

# **B.M.S COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



## **LAB REPORT**

**23CS3PCOOJ**

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

**Preeti T Korishettar**

**1BM22CS208**

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

Practice programs. 5/12/23

- Java is a high level programming language.
1. Write a c program to search for a key element using a linear search.
  2. Write a c program to sort the number using bubble sort.

```
#include <stdio.h>
```

```
void main()
```

```
{ int num;
```

```
int a[10];
```

```
scanf("%d", &a[i]);
```

```
printf("Enter the key");
```

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

```
{ scanf("%d", &a[i]);
```

```
printf("Enter the key element");
```

```
scanf("%d", &num);
```

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

```
{ if (num == a[i])
```

```
printf("The key element is %d", a[i]);
```

## Bubble sort

```
#include <stdio.h>
```

```
void main()
```

```
{
```

```
    int num, i, j;
```

```
    int a[10];
```

```
    int temp, n;
```

```
    printf("Enter the number of array elements");
```

```
    scanf("%d", &n);
```

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

```
{
```

```
    scanf("%d", &a[i]);
```

```
    for (i = 0; i < n - 1; i++)
```

```
{
```

```
        for (j = 0; j < n - i - 1; j++)
```

```
            if (a[j] > a[j + 1])
```

```
                temp = a[j];
```

```
                a[j] = a[j + 1];
```

```
                a[j + 1] = temp;
```

```
}
```

```
}
```

```
printf("The sorted array is\n");
```

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

```
{
```

```
    printf("%d", a[i]);
```

```
}
```

Output :

Enter the number of array elements 6

5

4

3

2

1

0

The sorted array is

0 1 2 3 4 5

Program to print "Hello Word"

Lab program

Quadratic class

Practice program

- 1) Scanner
- 2) Parse int
- 3) Fact
- 4) Palindrom
- 5) Sum of 5 digits
- 6)  $nCn$
- 7) Statis.

11 Program to print helloworld:

```
import java.util.*;
class demo1
{
    public static void main (String args[])
    {
        System.out.println ("Hello World");
    }
}
```

output : Hello World

11 Program to find area of  $\square$  using  
parse int

```
class rec_area
{
```

```
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 = " + l);

System.out.println ("Breadth = " + b);

System.out.println ("area = " + area);

```
}
```

output:

length = 10

Breadth = 2

area = 20.

II Program for the usage of Scanner.

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

```
class scanner
```

```
{
```

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

```
int a ; float b ; String c ;
```

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

```
System.out.println ("Enter a string")
```

```
c = in.nextLine();
```

```
System.out.println ("you entered a string "+c);
```

```
System.out.println ("Enter an integer");
```

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

```
System.out.println ("you entered an integer "+a);
```

```
System.out.println ("Enter a float value");
```

```
b = in.nextFloat();
```

```
System.out.println ("you entered a float "+b);
```

```
}
```

```
}
```

Output :

Enter a string

A

you entered a string A

enter an integer

2 you entered an integer 2

enter the float

0.2

you entered the float 0.2

" Program for the usage of array

in class array

{

public static void main (String args[])

{

int m.days [] = {31, 28, 31, 30, 31, 30, 31, 30, 31, 30,

System.out.println ("March has " + m.days [0] + " days");

}

}

Output:

March has 31 days.

" Program for factorial.

import java.util.Scanner;

class factorial

{

public static void main (String args[])

{

int fac = 1;

Scanner sc = new Scanner (System.in);

int i, n;

sc.n = sc.nextInt();

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

```

    {
        fac = fac * i;
    }
    System.out.println ("The factorial of " + n + " is " + fac);
}

```

output

The factorial of 5 is 120

### LAB -1

11 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 stating that there are no real solutions.

import java.util.Scanner;

class Quadratic

{

```

        int a, b, c;
        double r1, r2, d;
        void gfeld()
    }
```

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 & distinct");
        System.out.println ("Root1 = " + r1 + " Root2
                           = " + r2);
    }
    else if (d < 0)
    {
        System.out.println ("Roots are imaginary");
        r1 = (-b) / (2*a);
    }
}

```

```

n2 = Math.sqrt(-d)/(2*a);
System.out.println("Root1 = "+n1+i"+n2);
System.out.println("Root1 = "+n1+"-i"+n2);
}

}

class QuadraticMain {
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.gtd();
        q.compute();
    }
}

```

Output: Enter the coefficients of a,b,c;

1

-2

1

Roots are real and equal

Root1 = Root2 = 1.0

$$(A + \bar{A}B) = A + B$$

$$\bar{A} + A\bar{B}$$

19/12/23

// Develop Java program to create a class Student with members USN, name, an array credits and an array marks. Include method to accept and display details and calculate SGPA.

$$> \text{SGPA} = \frac{\sum [(\text{Course Credits})(\text{Grade Points})]}{\sum (\text{Course credits})}$$

$$> \text{CGPA} = \text{" "}$$

// Using 2D array, read & print the value of array

19/12/23

LAB - 2.

