

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



OBJECT ORIENTED JAVA PROGRAMMING

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SHIVARAJ KALLAPPA PUJARI
1BM22CS259

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

NAME: Shivaraj. K. Pujari

SUB: OOPS Lab

USN: IBM22CS259

SECTION: 3E

INDEX

1	Lab - 1	12/12/23	10
2	Lab - 2	19/12/23	10
3	Lab - 3	26/12/23	10
4	Lab - 4	2/1/24	10
5	Lab - 5	16/1/24	10
6	Lab - 6	23/1/24	10
7	Lab - 7	30/1/24	10
8	Lab - 8	06/2/24	10
9	Lab - 10	13/2/24	10
10	Lab - 9	20/2/24	10

D

*~~Palindrome~~ Shivaraj, K. Pyari

1BM22CS259

* Quadratic equation:-

→ import ~~java.util.*~~;
class ~~Quadratic~~ {
 int a, b, c;
 double ~~x1, x2, d;~~

```

void getd(){
    Scanner s0 = new Scanner (System.in);
    System.out.println ("Enter co-efficients");
    a = s0.nextInt();
    b = s0.nextInt();
    c = s0.nextInt();
}

void compute(){
    while (a == 0){
        System.out.println ("Not a quad eqn");
        System.out.println ("Enter a non-zero value of a");
        Scanner s1 = new Scanner (System.in);
        a = s1.nextInt();
    }

    d = b * b - 4 * a * c;
    if (d == 0){
        r1 = (-b) / (2 * a);
        System.out.println ("Roots are equal & real");
        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);
    r2 = Math.sqrt (-d) / (2 * a);
    System.out.println ("Root 1 = " + r1 + " + i " + r2);
    System.out.println ("Root 2 = " + r1 + " - i " + r2);
}

```

```

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

```

```

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

```

```

        System.out.println ("Name: Shivaraj KP");
        System.out.println ("USN: IBM22CS259");
    }
}
```

\Rightarrow O/p:

as Enter co-efficients :

1 -4 4

Roots are real & equal

$r_{\text{root1}} = r_{\text{root2}} = 2.0$

Name: Shivaraj. K. Pujari

USN: IBM22CS259.

6) 1 -9 4

Roots are imaginary

$r_{\text{root1}} = 1.0 + 11.732i$

$r_{\text{root2}} = 1.0 - 11.732i$

NAME : Shivaraj. K. Pujari

USN : IBM22CS259

$$\Delta > 0, -5, 6$$

Roots are real & distinct.

$$\text{Root } 1 = 3$$

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

SB
12/12/2023

19/12/23

Qn: Develop a Java program to create a class Student with members id, name, an array credits & an array marks. Include methods to accept & display details and a method to calculate CGPA of a student.

$$\rightarrow \text{CGPA} = \frac{\sum [\text{course credits}](\text{grade points})}{\sum [\text{course credits}]}$$

19/12/23

2) Main.java

```
→ import java.util.*;  
class Subject {  
    int submarks;  
    int credits;  
    int grade;  
}
```

```
class Student {  
    Subject subject[];  
    String name;
```



```
String USN;  
double CGPA;  
Scanner s;
```

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

```
void getstudentdetails()  
{
```

```
    System.out.println("Enter name & USN");  
    this.name = s.nextLine();  
    this.USN = s.nextLine();  
}
```

```
void getmarks()  
{
```

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

~~System.out.println("Enter the marks of " +
(i+1) + " Subject");~~

```
    subject[i].submarks = s.nextInt();
```

~~System.out.println("Enter the credits of " + (i+1)
+ " Subject");~~

```
subject[i].credits = s.nextInt();
subject[i].grade = (subject[i].submarks / 10) + 1;
if (subject[i].grade > 10) {
    subject[i].grade = 10;
}
if (subject[i].grade < 4) {
    subject[i].grade = 0;
}
}
```

```
void computeSGPA () {
    int totalCredits = 0;
    int sum = 0;
    for (int i=0; i<8; i++) {
        sum = sum + subject[i].grade * subject[i].credits;
        totalCredits = totalCredits + subject[i].credits;
    }
    this.SGPA = (double) sum / totalCredits;
}
```

```
public class Main {
    public static void main (String args[]) {
        Student s1 = new Student ();
        s1.getStudentDetails ();
        s1.getMarks ();
        s1.computeSGPA ();
        System.out.println ("Name :" + s1.name );
        System.out.println ("USN :" + s1.USN );
    }
}
```

