

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:

RUSHILA.V

1BM22CS226

Department of Computer Science and
Engineering,

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

INDEX

Sl.No.	Title	Date
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

05/12/23

- 1) Write a C program to search for a key element using linear search
- 2) Write a C program to sort the numbers using Selection sort

Program 1

```
#include < stdlib.h >
#include < stdio.h >
void main ()
{
    int arr[10];
    int i, key;
    printf ("Enter array elements");
    for (i=0; i<5; i++)
        scanf ("%d", &arr[i]);
    printf ("Enter key element");
    scanf ("%d", &key);
    for (i=0; i<5; i++)
        if (arr[i] == key)
    {
        printf ("Element found");
        break;
    }
    else
        printf ("Element not found");
}
```

Output :

Enter array elements
27 22 35 39 46

Enter key element: 35

Key found

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

```
void main()
```

```
{
```

```
int arr[10];
```

```
int i, j, m;
```

```
int temp;
```

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

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

```
scanf("Enter the element");
```

```
for (i = 0; i < m; i++) {
```

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

```
}
```

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

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

```
if (arr[i] > arr[i + 1])
```

```
{
```

```
temp = arr[i];
```

```
arr[i] = arr[i + 1];
```

```
arr[i + 1] = temp;
```

```
}
```

```
}
```

Output:

Recd m: 3

Enter array element 1

45 4 45

55 20
55 23
05-12-23

12/12/23

classmate

{

public

{

Sept

}

})

Output

hello w

12/12/23

Quadratic Equation

dab program 1

```

import java.util.Scanner;
class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void get()
    {
        System.out.println("Rushika V, 1BM22CS226");
        Scanner s = new Scanner(System.in);
        System.out.println("Enter coefficient of a, b, c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
        while(a == 0)
        {
            System.out.println("not a quadratic equation");
            System.out.println("enter a non zero number");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b * b - 4 * a * c;
        if (a == 0)
        {
            r1 = (-b) / (2 * a);
            System.out.println("roots are real and equal");
            System.out.println("Root 1 = Root 2 = " + r1);
        }
    }
}
  
```

class if ($a > 0$)

$$\gamma_1 = ((-b) + (\text{math.sqrt}(a))) / (\text{a} * \text{c});$$

$$\gamma_2 = ((-b) - (\text{math.sqrt}(a))) / (\text{a} * \text{c});$$

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

System.out.println ("Root1 = " + γ_1 + "Root2 = " + γ_2);

class if ($a < 0$)

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

$$\gamma_1 = (-b) / (\text{a} * \text{c});$$

System.out.println ("Root1 = " + γ_1 + " + i " + γ_2);

System.out.println ("Root1 = " + γ_1 + " + i " + γ_2);

class QuadraticMain

public static void main (String args [])

Quadratic q = new Quadratic ();

q.getd ();

q.compute ();

Output 1

Rishika.V

1BM22CS226

Enter the coefficients of a, b, c

1

-5

2

Roots are real and distinct

root1 = 4.5615628128 root2 = 1.5615528128

Output 2:

Rushila N 1BM22CS226

Enter the coefficients of a, b, c

1

2

1

roots are real and equal

Root 1 = Root 2 = 1.0

Output 3

Rushila N 1BM22CS226

Enter the coefficients of a, b, c

0

4

5

not a quadratic equation

enter a non zero value for a:

1

roots are imaginary

root 1 = -2.0 + iNaN

root 2 = -2.0 - iNaN

19/12/23 dab - program 2 : SGPA

import java.util.Scanner;
class Subject

{

int subjectMarks;

int credits;

int grade;

{

class Student

{

Subject subject[];

Scanner s = new Scanner(System.in);

String name;

String usn;

double sgpa=0;

Student();

{

int i;

subject = new Subject[9];

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

subject[i] = new Subject();

{

void getStudentDetails()

{

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

name = s.nextLine();

System.out.println("Enter the usn: ");

usn = s.nextLine();

{

```
void getmarks()
```

{

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

{

```
    System.out.println("Enter the subject marks  
of the subject " + (i + 1) + ": ");
```

```
    Subject[i].subjectMarks = s.nextInt();  
    if (Subject[i].subjectMarks >= 90)
```

{

```
        Subject[i].grade = 10;
```

}

```
    else if (Subject[i].subjectMarks >= 80)
```

{

```
        Subject[i].grade = 9;
```

}

```
    else if (Subject[i].subjectMarks >= 70)
```

{

```
        Subject[i].grade = 8;
```

}

```
    else if (Subject[i].subjectMarks >= 60)
```

{

```
        Subject[i].grade = 7;
```

}

```
    else if (Subject[i].subjectMarks >= 50)
```

{

```
        Subject[i].grade = 6;
```

}

```
    else if (Subject[i].subjectMarks >= 40)
```

{

```
        Subject[i].grade = 5;
```

}

```
else
```

{

```
    Subject[i].grade = 0;
```

}

System.out.println("Enter the credits of " +
"Subject " + (i+1) + " : ");
subject[i].credit = s.nextInt();
}

}

void computeSGPA()

{
int totalCredits = 0;
int totalCreditHours = 0;
for (int i = 0; i < 8; i++)

totalCredits += subject[i].grade * subject[i].
credits;

totalCreditHours += subject[i].credits;

SGPA = (double) totalCredits / totalCreditHours;
System.out.println("SGPA: " + SGPA);

}

clearStudentMain

{

public static void main (String args[])

{
Student S1 = new Student();
S1.getStudentDetails();
S1.getMarks();
S1.computeSGPA();

}

Output:

Enter the name:

Rushila. ✓

Enter the user id:

IBM2acsdaa

Enter the subject marks of the subject 1:

93

Enter the credit of the subject 1:

4

Enter the subject marks of the subject 2:

94

Enter the credits of subject 2:

4

Enter the subject marks of subject 3:

91

Enter the credits of subject 3:

3

Enter the subject marks of subject 4:

88

Enter the credits of subject 4:

1

Enter the subject marks of subject 5:

94

Enter the credits of the subject 5:

3

Enter the subject marks of subject 6:

92

Enter the credits of subject 6:

1

Enter the subject marks of subject 7:

87

Enter the credits of subject 7:

3

Enter the subject marks of subject 8:

100
100
100
100

26/12/23 lab-program3

Create a class Book which contains four members name, author, price, numPages. Include a constructor to set the values for the members. Include methods to set and get the details of the object. Include a toString() method that could display the complete details of the book.