Bafna Gold

Date:

Page:

import java.util.Scanner;

class Subject

{

int Marks;

int Credits;

int Grade;

}

class Student

{

Subject[] Subjects;

String name;

String USN;

double SGPA;

Scanner s;

Student ()

{

int i;

Subjects = new Subject[10];

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

{

Subjects[i] = new Subject();

}

s = new Scanner (System.in);

{

void getStudentDetails()

{

System.out.println ("Enter the name of  
the student");

name = s.nextLine();

System.out.println ("Enter the USN of  
the student");

```
    usn = s.nextInt();  
}
```

```
void getMarks()  
{
```

```
    int i;
```

```
    System.out.println("Enter the marks");  
    for (i=0; i<8; i++)  
{
```

```
        Subjects[i].Marks = s.nextInt();
```

```
        if (Subjects[i].Marks > 100)
```

```
            Subjects[i].Grade = 10;
```

```
        else if (Subjects[i].Marks >= 90 &&
```

```
                  Subjects[i].Marks <= 100)
```

```
            Subjects[i].Grade = 9;
```

```
        else if (Subjects[i].Marks >= 80 &&
```

```
                  Subjects[i].Marks < 90)
```

```
            Subjects[i].Grade = 8;
```

```
        else if (Subjects[i].Marks >= 70 &&
```

```
                  Subjects[i].Marks < 80)
```

```
            Subjects[i].Grade = 7;
```

```
        else if (Subjects[i].Marks >= 60 &&
```

```
                  Subjects[i].Marks < 70)
```

```
            Subjects[i].Grade = 6;
```

```
        else if (Subjects[i].Marks >= 50 &&
```

```
                  Subjects[i].Marks < 60)
```

```
            Subjects[i].Grade = 5;
```

else if (Subjects[i].Marks >= 40 && Subjects[i].Marks < 50)

    Subjects[i].Grade = 5;

else

    System.out.println ("Fail");

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

{

    System.out.println ("Enter the credits");

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

{

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

}

void calculateSGPA()

{

    int i;

    int n, SGPA;

    int d = 20;

    int n\_sum = 0;

    for (i=0; i<8; i++)

{

        n = (Subjects[i].Grade) \* (Subjects[i].Credits);

{

        n\_sum = n + n\_sum;

    SGPA = (n\_sum) / (d);

    System.out.println ("SGPA = " + SGPA);

{

```
class Main
```

```
{
```

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

```
    {
```

```
        Student s1 = new Student();
```

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

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

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

```
}
```

```
}
```

output:

Enter the <sup>name</sup> of the student

Preeti

Enter the usn of the student

208

Enter the marks

92

94

89

90

94

86

98

96

enter the credits

4

4

3

3

3

1

SGPA = 9.8

8  
10/12/23

26/12/23 LAB - 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;

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 = "number of pages :" + this.  
    numpages + "\n";
```

```
    return name + author + price + numpages;
```

```
}
```

```
class Parti
```

```
{
```

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

```
{
```

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

```
int n, i;
```

```
String name;
```

```
String author;
```

```
int price;
```

```
int numpages;
```

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

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

```
books b[];
```

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

```
System.out.println ("Enter the name,  
author, price and number of pages  
of the book");
```

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

```
{  
    name = s.nextLine();
```

```
    author = s.nextLine();
```

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

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

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

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

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

```
}
```

```
}.
```

Output:

Enter the number of books

2

Enter the name, author, price and number of pages  
of the book

ncert hcverma 100 1000

Kalinali Kanakadasa 50 70.

name book name : ncert

author name : hcverma

price = 100

numpage of pages = 1000

book name : Kalinali

author name : Kanakadasa

price = 50

number of pages = 70

24/11/23

- 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;
```

```
abstract class shape
```

```
{
```

```
    int a,b;
```

```
    abstract void printarea();
```

```
    abstract void input();
```

```
}
```

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

```
class rectangle extends shape
```

```
{
```

```
    void input()
```

```
{
```

```
    System.out.println ("Enter the length  
and breadth");
```

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

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

```
}
```

```
    void printarea()
```

```
{
```

```
    System.out.println ("Area of triangle  
rectangle is :" +(a*b)+" sq units");
```

}

3.

class triangle extends shape

{

void input()

System.out.println ("Enter the base and  
height :");

a = s.nextInt();

b = s.nextInt();

}

void printarea()

System.out.println ("Area of triangle  
is: " + ((a \* b) / 2) + " sq units");

}

}

class tri circle extends shape

{

void input()

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

a = s.nextInt();

}

void printarea()

System.out.println ("Area of circle is"  
+ ((3.14 \* a \* a)) + " sq. units");

}

}

class area

{

public static void main (String args[])

{rectangle r = new rectangle();

r.input();

r.printarea();

triangle t = new triangle();

t.input();

t.printarea();

circle c = new circle();

c.input();

c.printarea();

}

Output:

Enter the length and breadth.

