of books: ");

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

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

{

{

{

Output:

Enter number of books: 2

Enter name of book: PRIDE & PREJUDICE & PUNDITRY

Enter author's name: SHASHI THAROOR

Enter Price: 999

Enter number of pages: 872

Enter name of books: THE FAMOUS FIVE

Enter author's name: J.K. ROWLING

Enter Price: 449

Enter number of pages: 352

Details of books:

Name of book: PRIDE & PREJUDICE & PUNDITRY

Author of book: SHASHI THAROOR

Price of book: 999

Number of pages: 872

Name of book: THE FAMOUS FIVE

Author of book: J.K. ROWLING

Price of book: 449

Number of pages: 352

20/10/2024

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, Circle such that each one of the class 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 a, b;

Shape (int x)

{

a = x;

}

Shape (int x, int y)

{

a = x;

b = y;

}

abstract void printArea();

{

class rec extends Shape

{

rec (int a, int b)

{

super(a, b);

{

void printArea()

{

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

{

class tri extends Shape

```
{  
    tri(int a, int b)
```

```
{  
    super(a, b);  
}
```

```
void printarea()
```

```
{  
    System.out.println("The area of the triangle: " + 0.5*a*b);  
}
```

class circ extends Shape

```
{  
    circ(int a)
```

```
{  
    super(a);  
}
```

```
void printarea()
```

```
{  
    System.out.println("The area of the circle: " + 3.14*a*a);  
}
```

class Main

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

```
{  
    Shape r = new rec(4, 5);  
    r.printarea();
```

✓ Shape t = new tri(7, 8);  
t.printarea();

Shape c = new  $\pi r(2)$

C. printArea();

}

Output:

The area of the rectangle: 20

The area of the triangle: 28

The area of the circle: 12.56

Ans 01/01/24

Q: Develop a program in JAVA to create a class Bank that maintains two kinds of accounts 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 chequebook 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 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

class Bank

{

public static void main (String args [])

{

Scanner sc = new Scanner (System.in);

int choice;

Savings savingsAcc = new SavingsAccount ("1001", "Ram", 50000);

Current currentAcc = new CurrentAccount ("1004", "Krishna", 50000);

do

{

System.out.println("1. Deposit to Savings Account");  
System.out.println("2. Deposit to Current Account");  
System.out.println("3. Withdraw from Savings Account");  
System.out.println("4. Withdraw from Current Account");  
System.out.println("5. Display Savings Account Balance");  
System.out.println("6. Display Current Account Balance");  
System.out.println("7. Compute and deposit Interest to  
Savings Account");  
System.out.println("8. Exit");  
System.out.println("Enter your choice");  
choice = sc.nextInt();  
switch(choice){

## Case 1:

System.out.println("Enter amount to deposit in savings");  
double depositAmountSavings = sc.nextDouble();  
savingsAcc.deposit(depositAmountSavings);  
break;

## Case 2:

System.out.println("Enter amount to deposit in current  
Account");  
double depositAmountCurrent = sc.nextDouble();  
currentAcc.deposit(depositAmountCurrent);  
break;

## Case 3:

System.out.println("Enter the amount to withdraw from  
Savings account");  
double withdrawAmountSavings = sc.nextDouble();  
break;

## Case 4:

System.out.println("Enter amount to withdraw from

```
        Current account");  
        double withdrawalAmountCurrent = sc.nextInt();  
        currentAcc.withdraw(withdrawAmountCurrent);  
        break;
```

Case 5:

```
savingsAcc.display();  
break;
```

Case 6:

```
currentAcc.display();  
break;
```

Case 7:

```
savingsAcc.calculateInterest();  
break;
```

case 8:

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

default: "Invalid choice" without two input

```
System.out.println("Invalid choice. Please Enter a valid  
choice = ");
```

} which case 1 = 8)

}

}

class Account

int accNo;

String accHolder;

double balance;

Account (int accNo, String accHolder, double balance)

{

this.accNo = accNo;

this.accHolder = accHolder;

this.balance = balance;

}

~~void deposit (double amount)~~

{

balance += amount;

System.out.println ("Deposit of " + amount + " successful")

{

void display ()

{

System.out.println ("Account Number: " + accNo);

System.out.println ("Account Holder: " + accHolder);