Review a Java program to create and work with objects.

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

```
class Books
```

```
{
```

```
    String name;
```

```
    String author;
```

```
    int price;
```

```
    int numPages;
```

```
    int numPages)
```

```
Books(String name, String author, int price,
```

```
{
```

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

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

```
    this.price = price;
```

```
    this.numPages = numPages;
```

```
}
```

```
public String toString()
```

```
{
```

```
    String name, author, price, numPages;
```

```
    return "Book name" + this.name + "\n" +
```

```
    "Author Name" + this.author + "\n" +
```

```
    "Price" + this.price + "\n" + "number of pages"
```

```
    + this.numPages + "\n";
```

```
}
```

class Bookmain

{

public static void main (String args[])

{

Scanner s = new Scanner (System.in);

String; System.out.print ("Enter the name: ");

System.out.print ("Enter the USN ");

n = s.nextInt();

Books7 books = new Books [n];

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

{

System.out.print ("Enter the details of the book");

System.out.print ("Name of the book");

String name = s.next();

System.out.print ("Author of the book");

String author = s.next();

System.out.print ("Price of the book");

int price = s.nextInt();

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

int numPages = s.nextInt();

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

System.out.print ("Details of the book");

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

System.out.print (books[i]);

{

3

4

Output:

Enter the name:

Pushila. ✓

Enter the usn:

13M22CS225

Enter the number of books

3

Enter the details of the book

Name of the book

Divergent

Author of the book

Roth

Price of the book

230

Number of pages

123

Name of the book

Dracula

Author of the book

Stoker

Price of the book

345

Number of pages

276

Details of the book

Book name : Divergent

Author name : Roth

Price : 230

Number of pages : 123

Book name : Dracula

Author name :

02/01/24 Date Program-2 : Area

import java.util.Scanner;
abstract class Shape

double dims dim1, radius;
Shape (double a, double b)
{

dim1 = a;
dim2 = b;

shape (double a)
{

radius = a;
abstract void area();

class Rectangle extends shape
{

Rectangle (double a, double b)
{

super (a, b);

void area ()
{

System.out.println ("Area of Rectangle " +
(dim1 * dim2));

class Triangle extends shape
{

Triangle (double a, double b)
{

super (a, b);

void area ()
{

System.out.println("area of triangle: " + triangle);

} class Circle extends Shape

{ circle (double a)

{ super(a);

{ void area()

{ System.out.println("area of circle " +
(3.14 * (radius) * (radius)));

} class AbstractPreamain

{ public static void main (String args[])

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

System.out.println("Rushila.V");

System.out.println("1BM22CS226");

System.out.println("Enter the length & breadth
of a rectangle");

double l = s.nextInt();

double b = s.nextInt();

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

double n1 = s.nextInt();

double n2 = s.nextInt();

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

double r = s.nextInt();

double ar;

Rectangle rec = new Rectangle(l, b);

Triangle tri = new Triangle (l, n1, n2);

Circle cir = new Circle (r);

ar = rec.area();
ar area();
ar = tri;
ar area();
ar = cir;
ar area();
}

Output:

Rushila.V
1BM22CS226

Enter the length

9

6

Enter the breadth

3

4

Enter the base

9

Area of rectangle

Area of triangle

Area of circle

sh = recr ;
sh area();
sh = tri ;
sh area();
sh = cir ;
sh area();
}

Output :

Rushila.v

IBM22CS226

enter the length and breadth of rectangle

9

6

enter the base and height

3

4

enter the radius

9

area of rectangle : 45.0

area of triangle : 6.0

area of circle : 254.34

21/01/24

09/01/24 lab program 5: Bank

```
import java.util.Scanner;  
class BankAccount;
```

{

```
String customerName;
```

```
int accno;
```

```
String type;
```

```
double balance;
```

```
account (String name, int accno, String type,  
double balance)
```

{

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

```
this.accno = accno;
```

```
this.type = type;
```

```
this.balance = balance;
```

{

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

{

```
balance += amount;
```

{

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

{

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

{

```
balance -= amount;
```

{

```
else
```

{

```
System.out.println ("Insufficient balance,  
can't withdraw");
```

{

{

```
void display ()
```

S

```
System.out.  
println
```

{

{

```
class BankA
```

{

```
private  
BankAcct
```

{

```
deposit
```

{

```
void inter
```

{

```
balance
```

```
System
```

{

```
clear cu
```

{

```
privat
```

```
privat
```

```
withdr
```

{

```
acc
```

{

```
void d
```

{

```
if (ba
```

{

```
dep
```

{

```
mi
```

```
nu
```

2
system.out.println ("name: " + name + " accno: " +
accno + " type: " + type + " balance: " + balance);

3

3

class SavAcct extends Account

{

private static double rate = 5;

SavAcct (String name, int accno, double balance)

{

super(name, accno, "savings", balance);

3

3

void interest()

{

balance += balance * (rate) / 100;

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

3

3

class curAcct extends Account

{

private double minBal = 500;

private double servicecharge = 50.

curAcct (String name, int accno, double balance)

{

super(name, "accno", "current", balance);

3

3

void checkMin()

{

if (balance < minBal)

{

System.out.println ("Balance is less than

min balance service charges imposed: " + service
charge);

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

{

}

class AccountMain

{

public static void main (String args) {

{

Scanner s = new Scanner (System.in);

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

String name = s.next();

System.out.println ("Enter the type (current
savings) : ");

String type = s.next();

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

int accno = s.nextInt();

System.out.println ("Enter the initial balance");

double balance = s.nextDouble();

int ch;

double amount; int account;

balance;

Account acc = new Account (name, accno, type,

CurrentAcc ta = new CurrentAcc (name, accno, balance);

SavAcc sa = new SavAcc (name, accno, balance);

while (true)