4

5

The area of rectangle is : 20 square units

Enter the base and height

4

5

The area of triangle is : 10 square units

Enter the radius

2

The area of circle is : 12.56 square units.

import java.util.Scanner

class Account :

{

String custum\_name;

String acc\_no;

String acc\_type;

Scanner s = new Scanner (System.in);

void get()

{

System.out.println ("Enter customer name:");

custum\_name = s.next();

System.out.println ("Enter account number:");

acc\_no = s.next();

}

void get1()

{

System.out.println ("Enter the type of the account");

1 : Savings \n 2 : Current account \n ");

acc\_type = s.nextInt();

}

class Current extends Account

{

double balance = 0;

double deposit;

void cheque()

{

System.out.println ("Enter the amount to be issued in cheque \n ");

s.nextInt();

void deposit()

{  
System.out.println("Enter the amount to  
be deposited\n");

deposit = s.nextInt();

System.out.println("The amount is +  
deposit + " is deposited successfully");

}

void balance() ch-bal()

{

{ if (balance <= 500)

System.out.println("Balance is less  
than minimum balance.\n");

balance = balance - 5;

System.out.println("New Balance  
after applying service charges is  
+ balance);

}

else

{

System.out.println("The balance  
amount is " + balance);

}

class Sav-acct extends Account

{

double balance = 0;

double deposit, withdraw;

Scanner s1 = new Scanner(System.in);

void deposit()

{  
System.out.println ("Enter the amount to be deposited");

deposit = sc.nextInt();

System.out.println ("The amount is deposited successfully"); balance = balance + deposit;  
}

void withdraw()

{  
System.out.println ("Enter the amount to be withdrawn\n");

withdraw = sc.nextInt();

System.out.println ("

balance = balance - withdraw;

System.out.println ("The balance amount is " + balance);

}

void compInterest()

System.out.println ("

balance = balance + balance \* (0.06)

System.out.println ("The balance after applying interest is " + balance);

}

class Bank

{  
public static void main (String args [])  
{  
Scanner sc = new Scanner ();  
int ch;  
Scu.out a = new save.out();

(cur-act b = new cur-act();

b.get()

b.get(1)

do

{

System.out.println ("Enter the choice  
do \n 1:

{ if (acc-type == 1)

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

System.out.println (" Enter the choice  
1: Deposit \n 2: withdrawal \n 3: Compute

Interest);

ch = sc.nextInt();

switch(ch)

{

case 1: a.deposit();

break;

case 2: a.withdrawal();

break;

case 3: a.comp\_interest();

break;

}

else

System.out.println ("---Menu  
--- \n");

System.out.println ("Enter the  
choice \n 1: Deposit \n 2: cheque \n  
3: check balance \n");

```
ch = sc.nextInt();  
  
switch(ch){  
    case 1: b.deposit();  
    break;  
    case 2: b.cheque();  
    break;  
    case 3: b.ch_bal();  
    break;  
}  
}.while(ch!=4);  
}.while(ch!=3);  
}  
}
```

### Output:

Enter customer name:

Preeti

Enter account number:

123456

Enter the account type

1. Savings

2. Current account

1

--- Main menu ---

1 Dep Enter the choice

1 Deposit

2 Withdrawal

3 Compound Interest

1

Enter the amount to be deposited : 2000

Amount 2000 is deposited successfully.

Enter the account type.

1. Savings

2. Current account

2. Current

--- Main menu ---

1. Deposit

2. cheque issue

3. check balance

1.

Enter the amount to be deposited

2000

Amount 2000 is deposited successfully.

--- Main menu ---

1. Deposit

2. cheque issue

3. check balance

2.

The balance amount is 2000.

--- Main menu ---

1. Deposit

2. cheque issue

3. check balance

4.

K / 200  
OK

# Demonstration of String functions

Bafna Gold

16/01/24

KABZS

Date:

Page:

S = javaProgram

1. Demonstrate various string construction with proper java programs.

Input S = javaProgram

S1 = JavaP

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

String length of S = 11 // string length

Length of JAVA is = 4 // string literal

String concatenate : He is beautiful // string concat.