System.out.println ("SGPA ; " + S1, SGPA);

→ Output :

Enter name and usn

Shivaraj K Pujari

IBM22CS259

Enter marks of 1 subject : 88

Enter credits of 1 subject : 4

Enter marks of 2 subject : 94

Enter credits of 2 subject : 4

Enter marks of 3 subject : 90

Enter credits of 3 subject : 3

Enter marks of 4 subject : 93

Enter credits of 4 subject : 3

Enter marks of 5 subject : 92

Enter credits of 5 subject : 3

Enter marks of 6 subject : 99

Enter credits of 6 subject : 1

Enter marks of 7 subject : 87

Enter credits of 7 subject : 1

Enter ~~marks~~ of 8 subject : 91

Enter credits of 8 subject : 1

NAME : Shivaraj K Pujari

USN : IBM22CS259

SGPA : 9.75

SK
19/12/2023

26/12/2023

LAB PROGRAM no:3

BooksMain:

→ import java.util.*;

class Book {

String name;

String author;

int price;

int numpages;

Book (String name, String author, int price, int numpages)

{

this.name = name;

this.author = author;

this.price = price;

this.numpages = numpages;

}

String getname() {

return name;

}

String getauthor() {

return author;

}

int

getprice() {

return price;

}

int getnum() {

return numpages;

void setname (String name){
 this.name = name;

}

void setauthor (String author){
 this.author = author;

}

void setprice (int price){
 this.price = price;

}

void setnum (int numpages){
 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;

}

public class BooksMain {

 public static void main (String args[])

 Scanner s = new Scanner (System.in);
 int n, price, numpages;

```

String name, author;
System.out.println ("Enter n");
n = s.nextInt();
Book b[];
b = new Book[n];
System.out.println ("By toString method:");
for (int i=0; i<n; i++) {
    System.out.print ("Enter Book name, author  

name, price and number of pages");
    name = s.next();
    author = s.next();
    price = s.nextInt();
    numPages = s.nextInt();
    b[i] = new Book(name, author, price, numPages);
}
for (int i=0; i<n; i++) {
    String bookDetails = b[i].toString();
    System.out.println (bookDetails);
}
System.out.println ("By get & set methods");
for (int i=0; i<n; i++) {
    System.out.println ("Enter Book details");
    name = s.
    System.out.print ("Enter Book name, author  

name, price & nor of pages");
    name = s.next();
    b[i].setname ();
    author = s.next();
    b[i].setauthor ();
}

```

laxy F54 5G

```
    price = s.nextInt();
    b[i].setprice();
    numpages = s.nextInt();
    b[i].setnum();
```

{}

```
for (int i=0; i<n; i++) {
    System.out.println("Book name :" + b[i].getname());
    System.out.println("Author name :" + b[i].getauthor());
    System.out.println("price :" + b[i].getprice());
    System.out.println("nor of pages :" + b[i].getnum());
```

{}

```
System.out.println("NAME : Shivaraj K. Pujari");
System.out.println("USN : IBM22CS259");
```

{}

Output :-

Enter n

1

By toString method

Enter book name :

Chemistry

Enter author name:

Shivaraj

Enter price :

2000

Enter nor of pages :

laxy F54 5G

Book name: Chemistry

Author name: Shivaraj

Price: 2000

nor of pages: 150

By get & set methods

Enter book name:

Biology

Enter Author name:

Anu

Enter price:

5000

Enter nor of pages:

143

Books name: Biology

Author name: Anu

Price: 5000

nor of pages: 143

NAME: Shivaraj K. Pujari

OSN: IBM22CS259

26/10/23

On : Develop a Java program to create an abstract class named Shape that contains 2 integers & an empty method named printArea(). Provide 3 classes named Rectangle, Triangle & 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.*;
class InputScanner {
    Scanner s;
    InputScanner() {
        s = new Scanner (System.in);
    }
}
```