{

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

System.out.println ("Enter 1. deposit
2. withdraw 3. compute interest 4. display");

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

ch = s.nextInt();

switch (ch) {

{

case 1 : deposit;

amount

da. ac

break;

case 2 : withdraw;

amount

da. ac

break;

case 3 : da.

break;

case 4 : da.

break;

case 5 : default:

J

{

}

else

{

System.out.

withdraw;

System.out.

ch = s.nextInt();

switch

{

case 1 :

amount

da. ac

break;

case 2 :

amount

da. ac

break;

case 3 :

amount

da. ac

break;

```
case 1 : System.out.println ("Enter the amount: ");
amount1 = s.nextInt ();
ca.deposit (amount1);
break;

case 2 : System.out.println ("Enter the amount");
amount2 = s.nextInt ();
ca.withdraw (amount2);
break;

case 3 : ca.Interest ();
break;

case 4 : ca.display ();
break;

case 5 : System.exit (0);

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

}

}

else

{

```
System.out.println ("In main 1.deposit 2.
withdraw 3.display ");
System.out.println ("Enter the choice");
ch = s.nextInt ();
switch (ch)
```

{

```
case 1 : System.out.println ("Enter the amount: ");
amount1 = s.nextInt ();
ca.deposit (amount1);
break;
```

~~```
case 2 : System.out.println ("Enter the amount");
amount2 = s.nextInt ();
ca.withdraw (amount2);
ca.checkmin ();
break;
```~~

case 3 : cout << display();

break;

case 4 : system . exit(0);

default : system . out << endl ("Invalid input").  
break;

}

,

,

,

Output

enter the name : Rushilay

enter the type (current/savings),

current

enter the account number :

1234

enter the initial balance :

4555

menu

3.

name : Rushilay

accno. 1234

type current balance : 4578.0

menu

1. deposit 2. withdraw 3. display  
enter the choice:

1

enter the amount :

3456

menu

1. deposit

enter the ch

2

enter the am

345

menu

3. dis

1. deposit

enter the

3

name : Ru

accno. 1

type : cu

balance

menu

1. deposit 2. withdraw 3. display

Enter the choice

2

Enter the amount

345

menu

3. display

1. deposit 2. withdraw 3. display

Enter the choice:

3

Name: Rushila \*

accno: 1234

Type: current

balance: 7666

Open  
01-01-2019

23/01/24 Java program - Packages

student.java

package C16;

import java.util.Scanner;

public class student {

protected String USN = new String();

protected String name = new String();

protected int dem;

}

public void display Student details() {

{

System.out.println("USN of student : " + USN);

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

System.out.println("Dem : " + dem);

}

internal.java

import java.util.Scanner;

package C16;

public class internal extends student {

protected int marks[] = new int[5];

public void input(C16 marks[])

Scanner s = new Scanner(System.in);

System.out.println("Enter internal marks for five subjects");

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

marks[i] = s.nextInt();

}

## Externals.java

package S6;

import java.io.IOException;

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 inputS6() {

Scanner s = new Scanner(System.in);

for (int i = 0; i &lt; 5; i++) {

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

marks[i] = s.nextInt();

public void calculateFinalMarks() {

for (int i = 0; i &lt; 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])~~

{}

## Main.java

import

class

public

{

System

extern

los

{

fin

fi

su

fi

de

fo

s

fu

fi

dr

su

{

ac

s

u

d

Main.java

import SIS.Externals;

class Main {

    public static void main (String args) {

        int numofStudents = 2;

        Externals finalMarks[] = new Externals [numofStudents];

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

            finalMarks[i] = new Externals();

            finalMarks[i].inputStudentDetails();

            System.out.println ("Enter CIE marks");

            finalMarks[i].inputCIEmarks();

            System.out.println ("Enter SIS marks");

            finalMarks[i].inputSISmarks();

        System.out.println ("Displaying data: In ");

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

            finalMarks[i].calculateFinalMarks();

            finalMarks[i].displayFinalMarks();

}

}

Student.java

public void inputStudentDetails ()

Scanner s = new Scanner (System.in);

System.out.print ("Enter usn");

int usn = s.nextInt();

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

char name = s.next();

String name1 = name + ". " + usn; int sem = s.nextInt();

Output:

enter the usn:

13M22CS226

enter the name:

Rashida

enter the sem:

3

enter sic marks

enter internal marks of course 1

45

enter internal marks of course 2

46

enter internal marks of course 3

47

enter internal marks of course 4

48

enter internal marks of course 5

49

enter sic marks

enter external marks of course 1

99

enter external marks of course 2

88

enter external marks of course 3

98

enter external marks of course 4

87

enter external marks of course 5

99

center the usn

13M22CS226

center the name

Rashida

center the semester

3

center CFT marks

center internal marks

45

center internal marks

46

center internal marks

47

center internal marks

48

center internal marks

49

center internal marks

50

center internal marks

51

center internal marks

52

center external marks

99

center external marks

87

center external marks

89

center external marks

98

enter the course number  
9BMA22CS216

enter the name  
Renu

enter the sem  
3

enter CF marks

enter internal marks of course 1  
45

enter internal marks of course 2  
43

enter internal marks of course 3  
46

enter internal marks of course 4  
50

enter internal marks of course 5  
44

enter dec marks

enter external marks of course 1  
98

enter external marks of course 2  
99

enter external marks of course 3  
87

enter external marks of course 4  
89

enter external marks of course 5  
98

displaying details:

name: Rekha id: 1BM22CS226 Sem: 3

course 1: 94

course 2: 86

course 3: 96

course 4: 91

course 5: 98

name: Ravi id: 1BM22CS216 Sem: 3

course 1: 94

course 2: 92

course 3: 89

course 4: 94

course 5: 89

~~18/01/2023~~

~~23-01-2023~~

30/01/2023 lab prep

import pa

classmate

{

function

{

self

}

classmate

function

function

if (

else

for

3

then

3

function

{

re

3

3

classmate

function

function

if

else

3

## sol/offer lab program - 7: Exceptions

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

import java.util.Scanner;  
class WrongAge extends Exception  
{}

public WrongAge(String s)