3. Demonstrate toString()

Dimensions are 10.0 by 14.0 by 12.0

Box b: Dimensions are 10.0 by 14.0 by 12.0

4. Using getchars(), extract Beautiful from "Welcome to Briske college". "He is Beautiful"

Beautiful.

5. Demonstrate getbytes(), tocharArray() with proper java programs.

For "He is beautiful"

72

101

32

105

115

32

98

101

97

117

116

105

102

117

108

getCharAt(0)

output He is beautiful.

- 6 check the following output and write the java programs using string function.

output, Bmse = Bmse true

Bmse = College false

Bmse = BMSC false

Bmse equals Ignore Case BMSC true

7. Using regionMatches() find the substring "Bmse college" from the string "Welcome to Bmse College of Engineering", if matches display substring is matched otherwise display not matched.

output: substring is matched.

8. Demonstrate startWith() to give output true and false.

true

( 'FOOLS' starts with 'F' )

false

( 'HEY' starts with 'Y' )

9. Demonstrate `endswith()` to give output true and false.

true ( 'FOOLS' ends with 'S' )

false ( 'HEY' ends with 'H' )

10. Demonstrate a java program to show the output for `equals()` versus `=`.

Output true ( Bottle equals Bottle )

false ( Bottle == Bottle ) different instance

11. Write a java program to show the output for `equals()` perform sorting for alphabets using `compareTo()`.

"van", "watch", "ball", "cat", "xmas", "yatch",  
 "zec", "apple", "ice", "jug", "kite", "lift", "man";  
 "net", "orange", "dog", "ent", "face", "gum", "hen",  
 "parrot", "queen", "ring", "star", "tree", "umbrella",

Output apple

ball

cat

dog

ent

face

gum

hen

ice

jig  
kite  
lift  
man  
net  
orange  
parrot  
queen  
ring  
star  
tree  
umbrella  
van  
watch  
xmas  
yatch  
zee.

12 Sorting from 10-1 using compare.

output 1 2 3 4 5 6 7 8 9 10

13. on WAP to demonstrate `substring()`, `indexof()`,  
+, for replacing "was" to "is".  
"Thwas was a test, Thwas was, too".

output, This was a test. Thwas was, too  
This is a test. Thwas was too  
This is a test. This was too:  
This is a test. This is too.

14. WAP to concat() s1 = "Hello" & s2 = "world".

Output hello world.

15 WAP replace(), "College" with "Commge":

Output: BMS Commge.

16. WAP demonstrate trim() for "Hello Friends".

Output "Hello Friends".

8  
16/10/2021

```

package cie;
import java.util.Scanner;

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);

```

```

System.out.println("Enter the usn of student");
usn = s.next();

```

```

System.out.println("Enter the name of the student");
name = s.nextLine();

```

```

System.out.println("Enter the sem in
which student is studying");
sem = s.nextInt();
}

```

```

public void displaystudentdetails()
{
}

```

```

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

```

```

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

```

```

System.out.println("Sem: " + sem);
}

```

Internals.java

```

package cie;

```

```
import java.util.Scanner;  
public class internals extends student  
{  
    protected int marks[] = new int[5];  
  
    public void input_marks()  
    {  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter the marks");  
        for (int i=0 ; i<5 ; i++)  
        {  
            marks [i] = s.nextInt();  
        }  
    }  
}.
```

internals.java.

package see;

```
import cie.internals;  
import java.util.Scanner;
```

```
public class internals extends internals
```

protected int marks[];

protected int finalmarks[];

public internals()

marks = new int [5];

finalmarks = new int [5];

}

```
public void input_marks()
```

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

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

```
{  
    System.out.println ("Subject "+(i+1)+  
        " marks");
```

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

```
}
```

```
{  
    }  
}
```

```
public void calculate_final_marks()
```

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

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

```
{  
    public void display_final_marks()
```

```
{  
    display_studentDetails();
```

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

```
{  
    System.out.println ("Subject "+(i+1)+  
        " finalmarks[i]);
```

```
{  
    }  
}
```

```
main.java
```

```
import java.util.*;  
class main
```

```
{
```

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

```
{ int numofstudents = 1;
```

```
externals finalmarks[] = new externals[  
    numofstudents];
```

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

```
    finalmarks[i] = new externals();
```

```
    finalmarks[i].inputstudentdetails();
```

```
    System.out.println("Enter cie marks");
```

```
    finalmarks[i].inputciemarks();
```

```
}
```

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

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

```
    finalmarks[i].calculatefinalmarks();
```

```
    finalmarks[i].displayfinalmarks();
```

```
}
```

```
}
```

Output:

Enter no

Enter the VSN of student.