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

{

~~void withdraw (double amount)~~

{

~~if (balance >= amount)~~

{

~~balance -= amount;~~

~~System.out.println ("Withdrawal of \$ " + amount + "~~  
~~" + "Successful.");~~

{

```
    .else
```

```
        System.out.println("Insufficient Balance.");
```

```
}
```

```
}
```

```
class SavingsAccount extends Account
```

```
{
```

```
SavingsAccount (int accNo, String accholder, double balance)
```

```
{
```

```
    super (accNo, accholder, balance);
```

```
}
```

```
void calculateInterest()
```

```
{
```

```
    double interest = balance * 0.05;
```

```
    deposit (interest);
```

```
    System.out.println ("Interest added: £" + interest);
```

```
{
```

```
}
```

```
class CurrentAccount extends Account {
```

```
private double minimumBalance = 1000;
```

```
private double serviceCharge = 50;
```

```
CurrentAccount (int accNo, String accholder, double baln)
```

```
{
```

```
    super (accNo, accholder, baln);
```

```
{
```

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

```
{
```

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

```
{
```

```
        balance -= amount;
```

System.out.println("Withdraw of £ " + amount + " £")  
 }  
 close  
 }

Statement.println("Insufficient Balance - Service charge  
 £ " + serviceCharge + " imposed.");  
 balance -= serviceCharge;

~~MENU~~

~~Enter your choice : 1~~

~~Enter amount to deposit in Current Account: 1000~~

~~Deposit of £1000.0 Successful~~

~~MENU~~

~~Enter your choice : 5~~

~~Account number: 1001~~

~~Account Holder: Ram~~

~~Balance: £6000.0~~

~~MENU~~

~~Enter your choice: 3~~

~~Amount must be minimum from savings: 6001~~

~~Insufficient Balance.~~

~~MENU~~

~~Enter your choice: 2~~

~~Enter amount to deposit in Current Account: 0009~~

~~Insufficient balance. Service charge of £ 80 is imposed~~

~~Bank balance = £ - 41.00~~

*Kim*  
09/01/2024

- 1) Demonstrate various string constructor with proper program in JAVA.

class X

{

public static void main (String args [] )

{

byte ascii = {65, 66, 67, 68, 69, 70};

String s1 = new String (ascii);

System.out.println (s1);

String s2 = new String (ascii, 2, 3);

System.out.println (s2);

}

}

Output:

ABCDEF

- 2) Demonstrate string length, string literal, string concat.

char chars [] = {'a', 'b', 'c'};

String s = new String (chars); String age = "9"

System.out.println (s.length());

System.out.println ("abc".length());

System.out.println ("He is " + age + " years old.");

/ Output: 3

Output: 3

Output: He is 9 years old.

int a = 10;

String s = "He is " + a + " years older than me."

System.out.println (s);

/ Output: He is 9 years older than me.

## LABORATORY - F

3) Demonstrate toString()

class Box

{

    double width, height, depth;

    Box(double w, double h, double d)

{

    width = w;

    height = h;

    depth = d;

{

    public String toString()

{

        return "Dimensions are " + width + " by " + depth + " by "  
            + height + ":";

{

}

class ToStringDemo

{

public static void main (String args [] )

{

Box b = new Box (10, 12, 14);

String s = "Box b: " + b,

System.out.println (b);

System.out.println (s);

{

}

Output:

Dimensions are 10 by 12 by 14

Box b: 10, 12, 14

import java.u

class WrongAge

{

WrongAge

{

super

{

class Father

{

int age

Father (i

if (a,

{

this.a

{

class Son extends

{

int age

Son (in

super

{

if (s >

{

this.age

{

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

```
class WrongAge extends Exception
```

{

```
    WrongAge (String message)
```

{

```
    super (message);
```

}

```
class Father
```

{

```
    int age;
```

```
    Father (int age) throws WrongAge {
```

```
        if (age < 0)
```

{

```
            throws new WrongAge ("Age cannot be negative");
```

}

```
        this.age = age;
```

{

```
class Son extends Father
```

{

```
    int sonAge;
```

```
    Son (int fatherAge, int sonAge) throws WrongAge {
```

```
        super (fatherAge);
```

```
        if (sonAge >= fatherAge)
```

{

```
            throws new WrongAge ("Son's age should be less than Father's  
age");
```

}

```
        this.sonAge = sonAge;
```

{

public class Error

{

    public static void main (String args [] )