{super(s);}

class Father {

private int age;

public Father(int age) throws WrongAge {

if (age < 0) {

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

}

this.age = age;

}

public int getAge()

{

return age;

}

class Son extends Father {

private int sonAge;

public Son(int fatherAge, int sonAge)

throws WrongAge

{

super(fatherAge);

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");

this.sonAge = sonAge;

}

06/02/24 Threadedig

class BMS c  
S

public void

for (int i =

System.out.println("System.out.println(" + i + ")");

try {

Thread

catch (Tr

c . pri

}

}

class CS

public

for (ci

days

try {

}

catch (e

public int getDadAge() {

return sonAge;

}

}

public class ExceptionDemo {

public static void main (String args[]) {

}

Scanner s = new Scanner (System.in);

try {

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

int fatherAge = s.nextInt();

System.out.println("Enter son age");

int sonAge = s.nextInt();

Son son = new Son (fatherAge, sonAge);

System.out.println ("Father's age: " + son.getFpa());

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

}

catch (Exception e) {

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

}

}

Output:

Enter father age

-12

Enter son age

10

Error: Age cannot be negative

Enter father age

20

Enter son age

30

Error: Son's age

20.02.24  
 20.02.24  
 20.02.24

06/02/24 Arreaching

class BMS extends Thread

public void run()

for (int i=1; i<=5; i++)

system out. fourth ("BMS college of engineering"),  
try

? Thread.sleep(1000);

case (Interest+cdException) {

c .printStacktrace()

{

1

---

and its external form

and is often spread

buffic void sun(

www.vista.com)

for (int i = 1; i <= s.length)

$\text{g} = 1, \infty, ++$

system-out.println();

471

*any*

~~Thread. sleek (2000)~~

3.  $\frac{1}{2} \times \frac{1}{2}$

~~With C Interruptions Except~~

e. Print Stack Trace ( )

3

*[A blank horizontal line for a signature.]*

*[A horizontal line for a signature.]*

class Threadmain

{  
    public static void main (String args[7])

    BMS b1 = new BMS();  
    CS c1 = new CSC();  
    b1.start();  
    c1.start();  
}

Output :

BMS college of engineering,

ESB 1

ESE 1

CSB 3

CSB 4

CSB 5

BMS college of engineering 2

BMS college of engineering 3

BMS college of engineering 4

BMS college of engineering 5

date 15  
IPC

206/02/24

class

unit

boolc

sync

{

while

try

depth

wait

}

catch

args

{

depth

value

deps

noti

dcn

}

sync

whi

try

deps

wa

{ca

args

{

this

val

sysk

sysk

notif

class Q {

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("InterruptedException caught");

}

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

valueSet = false;

System.out.println("In Intimate producer (" + 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;

valueSet = true;

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

System.out.println("In Intimate consumer (" + n + ")");

notify();

}

~~class Producer implements Runnable~~

3

6. q

Production ( $Q_q$ ) {

~~the~~. q : q.

newRecord (This, "Products") start( ).

2

generic void run() {

until  $j \geq 0$ .

while ( $i \leq 15$ ). {

q. frw (ift);

2  
1

1

QAM consumer implements Ruuviable

Q q,

Consumo ( $A_q$ ) {

$$\text{Skin } q = q.$$

new thread (thus, "consumer"). `start()`

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

```
int i>0;
while(i<15)
```

```
int x = q.QetC()
```

~~System out . println ("consumed: " + s)~~

3

1

1

7

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

Output:

Put: 1

Get: 1

Put: 2

Get: 2

Put: 3

Get: 3

Put: 4

Get: 4

Put: 5

Get: 5

See SFS pg. 72 ~ 74

13/02/24

## 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("Interrupted");

}

System.out.println("trying to call B.bar");

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 out ()

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

}

}  
class Deadlock implements Runnable

{  
    A a = new A();

    B b = new B();

    Deadlock ()  
}

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

Thread t = new Thread (this, "Racing Thread");  
t.start();

a.foo();

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

Main thread entered A.f( )

Backing thread entered B.main

Main thread trying to call B.main()

Inside B.last

Back in main thread

Backing thread trying to call A.last()

Inside A.last

Back in other thread

~~13.02.21~~

~~13.02.21~~

~~13.02.21~~

20/02/24 lab 1 - program no: 9 : AWT

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class swingDemo {
 swing Demo() {
 // create JFrame container
 JFrame jfrm = new JFrame("Divide Two Numbers");
 jfrm.setSize(275, 150);
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 jfrm.setLayout(new GridLayout());
 // to terminate on close
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }
}
```

// text label

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

// add text for both numbers

```
JTextField jtf1 = new JTextField(8);
```

```
JTextField jtf2 = new JTextField(8);
```

// calc button

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

// answer

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

```
JLabel ansLab = new JLabel();
```

```
JLabel decLab = new JLabel();
```

```
JLabel ansLab = new JLabel();
```

// add in order :)