1BN22CS208

Enter the name of the student

Preeti T.B

Enter the sem in which student is studying

Enter cia marks  
Enter the marks

marks 0 = 10

marks 1 = 10

marks 2 = 10

marks 3 = 10

marks 4 = 10.

Enter ssc marks.

subject 1 ss marks 10

subject 2 ss marks 10

subject 3 ss marks 10

subject 4 ss marks 10

subject 5 ss marks 10.

## ② Displaying data

finalmarks[0] = 15

finalmarks[1] = 15

finalmarks[2] = 15

finalmarks[3] = 15

finalmarks[4] = 15

VSN : 1BN422CS208

Name : Prachi

Sem : 2

Subject 1 15

Subject 2 15

Subject 3 15

Subject 4 15

Subject 5 15

15 15 15  
15 15 15

WAP to implement exception handling using father, son inheritance tree.

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

```
class Wrongage extends Exception
```

```
{
```

```
    Wrongage (String s)
```

```
{
```

```
    super(s);
```

```
}
```

```
class Input
```

```
{
```

```
    int f_age;
```

```
    int s_age;
```

```
{
```

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

```
class Father extends Input
```

```
{
```

```
    father() throws Wrongage
```

```
{
```

```
    System.out.println ("Enter father's age:");
```

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

```
    if (f_age < 0)
```

```
        throw new Wrongage ("Age cannot be negative");
```

```
    }.
```

```
    void display()
```

```
{    System.out.println ("father's age is " +
```

```
f_age)
```

```
} }
```

class son extends father

{  
    son() throws Wrongage

{  
    System.out.println("Enter son's age");

s.age = sc.nextInt();

} if (s.age >= f.age)

{  
    throw new Wrongage ("Son's age  
    cannot be greater than father's");

} else if (s.age < 0)

    throws new Wrongage ("Age cannot  
    be negative");

}

    void display()

{

    System.out.println("Son's age is: " + s.age);

class Main

{

    public static void main (String args[])

    try

        Son s = new Son();

        s.display();

    }

catch (Wrongage e)

}. System.out.println ("Error: " + e);

{

}

### OUTPUT

Enter father's age:

-10

Error - Wrong : Age cannot be negative.

30 | 01/24

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

```
class threadone implements Runnable
```

```
{
```

```
String a;
```

```
threadone (String b)
```

```
{
```

```
a = b;
```

```
Thread t = new Thread (this,a);
```

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

```
}
```

```
public void run()
```

```
{
```

```
try
```

```
{
```

```
while (true)
```

```
{
```

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

```
Thread.sleep (10000);
```

```
}
```

```
.
```

```
catch (InterruptedException e)
```

```
{
```

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

```
System.out.println (a + " exiting");
```

```
.
```

class ThreadTwo implements Runnable

{

String a;

ThreadTwo (String b)

{

a = b;

ThreadTwo (String b)

{

a = b;

Thread t = new Thread (this, a);

t.start();

}.

public void run()

{

try

{

while (true)

{

System.out.println(a);

Thread.sleep(2000);

}

}.

catch (InterruptedException e)

{

System.out.println(a + " interrupted");

}

System.out.println(a + " exiting.");

}

}.

}

```
class mainthread  
{  
    public static void main (String args[]){  
        {  
            new Threadone ("BMSCE");  
            new Threadtwo ("CSE");  
        }  
    }.  
}
```

OUTPUT.

BMSCE

CSE // after 2sec. ~~Line will be~~

CSE

CSE

CSE

CSE

BMSCE // after 10s

CSE

CSE

CSE

CSE

BMSCE

class Q

{

int n;

synchronized int cl()

System.out.println("Q of :" + n);

return n;

}.

synchronized void put(int n)

this.n = n;

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

}.

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;

        newthread(this, "Consumer").start();

    }

    public void run()

    {

        int i = 0;

        while (i < 15)

        {

            int n = q.get();

            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");

}

Output:

Put : 1

put : 2

put : 6

Get : 1

put : 3

put : 7

Get : 1

put : 4

put : 7

Get : 1

put : 5

Get : 7

8  
06/02 PM

20/02/24

## LAB 9

Write a program that makes a user interface to perform integer division.

```
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());  
jfrm.setDefaultCloseOperation(JFrame.  
    EXIT_ON_CLOSE);
```

```
JLabel jlab = new JLabel("Enter the divisor  
and dividend : ");
```

```
JTextField ajtf = new JTextField(8);  
JTextField bjtf = new JTextField(8);  
JButton button = new JButton("Calculate");
```

```
JLabel err = new JLabel();
```

```
JLabel alab = new JLabel();
```

```
JLabel blab = new JLabel();
```

```
JLabel anslab = new JLabel();
```

```
jfrm.add(err);
```

```
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 e)
    {
        System.out.println("Action event from
                           text field");
    }
};

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