{

        Scanner sc = new Scanner (System.in);

        try

{

            System.out.println ("Enter the daddy's age:");  
            int a = sc.nextInt();

            Father father = new Father (a);

            System.out.println ("Enter the sonny's age:");

            int b = sc.nextInt();

            Son son = new Son (a, b);

{

        catch (WrongAge e)

{

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

{

}

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 BMSThread extends Thread

{

public void run()

{

while (true)

}{

}{

System.out.println("BMS College of Engineering");

try

{

Thread.sleep(10000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

f

{

} class CSEThread extends Thread

{

public void main()

{

while (true)

{

System.out.println("CSE");

try

{

Thread.sleep(2000);

{

e.print & break();

{  
    break; }  
    {  
        e.print & break();  
    }  
}

public class ThreadExample

{  
    public static void main (String args [] )

    BMSThread bmsThread = new BMSThread();

    bmsThread.start();

    CSE Thread cseThread = new CSEThread();

    cseThread.start();

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

(9 marks) Demonstrate interprocess communication and deadlock.

(10 marks) Implement "Producer-Consumer" using threads.

class A

```
{ 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("Cont: " + n);
```

```
valueset = true;
```

```
System.out.println("In Producer produce " + n);
```

```
notify();
```

```
return n;
```

```
}
```

~~synchronized void put(int n)~~

```
{ while (valueSet)
```

```
try {
```

```
System.out.println("In Producer waiting " + n);
```

```
wait();
```

```
}
```

catch (InterruptedException e)

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

this.n = n;

Value set = true;

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

System.out.println ("Infinite Consumption");

notify();

}

{

q q;

Producer(q, q)

{

this.q = q;

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

public void run()

{

int i = 0;

while (i < 5)

{

q.put(i++);

{

{

class Consumer implements Runnable

{

Queue q;

Consumer(Q q)

{

this.q = q;

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

{

public void run()

{

int i = 0;

while(i &lt; 5)

{

int r = q.get();

System.out.println("Consumed: " + r);

i++;

}

}

class PC {

public static void main(String args[])

{

Q q = new Q();

new Producer(q);

new Consumer(q);

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

{

{

## LABORATORY - 9

Output:

Press Control-C to stop!

Put: 0

Intimate Consumer

Producer Waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer

Producer Waiting

Consumed: 0

Got: 1

Intimate Producer

Consumed: 1

Put: 2

Intimate Consumer

Producer Waiting

Got: 2

Intimate Producer

Consumed: 2

Put: 3

Intimate Consumer

Producer Waiting

Got: 3

Intimate Producer

Consumed: 3

Put: 4

Intimate Consumer

Got: 4

Intimate Producer

Consumed: 4.

Write a program  
divisions. The  
and Num2. Th  
the result file  
Num2 were no  
NumberFormat  
would show an  
messagedialogue

```
import javax.s
import java.awt
import java.awt
class swingdem
SwingDemo {
}
```

```
// create JF
Frame f
```

```
ffrm.setS
frm.setB
t
```

```
// to termi
ffm.setAf
```

```
// textlabel
JLabel jlab =
// add entit
JTextField a
JTextField b
f
```

```
// calculation
JButton butc
```

```
// labels
JLabel err = ne
```

LABORATORY - 9

20<sup>TH</sup> FEBRUARY, 2024

TUESDAY

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 SwingDemo{  
    SwingDemo(){  
        // create jframe container  
        JFrame jfrm = new JFrame("Divider App");  
        jfrm.setSize(275, 600);  
        jfrm.setLayout(new flowLayout());  
        // to terminate on close  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        // textlabel  
        JLabel jLab = new JLabel("Enter the divisor and dividend.");  
        // add textfield for both numbers  
        JTextField aJtf = new JTextField(8);  
        JTextField bJtf = new JTextField(8);  
  
        // calculation  
        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(cerr);
```

```
jfrm.add(jlab);
```

```
jfrm.add(ajtf);
```

```
jfrm.add(bjtf);
```

```
jfrm.add(button);
```

```
jfrm.add(alab);
```

```
jfrm.add(blab);
```

```
jfrm.add(anslab);
```

```
ActionListener I = new ActionListener()
```

```
{
```

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

```
{
```

```
System.out.println("Action event from 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)

```

{
    a.lab.setText("0");
    blab.setText("0");
    anslab.setText("0");
    err.setText("Enter Only Integer");
}

```

catch (ArithmaticException e)

```

{
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be Non zero!");
}

```

// display frame  
jform.setVisible(true);

public static void main (String args [])

```

{
    // create frame on event dispatching thread
    SwingUtilities.invokeLater (new Runnable ());
}

```

~~public void run ()  
new Swing (err);~~

}

Output:

Enter the divisor and dividend:

39	13
----	----

[Calculate]

$$A = 39 \quad B = 13 \quad \text{Ans} = 3$$

Enter only Divisor!