```
jfrm.add(jlab); // to display error box
```

```
jfrm.add(jtf1);
```

```
jfrm.add(jtf2);
```

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

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

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

jfm.add(blue);  
jfm.add(button);

ifm.addActionListener();

ActionListener l : new ActionListener() {

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

}

3.

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;

else setText("In A = " + a);

else setText("In B = " + b);

ansLabel.setText("In Ans: " + ans);

} catch (NumberFormatException e) {

else.setText(" ");

else.setText(" ");

ansLabel.setText(" ");

err.setText("Enter only Integers!");

} catch (ArithmeticException e) {

else.setText(" ");

else.setText(" ");

ansLabel.setText(" ");

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

3.

3).

1) display  
jfml.setVisible(true);

freebie at

1) create

Swing util

freebie

new su

3

3);

3;

}

Output :

Enter A:

24

Enter B:

24

call

Function

1) JFrame

division

2) setA

3) setDa

for st

for ce

do wh

//display frame

ifrm.setVisible(true);

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

Output:

Enter the divisor and dividend :

24

12

calculate

A=24, B=12 Ans=2

Functions

1) JFrame : used to create GUI for a simple division app

2) setsize : set the size of the JFrame

3) setLayout : used to set the layout manager for the JFrame. The layout manager is responsible for arranging the components that are added to the JFrame.

- 4) setDefaultCloseOperation() method: used to set the default close operation for the JFrame.
- 5) JLabel: used to create a label component.
- 6) JTextField: used to create a textfield component.  
A text field component is a GUI component that allows the user to enter and edit a single line of text.
- 7) add() method: used to add components to the JFrame. The add() method is called on the JFrame object 'jfum'. The add() method takes a single parameter which is the component that should be added to the JFrame.
- 8) addActionListener() method: used to add an action listener to a component. An action listener is a GUI component that listens for action events.
- 9) setLayout() method: can be used to set the layout of any label component. The 'setLayout()' method can be used to change the layout of a label component dynamically.

~~Set & Change Layout~~

## 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;
 Scanner s;

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

 }

 void getd() {
 System.out.println("Rushila.V");
 System.out.println("1BM22CS226");
 System.out.println("Enter coefficients of a, b and 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 number");
a = s.nextInt();
}

d = b * b - 4 * a * c;
if (d == 0) {
 r1 = (-b) / (2 * a);
 System.out.println("Roots are real and equal");
 System.out.println("Root1 = Root2=" + r1);
} else if (d > 0) {
 r1 = ((-b) + (Math.sqrt(d))) / (double) (2 * a);
 r2 = ((-b) - (Math.sqrt(d))) / (double) (2 * a);
 System.out.println("Roots are real and distinct");
 System.out.println("Root1 = " + r1 + " Root2 = " + r2);
} else if (d < 0) {
 System.out.println("Roots are imaginary");
 r1 = (-b) / (2 * a);
 r2 = Math.sqrt(-d) / (2 * a);
 System.out.println("Root1 = " + r1 + " + i" + r2);
 System.out.println("Root1 = " + r1 + " - i" + r2);
}
}

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

```

Quadratic q = new Quadratic();
q.getd();
q.compute();
}

}

```

```

PS C:\Users\Rushila V\OneDrive\Desktop> javac QuadraticMain.java
PS C:\Users\Rushila V\OneDrive\Desktop> java QuadraticMain
Rushila.V
1BM22CS226
Enter coefficients of a, b and c
1
2
1
Roots are real and equal
Root1 = Root2=-1.0
PS C:\Users\Rushila V\OneDrive\Desktop> java QuadraticMain
Rushila.V
1BM22CS226
Enter coefficients of a, b and c
0
4
5
not a quadratic equation
Enter a non zero number
1
Roots are imaginary
Root1 = -2.0 + i1.0
Root1 = -2.0 - i1.0
PS C:\Users\Rushila V\OneDrive\Desktop> java QuadraticMain
Rushila.V
1BM22CS226
Enter coefficients of a, b and c
0
1
5
not a quadratic equation
Enter a non zero number
2
Roots are imaginary
Root1 = 0.0 + i1.5612494995995996
Root1 = 0.0 - i1.5612494995995996

```

## 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 subjectmarks;
 int credits;
 int grade;

 Subject(int credits) {
 this.credits = credits;
 }
}
```

```
class Student {
 Subject[] subject;
 Scanner s = new Scanner(System.in);
 String name;
 String usn;
 double sgpa = 0;
```

```
Student() {
 int i;
 subject = new Subject[9];
 for (i = 0; i < 9; i++) {
 subject[i] = new Subject(4);
 }
}
```

```
void getStudentDetails() {
 System.out.println("Rushila.V");
 System.out.println("1BM22CS226");
 System.out.println("Enter the name:");
 name = s.nextLine();
 System.out.println("Enter the usn:");
 usn = s.nextLine();
}

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

 if (subject[i].subjectmarks >= 90) {
 subject[i].grade = 10;
 } else if (subject[i].subjectmarks >= 80) {
 subject[i].grade = 9;
 } else if (subject[i].subjectmarks >= 70) {
 subject[i].grade = 8;
 } else if (subject[i].subjectmarks >= 60) {
 subject[i].grade = 7;
 } else if (subject[i].subjectmarks >= 50) {
 subject[i].grade = 6;
 } else {
 subject[i].grade = 0;
 }
 }
}
```

```
 }

 System.out.println("Enter the credits of the subject" + (i + 1) + ":");

 subject[i].credits = s.nextInt();

}

}

void computesgpa() {

 int totalcredits = 0;

 int totalcredithours = 0;

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

 totalcredits += (subject[i].grade * subject[i].credits);

 totalcredithours += subject[i].credits;

 }

 sgpa = (double) totalcredits / totalcredithours;

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

}

}

class studentmain {

 public static void main(String[] args) {

 Student s1 = new Student();

 s1.getStudentDetails();

 s1.getMarks();

 s1.computesgpa();

 }

}
```

```
}
```

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac studentmain.java
PS C:\Users\Rushila V\OneDrive\Desktop> java studentmain
Rushila.V
1BM22CS226
Enter the name:
luke
Enter the usn:
1bm22cs100
Enter the subject marks of the subject1:
99
Enter the credits of the subject1:
4
Enter the subject marks of the subject2:
88
Enter the credits of the subject2:
3
Enter the subject marks of the subject3:
97
Enter the credits of the subject3:
3
Enter the subject marks of the subject4:
86
Enter the credits of the subject4:
1
Enter the subject marks of the subject5:
94
Enter the credits of the subject5:
3
Enter the subject marks of the subject6:
92
Enter the credits of the subject6:
1
```

```
Enter the subject marks of the subject7:
96
Enter the credits of the subject7:
1
Enter the subject marks of the subject8:
87
Enter the credits of the subject8:
3
SGPA=9.631578947368421
```

### 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() {
```

```
 return "Book name: " + this.name + "\n" +
```

```
 "Author name: " + this.author + "\n" +
```

```
"Price: " + this.price + "\n" +
"Number of pages: " + this.numPages + "\n";
}

}

public class Main {
 public static void main(String args[]) { Scanner
 s = new Scanner(System.in);int n;
 System.out.println("Rushila.V");
 System.out.println("1BM22CS226");
 System.out.print("Enter the number of books: ");n =
 s.nextInt();
 Books[] b = new Books[n];for
 (int i = 0; i < n; i++) {
 System.out.print("Enter the name of book " + (i + 1) + ": ");String name =
 s.next();
 System.out.print("Enter the author of book " + (i + 1) + ": ");String author
 = s.next();
 System.out.print("Enter the price of book " + (i + 1) + ": ");int price =
 s.nextInt();
 System.out.print("Enter the number of pages of book " + (i + 1)
 + ": ");
 int numPages = s.nextInt();
 }
 }
}
```

```

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

 }

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