```
abstract class Shape extends InputScanner {
    double a, b;
    abstract void getinput();
    abstract void displayarea();
}
```

```
class Rectangle extends Shape {
    void getinput()
```

```
InputScanner sc = new InputScanner();
System.out.println ("Enter the dimensions of
Rectangle:");
a = sc.s.nextDouble();
b = sc.s.nextDouble();
```

```
a = sc.s.nextDouble();
b = sc.s.nextDouble();
```

```
void displayarea() {  
    System.out.println("Area of Rectangle is : " +  
        (a * b));  
}
```

```
class Triangle extends Shape {
```

```
    void getinput()
```

```
{
```

```
    InputScanner sc = new InputScanner();
```

```
    a = sc.s.nextDouble();
```

```
    b = sc.s.nextDouble();
```

```
}
```

```
    void displayarea()
```

```
{
```

```
    System.out.println("Area of Triangle is : " +  
        (0.5 * a * b));
```

```
}
```

```
}
```

```
class Circle extends Shape {
```

```
    void getinput()
```

```
{
```

~~InputScanner sc = new InputScanner();~~~~System.out.println("Enter the Radius of circle");~~~~a = sc.s.nextDouble();~~~~{~~

void displayarea()

{ double pi = 3.14;

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

}

public class AreaMain

{ public static void main (String args [])

Rectangle r = new Rectangle();

r. getinput();

r. displayarea();

Triangle t = new Triangle();

t. getinput();

t. displayarea();

Circle c = new Circle();

c. getinput();

c. displayarea();

System.out.println("NAME: Shivaraj. K. Pujari");

~~System.out.println("USN: 1BM122(S254)");~~

3

}

3

Output:

Enter the dimensions of rectangle

2

3

Area of Rectangle : 6.0

Enter the dimensions of Triangle

4

5

Area of Triangle : 20.0

Enter the & Radius of circle

4

Area of circle : 50.24

NAME: Shivaraj. K. Pujari

OSNL : IBM 220S259

SB
21/12/24

9/11/24

LAB PROGRAM-5

Date _____
Page _____

Qn: Develop a Java program which has class Bank, Account & savings account which extends Account.

```
Class import java.util.*;
Class Account {
    String name;
    String int accno;
    String acctype;
    double balance;
    Account (String name, int accno, String acctype,
             double balance) {
        this.name = name;
        this.accno = accno;
        this.acctype = acctype;
        this.balance = balance;
    }
    void deposit (double amt) {
        balance += amt;
    }
    void withdraw (double amt) {
        if ((balance - amt) >= 0)
            balance -= amt;
        else
            System.out.println ("Insufficient balance");
    }
}
```

```
void display () {  
    System.out.println ("Name: " + name + "accno: " +  
        accno + "account-type: " + acctype + "balance: " +  
        balance);  
}
```

```
class Savingaccount extends Account {  
    private static double r = 5;  
    int t = 1;  
    Savingaccount (String name, int accno, double  
        balance) {  
        super (name, accno, "savings", balance);  
    }
```

```
    void interest () {  
        balance += (balance * t * r) / 100;  
        System.out.println ("Balance: " + balance);  
    }
```

```
class Curacc extends Account {  
    private double minbal = 500;  
    private double charge = 50;  
    Curacc (String name, int accno, double balance) {  
        super (name, accno, "current", balance);  
    }
```

```

void checkmin() {
    if (balance < minbal) {
        System.out.println("balance is less than min
                           balance, service charges imposed: " + charge);
        balance -= charge;
        System.out.println("balance is " + balance);
    }
}

```

```

class Bank {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name:");
        String name = sc.nextLine();
        System.out.println("enter the type of account:");
        String type = sc.nextLine();
        System.out.println("Enter the acc no:");
        int accno = sc.nextInt();
        System.out.println("Enter the balance:");
        double balance = sc.nextDouble();
        int ch;
        double amt1, amt2;
        Account ac = new Account(name, accno, type,
                                 balance);
        Saving account sa = new Savingaccount(name,
                                              accno, balance);
    }
}

```