Enter the divisor and dividend:

39
----

13
----

[Calculate]

B should be Non zero

Enter the divisor and ans are direct

39
----

0
---

[Calculate]

Tip: 02.24  
go.02.24

## Lab Programs

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 quad {
    public static void main(String args[]) {
        int a, b, c;
        double r1, r2, d;
        Scanner s = new Scanner(System.in);
        System.out.println("Nehal A K\\n1BM22CS176");
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            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);
    }

}

```

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 grade;
}

```

```

class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subject;
    Scanner s;
}

```

```

Student() {
    int i;
    subject = new Subject[9];
    for (i = 0; i < 9; i++)

```

```
    subject[i] = new Subject();
    s = new Scanner(System.in);
}

void getStudentDetails() {
    System.out.println("Enter student name");
    name = s.next();
    System.out.println("Enter student usn");
    usn = s.next();
}
```

```
void getMarks() {
    int i;
    for (i = 0; i < 9; i++) {
        System.out.println("Enter marks for subject " + (i + 1));
        subject[i].subjectMarks = s.nextInt();
        System.out.println("Enter credits for subject " + (i + 1));
        subject[i].credits = s.nextInt();

        if (subject[i].subjectMarks >= 90) {
            subject[i].grade = 10;
        } else if (subject[i].subjectMarks >= 80 && subject[i].subjectMarks
< 90) {
            subject[i].grade = 9;
```

```
        } else if (subject[i].subjectMarks >= 70 && subject[i].subjectMarks  
        < 80) {  
  
            subject[i].grade = 8;  
  
        } else if (subject[i].subjectMarks >= 60 && subject[i].subjectMarks  
        < 70) {  
  
            subject[i].grade = 7;  
  
        } else if (subject[i].subjectMarks >= 50 && subject[i].subjectMarks  
        < 60) {  
  
            subject[i].grade = 6;  
  
        } else if (subject[i].subjectMarks >= 40 && subject[i].subjectMarks  
        < 50) {  
  
            subject[i].grade = 5;  
  
        } else {  
  
            System.out.println("Failed");  
  
            System.exit(0);  
  
        }  
    }  
}
```

```
void computeSGPA() {  
  
    int totalCredits = 0;  
  
    int creditsGained = 0;  
  
    int i;  
  
  
    for (i = 0; i < 9; i++) {  
  
        totalCredits += subject[i].credits;
```

```
    creditsGained += subject[i].credits * subject[i].grade;

}

SGPA = (double) creditsGained / totalCredits;

}

void displayResult() {
    System.out.println("Name = " + name);
    System.out.println("Usn = " + usn);
    System.out.println("SGPA = " + SGPA);
}

}

public class Main {
    public static void main(String args[]) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResult();
    }
}
```

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 Book {  
    String name;  
    String author;  
    int price;  
    int numPages;  
  
    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 bookDetails = "Book name: " + this.name + "\n" +  
            "Author name: " + this.author + "\n" +  
            "Price: " + this.price + "\n" +  
            "Number of pages: " + this.numPages + "\n";  
        return bookDetails;  
    }  
}
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.println("Enter the number of books: ");
        int n = s.nextInt();

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

```

System.out.println("\nDetails of the books:");

for (int i = 0; i < n; i++) {
    System.out.println("Book " + (i + 1) + ":\n" + books[i].toString());
}

}

```

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.*;

abstract class AbsArea {
    int a, b;

    AbsArea(int x) {
        a = x;
    }

    AbsArea(int x, int y) {
        a = x;
        b = y;
    }

    abstract void area();
}

class rec extends AbsArea {

```

```
rec(int a, int b) {
    super(a, b);
}

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

class tri extends AbsArea {
    tri(int a, int b) {
        super(a, b);
    }

    void area() {
        System.out.println("The area of the triangle is: " + (a * b) / 2);
    }
}

class cir extends AbsArea {
    cir(int a) {
        super(a);
    }

    void area() {
        System.out.println("The area of the circle is: " + 3.14 * a * a);
    }
}

class Main {
    public static void main(String args[]) {
        System.out.println("This is done by Nehal AK\n1BM22CS176");
        int x, y;
        Scanner n = new Scanner(System.in);