```

```

PS C:\Users\Rushila V\OneDrive\Desktop> javac Main.java
PS C:\Users\Rushila V\OneDrive\Desktop> java Main
Rushila.V
1BM22CS226
Enter the number of books: 3
Enter the name of book 1: Divergent
Enter the author of book 1: Ruth
Enter the price of book 1: 230
Enter the number of pages of book 1: 123
Enter the name of book 2: Dracula
Enter the author of book 2: Stoker
Enter the price of book 2: 345
Enter the number of pages of book 2: 276
Enter the name of book 3: 1876
Enter the author of book 3: bailey
Enter the price of book 3: 400
Enter the number of pages of book 3: 321
Book name: Divergent
Author name: Ruth
Price: 230
Number of pages: 123

Book name: Dracula
Author name: Stoker
Price: 345
Number of pages: 276

Book name: 1876
Author name: bailey
Price: 400
Number of pages: 321

```

## 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 suchthat each one of the classes extends the classShape. Each one of the classes contain only the method printArea( ) that prints the areaof the given shape.**

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

```
abstract class Shape {
```

```
 double dim1, dim2, radius;
```

```
 Shape(double a, double b) {
```

```
 dim1 = a;
```

```
 dim2 = b;
```

```
}
```

```
 Shape(double a) {
```

```
 radius = a;
```

```
}
```

```
 abstract void area();
```

```
}
```

```
class Rectangle extends Shape {
 Rectangle(double a, double b) {
 super(a, b);

 }

 void area() {
 System.out.println("Area of rectangle: " + (dim1 * dim2));

 }
}
```

```
class Triangle extends Shape {
 Triangle(double a,
 double b) {
 super(a, b);

 }

 void area() {
 System.out.println("Area of triangle: " + (dim1 * dim2) / 2);

 }
}
```

```
class Circle extends Shape {
 Circle(double a) {
 super(a);

 }
}
```

```
void area() {
 System.out.println("Area of circle: " + (3.14 * (radius) * (radius)));
}
}

class AbstractAreaMain {
 public static void main(String args[]) {
 Scanner s = new Scanner(System.in);
 System.out.println("Rushila.V");
 System.out.println("1BM22CS226");
 System.out.println("Enter the length and breadth of a rectangle");
 double l = s.nextDouble(); double b
 = s.nextDouble();
 System.out.println("Enter the base and height of a triangle");
 double h1 = s.nextDouble();
 double h2 = s.nextDouble();
 System.out.println("Enter the radius of a circle");
 double r
 = s.nextDouble();
 Shape sh;
 Rectangle rect = new Rectangle(l, b);
 Triangle tri = new Triangle(h1, h2); Circle cir
 = new Circle(r);
 }
}
```

```
 sh = rect;
 sh.area();
 sh = tri;
 sh.area();
 sh = cir;
 sh.area();
}
}
```

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac AbstractAreaMain.java
PS C:\Users\Rushila V\OneDrive\Desktop> java AbstractAreaMain
Rushila.V
1BM22CS226
Enter the length and breadth of a rectangle
9
5
Enter the base and height of a triangle
3
4
Enter the radius of a circle
9
Area of rectangle: 45.0
Area of triangle: 6.0
Area of circle: 254.34
```

## 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)

**a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance**

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

```
class Account {
```

```
 String name;
```

```
 int accno;
```

```
 String type;
```

```
 double balance;
```

```
 Account(String name, int accno, String type, double balance) {this.name =
```

```
 name;
```

```
 this.accno = accno;
```

```
this.type = type; this.balance
= balance;
}

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

void withdraw(double amount) {if
((balance - amount) >= 0) {
 balance -= amount;
} else {
 System.out.println("Insufficient balance, cannot withdraw");
}
}

void display() {
 System.out.println("Name: " + name + ", Account Number: " +accno + ",
Type: " + type + ", Balance: " + balance);
}
}

class SavAcct extends Account {
```

```
private static double rate = 5;

SavAcct(String name, int accno, double balance) {super(name, accno,
 "savings", balance);
}

void interest() {
 balance += balance * (rate) / 100;
 System.out.println("Balance after adding interest: " + balance);
}

}

class CurAcct extends Account {

private double minBal = 500; private
double serviceCharges = 50;

CurAcct(String name, int accno, double balance) {
 super(name, accno, "current", balance);
}

void checkMin() {
 if (balance < minBal) {
```

```
 System.out.println("Balance is less than minimum balance,service
charges imposed: " + serviceCharges);

 balance -= serviceCharges;

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

 }

}

}
```

```
public class AccountMain {

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

 System.out.println("Rushila.V");

 System.out.println("1BM22CS226");

 System.out.println("Enter the name:");String
 name = s.next();

 System.out.println("Enter the type (current/savings):");String type
 = s.next();

 System.out.println("Enter the account number:");int
 accno = s.nextInt();

 System.out.println("Enter the initial balance.");double
 balance = s.nextDouble();

 int ch;

 double amount1, amount2;
```

```
Account acc;

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

 acc = new SavAcct(name, accno, balance);

} else {

 acc = new CurAcct(name, accno, balance);

}

while (true) {

 if (acc.type.equalsIgnoreCase("savings")) {

 System.out.println("\nMenu\n1. Deposit\n2. Withdraw\n3.
Compute Interest\n4. Display\n5. Exit");

 System.out.println("Enter your choice:");
ch =
s.nextInt();

switch (ch) {

 case 1:

 System.out.println("Enter the amount to deposit:");
amount1 =
s.nextDouble();
acc.deposit(amount1);

 break;

 case 2:

 System.out.println("Enter the amount to withdraw:");
amount2 =
s.nextDouble();
acc.withdraw(amount2);

 break;
 }
}
}
```

```
case 3:
 ((SavAcct) acc).interest();
 break;

case 4:
 acc.display();
 break;

case 5:
 System.exit(0);

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

} else {
 System.out.println("\nMenu\n1. Deposit\n2. Withdraw\n3.
Display\n4. Exit");

 System.out.println("Enter your choice:");ch =
 s.nextInt();

 switch (ch) {

 case 1:
 System.out.println("Enter the amount to deposit:");amount1 =
 s.nextDouble();
 acc.deposit(amount1);
 break;

 case 2:
```

```
System.out.println("Enter the amount to withdraw:");amount2 =
s.nextDouble();
acc.withdraw(amount2);
((CurAcct) acc).checkMin();
break;

case 3:
 acc.display();
 break;

case 4:
 System.exit(0);

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

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac AccountMain.java
PS C:\Users\Rushila V\OneDrive\Desktop> java AccountMain
Rushila.V
1BM22CS226
Enter the name:
ravi
Enter the type (current/savings):
current
Enter the account number:
1234
Enter the initial balance:
4555

Menu
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice:
1
Enter the amount to deposit:
2

Menu
1. Deposit
2. Withdraw
3. Display
4. Exit
Enter your choice:
3
Name: ravi, Account Number: 1234, Type: current, Balance: 4557.0
```

## 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;
```

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

```
public class Student {
 protected String usn = "";
 protected String name = "";
 protected int sem;

 // Method to input student details
 public void inputStudentDetails() {
 Scanner scanner = new Scanner(System.in); System.out.print("Enter USN: ");
 usn = scanner.nextLine(); System.out.print("Enter
 Name: ");
```

```
name = scanner.nextLine();

System.out.print("Enter Semester: ");sem
= scanner.nextInt();

}

// Method to display student detailspublic
 void displayStudentDetails() {
 System.out.println("USN: " + usn);
 System.out.println("Name: " + name); System.out.println("Semester: " +
sem);
 }
}

package CIE;

import java.util.Scanner;

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

 // Method to input CIE marks
 public void inputCIEMarks() {
 Scanner scanner = new Scanner(System.in);
```

```
System.out.println("Enter CIE marks for 5 subjects:");for (int i
= 0; i < 5; i++) {
 System.out.print("Subject " + (i + 1) + " marks: ");marks[i] =
scanner.nextInt();
}
}

package SEE;

import CIE.Internals; import

java.util.Scanner;

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

 public void inputSEEMarks() {
 Scanner scanner = new Scanner(System.in);
 System.out.println("Enter SEE marks for 5 subjects:");for (int i
= 0; i < 5; i++) {
 System.out.print("Subject " + (i + 1) + " marks: ");marks[i] =
scanner.nextInt();
```

```
 }

}

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

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

import SEE.Externals;

class Main {

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

```
int numOfStudents = 2;

Externals finalMarks[] = new Externals[numOfStudents];for (int i =
0; i < numOfStudents; i++) {
 finalMarks[i] = new Externals(); finalMarks[i].inputStudentDetails();
 System.out.println("Enter CIE marks for Student " + (i + 1));
 finalMarks[i].inputCIEMarks();
 System.out.println("Enter SEE marks for Student " + (i + 1));
 finalMarks[i].inputSEEMarks();
}

System.out.println("\nDisplaying data:\n");
for (int i = 0; i < numOfStudents; i++) {
 finalMarks[i].calculateFinalMarks();finalMarks[i].displayFinalMarks();
}
}
```

```
Enter subject 1 marks:
10
Enter subject 2 marks:
10
Enter subject 3 marks:
10
Enter subject 4 marks:
10
Enter subject 5 marks:
10
Enter subject 1 marks:
10
Enter subject 2 marks:
10
Enter subject 3 marks:
10
Enter subject 4 marks:
10
Enter subject 5 marks:
10
Enter subject 1 marks:
10
Enter subject 2 marks:
10
Enter subject 3 marks:
10
Enter subject 4 marks:
10
Enter subject 5 marks:
10
Enter subject 1 marks:
10
Enter subject 2 marks:
10
Enter subject 3 marks:
10
Enter subject 4 marks:
101
Enter subject 5 marks:
10
CIE:
Subject 1 marks: 10
Subject 2 marks: 10
Subject 3 marks: 10
Subject 4 marks: 10
Subject 5 marks: 10
SEE:
Subject 1 marks: 10
Subject 2 marks: 10
Subject 3 marks: 10
Subject 4 marks: 10
Subject 5 marks: 10
CIE:
Subject 1 marks: 10
Subject 2 marks: 10
Subject 3 marks: 10
```

## Lab 7

**Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derivedclass 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=father’s age.**

```
import java.util.Scanner;

class WrongAge extends Exception {
 public WrongAge(String e) {
 super(e);
 }
}
```

```
class Father { private
 int age;

 public Father(int age) throws WrongAge {if (age
 < 0) {
 throw new WrongAge("Age cannot be negative");
 }
 this.age = age;
}

public int getAge() {
 return age;
}

}

class Son extends Father{
 private int sonAge;
 public Son(int fatherAge, int sonAge)throws WrongAge
 {
 super(fatherAge);
 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");
 }
}
```

```
this.sonAge=sonAge;
}

public int getSonAge(){
 return sonAge;
}

}

public class ExceptionDemo{
 public static void main(String args[]){

 Scanner s=new Scanner(System.in);try{
 System.out.println("enter father age");int
 fatherAge=s.nextInt();
 System.out.println("enter son age"); int
 sonAge=s.nextInt();
 Son son =new Son(fatherAge,sonAge); System.out.println("Father's
 age:"+son.getAge()); System.out.println("Son's age:"+son.getSonAge());
 }
 catch(WrongAge e){
 System.out.println("error:"+e.getMessage());
 }
 }
}
```