```
Customer ca = new Customer (name, accno, balance);  
System.out.println ("NAME : Ghiraraj K Pujari");  
System.out.println ("OSN : IBM22G259");  
while (true) {  
    if (ac.accType.equals ("savings"))
```

```
        System.out.println ("In MENU In 1. deposit  
2. withdraw 3. compute interest 4. display  
5. exit ");
```

```
        System.out.print ("Enter the choice ");  
        ch = sc.nextInt ();  
        switch (ch) {
```

```
            case 1: System.out.print ("Enter the  
amount ");
```

```
            amt1 = sc.nextInt ();  
            sa.deposit (amt1);  
            break;
```

```
            case 2: System.out.print ("Enter the  
amount ");
```

```
            amt2 = sc.nextInt ();  
            sa.withdraw (amt2);  
            sa.checkmin ();  
            break;
```

```
            case 3: sa.interest ();  
            break;
```

```
            case 4: sa.display ();  
            break;
```

```
            case 5: System.exit (0);
```



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

}

else if

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

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

ch = sc.nextInt();

switch (ch){

case 1: System.out.println ("Enter amount");

amt1 = sc.nextInt();

ca.deposit(amt1);

break;

case 2: System.out.println ("Amount to be
withdraw");

amt2 = sc.nextInt();

ca.withdraw(amt2);

ca.checkmin();

break;

case 3: ca.display();

break;

case 4: System.exit(0);

}

}

}

Output:

Enter the name:

Shiv

Enter the type of account:

savings

Enter the account number:

143

Enter the balance:

50000

NAME: Shivaraj. K. Pugari

USN: IBM22CS259

MENU

1. deposit
2. withdraw
3. compute interest
4. display
5. Exit

Enter the choice

2

Enter amount

49600

Balance is less than minimum balance, insufficient balance

MENU

1. deposit
2. withdraw
3. compute interest
4. display
5. exit

Enter the choice

4

Name: Shiv accno: 143 acc-type: savings balance: 400.0

10/1/2024



LAB-6

23/1/24

PACKAGES

1) CIE - Student.java

→ package CIE;

```
import java.util.*;  
public class Student {  
    protected String name = new String();  
    protected String USN = new String();  
    protected int sem;  
    public void inputStudDetails() {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Name:");  
        name = s.nextLine();  
        System.out.println("USN:");  
        USN = s.nextLine();  
        System.out.println("Sem:");  
        sem = s.nextInt();  
    }  
    public void display() {  
        System.out.println("Student Details");  
        System.out.println("Name: " + name);  
        System.out.println("USN: " + USN);  
        System.out.println("Sem: " + sem);  
    }  
}
```

2) CIE - Internals.java

→ package CIE;

~~import java.util.*;~~~~public class Internals extends Student {
 protected int marks[] = new int[5];
}~~

by F54 5G

```
public void inputcie() {
    Scanner s = new Scanner (System.in);
    System.out.println("Enter the marks in 5 subjects:");
    for (int i=0; i<5; i++) {
        marks[i] = s.nextInt();
    }
}
```

3) SEE - External.java

→ package SEE;

```
import CIE.Internals;
import java.util.*;
public class External extends Internals {
    protected int marks[];
    protected int finalmarks[];
    public External() {
        marks = new int[5];
        finalmarks = new int[5];
    }
}
```

```
public void inputsee() {
    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 cfmarks() {
    for (int i = 0; i < 5; i++) {
        finalmarks[i] = marks[i]/2 + supermarks[i];
    }
}

```

```

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

```

4) Student Main.java

```

import SEE.Externals;
public class StudentMain {
    public static void main (String args[]) {
        int n = 1;
        Externals finalmarks[] = new Externals[n];
        for (int i = 0; i < n; i++) {
            finalmarks[i] = new Externals();
            finalmarks[i].inputstuddetails();
            System.out.println("Enter CIE marks:");
            finalmarks[i].inputcie();
            System.out.println("Enter SEE marks:");
            finalmarks[i].inputsee();
        }
    }
}

```