        // Input for rectangle dimensions (x and y) with validation
        System.out.println("Give input for rectangle");
    }
}
```

```

x = n.nextInt();
y = n.nextInt();
if (x < 0 || y < 0) {
    System.out.println("Invalid input for rectangle. Please enter
positive values.");
    // You might want to handle this situation differently, such as
    asking the user to enter values again.
    System.exit(1); // Exiting with status code 1 (indicating abnormal
exit)
}

AbsArea r = new rec(x, y);
r.area();

// Input for triangle dimensions (x and y) with validation
System.out.println("Give input for triangle");
x = n.nextInt();
y = n.nextInt();
if (x < 0 || y < 0) {
    System.out.println("Invalid input for triangle. Please enter positive
values.");
    System.exit(1);
}

AbsArea t = new tri(x, y);
t.area();

// Input for circle radius (x) with validation
System.out.println("Give input for circle");
x = n.nextInt();
if (x < 0) {
    System.out.println("Invalid input for circle. Please enter a positive
value.");
    System.exit(1);
}

AbsArea c = new cir(x);

```

```
    c.area();
}
}
```

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.

```
import java.util.Scanner;

class Account {

    String CustomerName;
    double AccNo, Balance;

    Account(String CustomerName, double AccNo, double Balance) {
        this.CustomerName = CustomerName;
        this.AccNo = AccNo;
        this.Balance = Balance;
    }

    public void Deposit(double Amount) {
        Balance += Amount;
        System.out.println("DEPOSIT SUCCESSFUL");
        DisplayBalance();
    }
}
```

```
void DisplayBalance() {  
    System.out.println("BALANCE:" + Balance);  
}  
}  
  
class CurrentAccount extends Account {  
  
    double MinBalance = 500.0;  
  
    double Charges = 10.0;  
  
    CurrentAccount(String CustomerName, double AccNo, double Balance) {  
        super(CustomerName, AccNo, Balance);  
    }  
  
    void Withdraw(double Amount) {  
        if (Balance >= Amount) {  
            Balance -= Amount;  
            System.out.println(Amount + " withdrawn successfully");  
            DisplayBalance();  
        } else {  
            System.out.println("insufficient Balance");  
        }  
    }  
  
    void UpdateBalance() {  
        if (Balance <= MinBalance) {  
            Balance -= Charges;  
        }  
    }  
}
```

```
        System.out.println("service charge applied for maintaining low
balance");

        DisplayBalance();

    }

}

class SavingsAccount extends Account {

    SavingsAccount(String CustomerName, double AccNo, double Balance) {
        super(CustomerName, AccNo, Balance);

    }

    double interest = 0.05;

    void UpdateBalance() {
        Balance = Balance + (interest * Balance);
        DisplayBalance();
    }

    void Withdraw(double Amount) {
        if (Balance >= Amount) {
            Balance -= Amount;
            DisplayBalance();
        } else {
            System.out.println("insufficient Balance");
        }
    }
}
```

```
}
```

```
class Bank {  
    public static void main(String args[]) {  
        String CustomerName;  
        double AccNo, Balance;  
        double amt, amt1;  
        Scanner in = new Scanner(System.in);  
  
        System.out.println("enter name:");  
        CustomerName = in.next();  
  
        System.out.println("enter AccNo:");  
        AccNo = in.nextDouble();  
        System.out.println("enter Balance:");  
        Balance = in.nextDouble();  
        System.out.println("enter amount to deposit");  
        amt = in.nextDouble();  
        System.out.println("enter amount to withdraw");  
        amt1 = in.nextDouble();  
  
        CurrentAccount c = new CurrentAccount(CustomerName, AccNo,  
        Balance);  
        c.Deposit(amt);  
        c.Withdraw(amt1);  
        c.UpdateBalance();
```

```

System.out.println(" ");

SavingsAccount s = new SavingsAccount(CustomerName, AccNo,
Balance);

s.Deposit(amt);

s.Withdraw(amt1);

s.UpdateBalance();

}

}

```

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

    public Student(String name,String usn,int sem)
    {
        this.name=name;
        this.usn=usn;
        this.sem=sem;
    }
}