```
}

PS C:\Users\Rushila V\OneDrive\Desktop> javac ExceptionDemo.java
PS C:\Users\Rushila V\OneDrive\Desktop> java ExceptionDemo
enter father age
20
enter son age
30
error:Son's Age cannot be greater than father's age
PS C:\Users\Rushila V\OneDrive\Desktop> java ExceptionDemo
enter father age
-12
enter son age
10
error:Age cannot be negative
```

## 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.**

```
class BMS extends Thread {
 public void run() {
 for (int i = 1; i <= 5; i++) {
 System.out.println("BMS college of engineering");
 try {
 Thread.sleep(10000);
 } catch (InterruptedException e) {
 e.printStackTrace();
 }
 }
 }
}
```

```
 }

 }

}

}

class CS extends Thread {

 public void run() {

 for (int i = 1; i <= 5; i++) {

 System.out.println("CSE" + i);try {

 Thread.sleep(2000);

 } catch (InterruptedException e) {

 e.printStackTrace();

 }

 }

 }

}

public class ThreadMain {

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

 b1 = new BMS();

 CS c1 = new CS();

 b1.start();

 c1.start();
```

```
 }
 }
```

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac ThreadMain.java
PS C:\Users\Rushila V\OneDrive\Desktop> java ThreadMain
BMS college of engineering
CSE1
CSE2
CSE3
CSE4
CSE5
BMS college of engineering
BMS college of engineering
BMS college of engineering
BMS college of engineering
|
```

## 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 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 message
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);

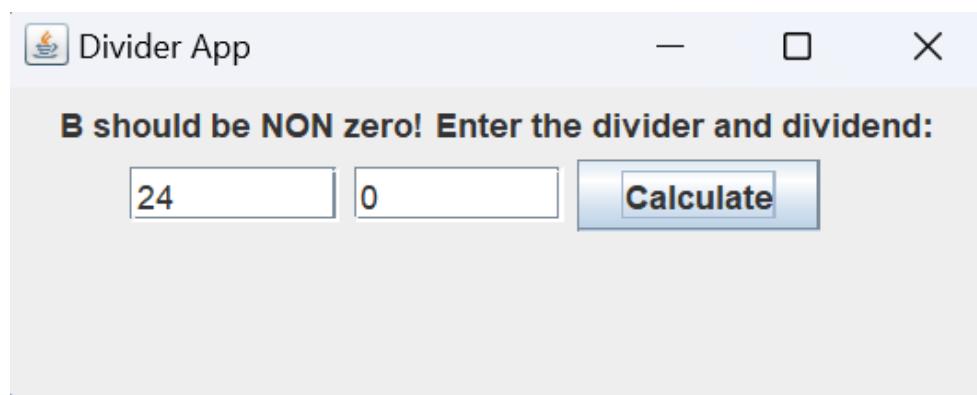
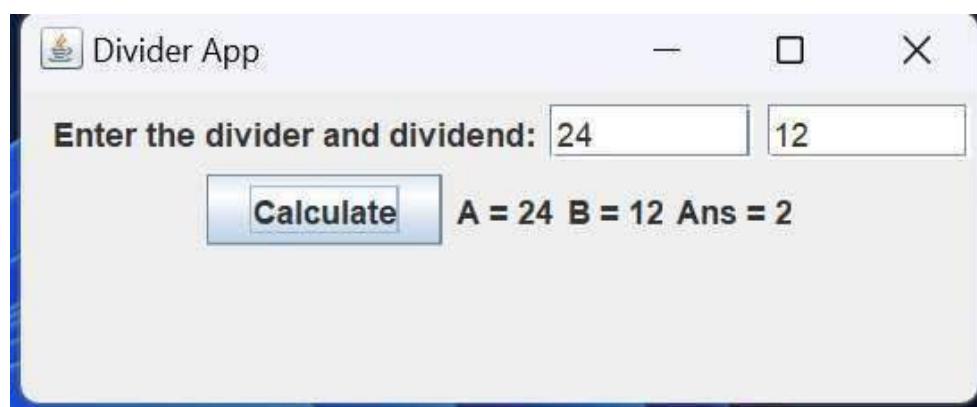
button.addActionListener(new ActionListener() { public
 void actionPerformed(ActionEvent evt) {
 try {
 int a = Integer.parseInt(ajtf.getText());int b =
 Integer.parseInt(bjtf.getText());int ans = a /
 b;
 }
 }
});
```

```
 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();
}
};
}
}
```



## Lab 10

Demonstrate Inter process Communication and deadlock.IPC

```
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("\nNotify 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");
 }
 valueSet = true;
 n++;
 }
}
```

```
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 this.n = n;
 valueSet = true;
 System.out.println("Put: " + n);
 System.out.println("\nNotify 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 < 15) {
 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 < 8) { 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.");

}

}
```

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac PCFixed.java
PS C:\Users\Rushila V\OneDrive\Desktop> java PCFixed
Press Control-C to stop.
Put: 0

Notify Consumer

Producer waiting

Got: 0

Notify Producer

Put: 1

Notify Consumer

Producer waiting

Consumed: 0
Got: 1

Notify Producer

Consumed: 1
Put: 2

Notify Consumer

Producer waiting

Got: 2
```

```
Notify Producer
```

```
Consumed: 2
Put: 3
```

```
Notify Consumer
```

```
Producer waiting
```

```
Got: 3
```

```
Notify Producer
```

```
Consumed: 3
Put: 4
```

```
Notify Consumer
```

```
Producer waiting
```

```
Got: 4
```

```
Notify Producer
```

```
Consumed: 4
Put: 5
```

```
Notify Consumer
```

```
Producer waiting
```

```
Got: 5
```

```
Got: 5
Notify Producer
Consumed: 5
Put: 6
Notify Consumer

Producer waiting

Got: 6
Notify Producer
Consumed: 6
Put: 7
Notify Consumer

Producer waiting

Got: 7
Notify Producer
Consumed: 7
Put: 8
Notify Consumer

Producer waiting
```

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

```
PS C:\Users\Rushila V\OneDrive\Desktop> javac Deadlock.java
PS C:\Users\Rushila V\OneDrive\Desktop> java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
Inside A.last
Back in mainthread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
PS C:\Users\Rushila V\OneDrive\Desktop>
```