```
button.addActionListener(new ActionListener())
```

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

```
        try
```

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

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

```
            int ans = a / b;
```

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

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

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

```
        catch (NumberFormatException e)
```

```
            alab.setText(" ");
```

```

        blab.SetText(" ");
        anslab.SetText(" ");
        err.SetText(" Enter Only Integers! ");
    }

}

jfrm.setVisible(true);

}

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

```

Output :

<input type="checkbox"/> Divider App	- <input type="checkbox"/> X
Enter divisor and dividend	
<input type="text"/> 10	<input type="text"/> 2
<input type="button" value="Calculate"/>	A=10 B=2 Ans=5

- > JFrame - The javax.swing.JFrame class is a type of container which inherits the javax.awt.Frame class. JFrame works like the main window where components like labels, textfields are added to create UI.
- > setSize (int width, int height) - used to resize a frame using width and height parameters.
- > setLayout() - method allows you to set the layout of the container. The layout manager helps lay out the components held by this container.
- > setDefaultCloseOperation() - methods is used to specify one of several options for the close button.  
JFrame.EXIT\_ON\_CLOSE - exit the application
- > JLabel - The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text.
- > JTextField - The object of a JTextField class is a text component that allows the editing of a single line text. It inherits JTextComponent class.
- > add(frame) - adds new frame in the existing frame.

- **ActionListener** - The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. This interface is found in `java.awt.event` package.
- **SetText()** - This method substitutes new text for all or part of the text in the text field. This works only with the first line of multi-line text fields.
- **SetVisible()** - is a method that has return type boolean

## LAB 10.

### Synchronous communication

class A

{

    int n;

    boolean valueSet = false;

    synchronized int get()

{

        while (!valueSet)

            try

            {

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

                wait();

            }

            catch (InterruptedException e)

            {

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

            }

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

            valueSet = true;

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

            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 ("In Intime 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 < 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<15)  
    {  
        int x=q.get();  
        System.out.println("Consumed:" + x);  
        i++;  
    }  
}
```

class PCFind

```
{  
    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:

Put : 0

Intimate consumer

Producer waiting

Got : 0

Intimate Producer

Put : 1

Intimate consumer

Producer waiting

Got : 1

Intimate Producer

Put : 2

Intimate Consumer

Producer waiting

Got : 2

Intimate Producer

Put : 3

Intimate Consumer

Producer waiting

Got : 3

Intimate Producer

Put : 4

Intimate Consumer

Producer waiting

Got : 4

Intimate Producer.

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);

```
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, "Racing Thread");
```

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

```
    a.foo(b);
```

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

```
public void run()
```

```
{
```

```
    b.bar(a);
```

System.out.println ("Back in other thread"),  
}

public static void main (String args[])

{  
    new Deadlock ();  
}

}

Output :

MainThread entered A.foo

RacingThread entered B.bar

MainThread trying to call B.last()

Inside A.last

Back in main thread

RacingThread trying to call A.last()

Inside A.last

Back in other thread

8  
13/27/2024

## LAB 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("Roo1 = 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("Roo1 = " + 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("Root2 = " + r1 + " - i " +r2);

    }

}

}

class QuadraticMain

{

    public static void main(String args[])

    {

        Quadratic q = new Quadratic();

        q.getd();

        q.compute();

    }

}

```

## LAB 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 Marks;
```

```
    int Credits;
```

```
    int Grade;
```

```
}
```

```
class Student
```

```
{
```

```
    Subject[] Subjects;
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Scanner s;
```

```
Student()
```

```
{
```

```
    int i;
```

```
    Subjects=new Subject[10];
```

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