```

```
package CIE;

public class Internals extends CIE.Student
{
    public int [] InternalMarks;

    public Internals(String name , String usn , int sem , int []InternalMarks)
    {
        super(name , usn , sem);
        this.InternalMarks=InternalMarks;
    }
}

package SEE;
import CIE.Student;

public class Externals extends Student
{
    public int [] SeeMarks;

    public Externals(String name , String usn , int sem , int []SeeMarks)
    {
        super(name , usn , sem);
        this.SeeMarks=SeeMarks;
    }
}

import CIE.Student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;

public class FinalMarks
{
    public static void main(String [] args)
    {
        Scanner s1=new Scanner(System.in);

        System.out.println("Enter the number of Students");
        int n=s1.nextInt();
```

```

String []names=new String[n];
String []usn=new String[n];
int []sem = new int[n];
int [][] InternalMarks = new int[n][5];
int [][] SeeMarks = new int[n][5];

for(int i=0 ; i<n; i++)
{
    System.out.println("Enter details for Student" + (i+1) + ":");

    System.out.println("Name:");
    names[i]=s1.next();
    System.out.println("USN:");
    usn[i]=s1.next();
    System.out.println("SEM:");
    sem[i]=s1.nextInt();

    System.out.println("Enter Internal marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        System.out.println("Course"+(j+1)+":");
        InternalMarks[i][j]=s1.nextInt();
    }
    System.out.println("Enter External marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        System.out.println("Course"+(j+1)+":");
        SeeMarks[i][j]=s1.nextInt();
    }
}

int [][]FinalMarks = new int[n][5];
for(int i=0 ; i<n ; i++)
{
    Internals I1 = new Internals(names[i] , usn[i] , sem[i] , InternalMarks[i]);
    Externals E1 = new Externals(names[i] , usn[i] , sem[i] , SeeMarks[i]);

    for(int j=0; j<5 ;j++)
    {
        FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];
    }
    System.out.println("Finals Marks for " + n+ "Students in 5 courses:");
    for(i=0 ;i<n ;i++)
    {
        System.out.println(names[i] +":");
    }
    for(int j=0; j<5;j++)
    {
        System.out.println(FinalMarks[i][j] + ":");

    }
}

```

```
    }  
    System.out.println();  
}  
s1.close();  
}  
}  
}
```

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 message) {
        super(message);
    }
}

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

class Father extends InputScanner {
    int fatherAge;

    Father() throws WrongAge {
        System.out.println("Enter Father's age");
        fatherAge = sc.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}
```

```
}

void display() {
    System.out.println("Father's age is " + fatherAge);
}

}

class Son extends Father {
    int sonAge;

    Son() throws WrongAge {
        System.out.println("Enter son's age");
        sonAge = 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 display() {
        System.out.println("Son's age is " + sonAge);
    }
}

public class ExceptionHandling {
    public static void Main(String args[]) {
        try{
            Son son= new Son();
            son.display();
        }
        catch(WrongAge e){
            System.out.println("Exception "+ e.getMessage());
        }
    }
}
```

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.

```
import java.io.*;
class B extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("BMS"); Thread.sleep(10000);
}
} catch (InterruptedException e){
System.out.println(e);
}
}
}
class C extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("CSE"); Thread.sleep(2000);
}
} catch (InterruptedException e){
System.out.println(e);
}
}
}
class ThreadMain{
public static void main(String args[]){
B b = new B();
C c = new C();
b.start(); c.start();
}
}
```

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 SwingDemo {
    SwingDemo() {
        // 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 dividend:");
        // 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
        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 (ArithmaticException 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 SwingDemo();
    }
});
}
```

10) Demonstrate Inter process Communication and deadlock.

IPC

ProCon.java

```
class Q {  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        while (!valueSet)  
            try {  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        notify();  
        return n;  
    }  
    synchronized void put(int n) {  
        while (valueSet)  
            try {  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + 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 < 5) {  
            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 < 5) {
int r = q.get();
i++;
}
}
}

class ProCon {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
}

```

## **Deadlock.java**

```

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 this thread.
    System.out.println("Back in main thread");
}
public void run() {
    b.bar(a); // get lock on b in other thread.
    System.out.println("Back in other thread");
}
public static void main(String args[]) {
    new Deadlock();
}
}
```