```
{
```

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

```
}
```

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

```
}
```

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

void getMarks()
{
    int i;
    System.out.println("Enter the marks and credits of each subject");
    for(i=0;i<8;i++)
    {
        Subjects[i].Marks=s.nextInt();
        if(Subjects[i].Marks>100)
            Subjects[i].Grade=10;
        else if(Subjects[i].Marks>=90 && Subjects[i].Marks<=100)
            Subjects[i].Grade=10;
        else if(Subjects[i].Marks>=80 && Subjects[i].Marks<90)
            Subjects[i].Grade=9;
        else if(Subjects[i].Marks>=70 && Subjects[i].Marks<80)
            Subjects[i].Grade=8;
        else if(Subjects[i].Marks>=60 && Subjects[i].Marks<70)
            Subjects[i].Grade=7;
        else if(Subjects[i].Marks>=50 && Subjects[i].Marks<60)
            Subjects[i].Grade=6;
        else if(Subjects[i].Marks>=40 && Subjects[i].Marks<50)
            Subjects[i].Grade=5;
        else
            System.out.println("Fail");
    }
}
```

```
for(i=0;i<8;i++)
{
    System.out.println("enter the credits");
    Subjects[i].Credits=s.nextInt();
}

void computeSGPA()
{
    int i;
    int n,SGPA;
    int d=20;
    int n_sum=0;
    for(i=0;i<8;i++)
    {
        n=(Subjects[i].Grade)*(Subjects[i].Credits);
        n_sum=n+n_sum;
    }
    SGPA=(n_sum)/(d);
    System.out.println("SGPA="+SGPA);
}

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

### LAB 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;

    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="number of pages:"+this.numpages+"\n";
        return name+author+price+numpages;
    }
}
```

```
}

class Preeti

{

public static void main(String args[])

{

Scanner s=new Scanner(System.in);

int n,i;

String name;

String author;

int price;

int numpages;

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

n=s.nextInt();

books b[];

b=new books[n];

System.out.println("Enter the name,author,price and number of pages of the book");

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

{

name=s.next();

author=s.next();

price=s.nextInt();

numpages=s.nextInt();

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

}

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

{

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

}

}
```

## LAB 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;

abstract class shape
{
    int a,b;

    abstract void printarea();

    abstract void input();

    Scanner s=new Scanner(System.in);

}

class rectangle extends shape
{
    void input()
    {
        System.out.println("Enter the length and breadth");
        a=s.nextInt();
        b=s.nextInt();
    }

    void printarea()
    {
        System.out.println("The area of rectangle is:"+ (a*b)+ " +" + "square units\n");
    }
}

class triangle extends shape
{
```

```

void input()
{
    System.out.println("Enter the base and height");
    a=s.nextInt();
    b=s.nextInt();
}

void printarea()
{
    System.out.println("The area of triangle is:"+((a*b)/2)+" "+"square units\n");
}

class circle extends shape
{
    void input()
    {
        System.out.println("Enter the radius");
        a=s.nextInt();
    }

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

class area
{
    public static void main(String args[])
    {
        rectangle r=new rectangle();
        r.input();
    }
}

```

```

r.printarea();

triangle t=new triangle();
t.input();
t.printarea();

circle c=new circle();
c.input();
c.printarea();

}

}

```

## **LAB 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 cust_name;
    String acc_no;
    int acc_type;
    Scanner s=new Scanner(System.in);

    void get()
    {
        System.out.println("Enter the customer name:");
        cust_name=s.next();
        System.out.println("Enter the account number:");
        acc_no=s.next();
    }

    void get1()
    {
        System.out.println("Enter the account type:\n1:Savings\n2:Current account");
        acc_type=s.nextInt();
    }
}

class Cur_acc extends Account
{
    double balance=0;
    double deposit;
    void accept_dep()
    {
        System.out.println("Enter the amount to be deposited");
        deposit=s.nextInt();
        balance=balance+deposit;
        System.out.println("Amount"+deposit+" is successfully deposited");
    }

    void cheque()
}

```

```

{
    System.out.println("Enter the amount to be provided in cheque book");
    s.nextInt();
    System.out.println("The cheque book is issued successfully");
}

void ch_bal()
{
    if(balance>=500)
    {
        System.out.println("The balance amount is"+balance+"");
    }
    else
    {
        System.out.println("balance is less than minimum balance amount");
        balance=balance-5;
        System.out.println("Service charge is imposed and the current balance is"+balance);
    }
}

}

class Sav_acc extends Account
{
    double balance=0;
    double deposit,withd;
    int rate=6;

    void deposit()
    {
        System.out.println("Enter the amount to be deposited");
        deposit=s.nextInt();
        balance =balance+deposit;
    }
}

```

```
        System.out.println("Amount"+deposit+" is successfully deposited");
    }

    void withdrawal()
    {
        System.out.println("Enter the amount to withdraw");
        withd=s.nextInt();
        balance=balance-withd;
        System.out.println("The balance amount is"+balance);
    }

    void comp_interest()
    {
        balance=balance+balance*0.05;
        System.out.println("The balance amount is"+balance);
    }
}

class bank
{
    public static void main(String args[])
    {
        Scanner s1=new Scanner(System.in);
        int ch;
        Account b=new Account();
        b.get();
        Sav_acc a=new Sav_acc();
        Cur_acc c=new Cur_acc();
        do
        {
            b.get1();
            if(b.acc_type==1)
            {
                do
```

```

{
    System.out.println("-----Menu-----");
    System.out.println("Enter the choice\n1:deposit\n2:Withdrw\n3.Compound
interest");
    ch=s1.nextInt();
    switch(ch)
    {
        case 1:
            a.deposit();
            break;
        case 2:
            a.withdrawal();
            break;
        case 3:
            a.comp_interest();
            break;
    }
} while(ch!=4);

}

else
{
    do
    {
        System.out.println("-----Menu-----");
        System.out.println("Enter the choice\n1:deopsit\n2:Issue cheque\n3.check
balance");
        ch=s1.nextInt();
        switch(ch)
        {
            case 1:
                c.accept_dep();

```

```

        break;

    case 2:

        c.cheque();

        break;

    case 3:

        c.ch_bal();

        break;

    }

}

}while(ch!=4);

}

}

while(b.acc_type!=3);

}

}

```

## LAB 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals 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

package cie;

import java.util.Scanner;

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);

System.out.println("enter the usn of student");

usn= s.next();

System.out.println("enter the name of the student");

name= s.next();

System.out.println("enter the sem in which student is studying");

sem =s.nextInt();

}

public void displayStudentDetails()

{

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

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

System.out.println("Sem:" +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);

System.out.println("enter the marks ");

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

{

marks[i]=s.nextInt();

}
```

```
}

}

// Package see

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 calcuatefinalmarks()

{

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]);
}

}

// Class Main

import see.externals;

class main
{
    public static void main(String args[])
    {
        int numofstudents=2;
        externlas finalmarks[] =new externals[numofstudents];
        for(int i=0;i<numofstudents;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<numofstudents;i++)
        {
            finalmarks[i].calculatemark();
            finalmarks[i].dispalyfinalmarks();
        }
    }
}
```

## LAB 7

Write a program that demonstrates handling of exceptions in inheritance tree.

Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that 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 e)

    {

        super(e);

    }

}

class inputscanner

{

    Scanner s=new Scanner(System.in);

}

class father extends inputscanner

{

    int fatherage;

    void father_check() throws wrongage

    {

        System.out.println("Enter age of father\n");

        fatherage=s.nextInt();

        if(fatherage<0)

        {

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

        }

    }

}
```

```
        }

    }

void display()
{
    System.out.println("Age of the father is "+fatherage);
}

}

class son extends father
{
    int sonage;

    void son_check() throws wrongage
    {
        System.out.println("Enter age of son\n");
        sonage=s.nextInt();

        if(sonage>fatherage)
        {
            throw new wrongage("Age of son cannot be greater than father");
        }

        else if(sonage<0)
        {
            throw new wrongage("Age of son cannot be negative");
        }
    }

    void display()
    {
        System.out.println("Age of the father is "+sonage);
    }
}
```

```

class main
{
    public static void main(String[] args)
    {
        father f=new father();
        son s=new son();
        try
        {
            f.father_check();
            f.display();
            s.son_check();
            s.display();
        }
        catch(wrongage e)
        {
            System.out.println("Exception caught");
        }
    }
}

```

## **LAB 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.util.*;
class threadone implements Runnable
{
    String a;
    threadone(String b)
    {

```

```
a=b;  
Thread t=new Thread(this,a);  
t.start();  
}  
  
public void run()  
{  
    try  
{  
        while(true)  
{  
            System.out.println(a);  
            Thread.sleep(10000);  
        }  
    }  
    catch(InterruptedException e)  
{  
        System.out.println(a+"interrupted");  
    }  
    System.out.println(a+"exiting");  
}
```

```
class threadtwo implements Runnable  
{  
    String a;  
    threadtwo(String b)  
    {  
        a=b;  
        Thread t=new Thread(this,a);  
        t.start();  
    }
```

```

public void run()
{
    try
    {
        while(true)
        {
            System.out.println(a);
            Thread.sleep(2000);
        }
    }

    catch(InterruptedException e)
    {
        System.out.println(a+"interrupted");
    }

    System.out.println(a+"exiting");
}

}

class mainthread
{
    public static void main(String args[])
    {
        new threadone("BMSCE");
        new threadtwo("CSE");
    }
}

```

## **LAB 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 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();
```

```
jfrm.add(err);
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(ArithmeticException 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();
        }
    }
}
```

```
};  
}  
}
```

## LAB 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 Arithmetic Exception 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 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();
jfrm.add(err);
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(ArithmeticException 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();
        }
    });
}
}

```

## LAB 10

Demonstrate Inter process Communication and deadlock

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

    public void run()
{
```

```
b.bar(a);

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

}

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

}

// InterProcess 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<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<15)
        {
            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.");
}

}
```