```
System.out.println ("Displaying data:");
for (int i = 0; i < n; i++) {
    finalmarks[i].cmarks();
    finalmarks[i].displaymarks();
}
```

→ OUTPUT:-

Enter Student's Name: Shivaraj K Pujari

USN : 1BM22CS259

Sem : 3

Enter CIE marks:

50 50 48 48 48

Enter SEE marks:

Subject 1 marks : 98

Subject 2 marks : 96

Subject 3 marks : 100

Subject 4 marks : 98

Subject 5 marks : 96

Displaying Data:

NAME : Shivaraj . K. Pujari

USN : 1BM22CS259

Sem : 3

Subject 1 : 99

Subject 2 : 98

Subject 3 : 98

Subject 4 : 97

Subject 5 : 96

100 / 100

LAB - PROGRAM - 6

* Strings

1) Shivu

Shivraj
Ansh

2) length = 5

Ht Shiv

3) a = 333

4) Bmsce

5) 65

66

67

abc

6) true

false

false

true

7) substring is matched

8) true

true

9) false

true

10) true
false

11) Apple ball cat dog ed free gun hen ice jug
kite lift man net orange parrot queen ring star tru

12) 1 2 3 4 5 6 7 8 9 10

13) This is a fest, This is, too

14) hello world

15) Commge

16) Hello friends

17) Student 1

name: anu
Reg no : 123
Sem : 3
CGPA : 9.6

Student 2

name: Shiva
Reg no: 143
Sem : 3
CGPA : 9.6



18) char at 3 is 'x'

abc

reverse : inohd

19) Eagle is flying

Eagle makes a sound

Hawk is flying

Hawk makes a sound

20) Caria : 28.26

Rperi : 18.84

Tara : 40

Tperi : 29.83

NAME : Shivaraj , K. Pujari
USN : IBM 22C5259

* Generics

Qn. Write a Java program to create a generic class Stack which hold 5 integers & 5 double values

→ import java.util.*;

```
class Stack<E> {
```

```
    E stk[];
```

```
    int top;
```

```
    int size = 10;
```

```
    Stack() {
```

```
        stk = (E[]) new Object[size];
```

```
        top = -1;
```

```
    void push(E item) {
```

```
        if (top == size - 1) {
```

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

```
        } else {
```

```
            stk[++top] = item;
```

```
        }
```

```
    if (top < 0) {
```

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

```
        return null;
```

```
    }
```

```
    else {
```

```
        return stk[top - 1];
```

```
}
```



```
public class TestStack {  
    public static void main (String args[]) {  
        Stack < Integer > s1 = new Stack < Integer > ();  
        Stack < Double > s2 = new Stack < Double > ();  
        Scanner s = new Scanner (System.in);  
        System.out.println ("Enter elements in integer stack");  
        for (int i=0 ; i<5 ; i++) {  
            int n = s.nextInt();  
            s1.push (n);  
        }  
  
        System.out.println ("Enter elements in Double stack");  
        for (int i=0 ; i<5 ; i++) {  
            double m = s.nextDouble();  
            s2.push (m);  
        }  
  
        System.out.println ("Elements of s1");  
        for (int i=0 ; i<5 ; i++) {  
            System.out.println (s1.pop());  
        }  
  
        System.out.println ("Elements of s2");  
        for (int i=0 ; i<5 ; i++) {  
            System.out.println (s2.pop());  
        }  
        s.close();  
    }  
}
```

Output:-

+ - 6 9 + Enter elements in integer stack

1 3 6 9 12

Enter elements in Double stack

2 4 5 7 8

Elements of s1:

12

9

6

3

1

Elements of s2:

8.0

7.0

5.0

4.0

2.0

NAME: Shivaraj K. Pujari

USN: IBM 22 CS 259

54 5G



30/1/24

LAB - 7

Date _____
Page _____

Exception Handling

```
→ import java.util.*;  
class Wrongage extends Exception {  
    public Wrongage (String s) {  
        super(s);  
    } }
```

```
class Father {  
    int page;  
    Father () throws Wrongage {  
        System.out.println ("Enter father's age :");  
        Scanner s = new Scanner (System.in);  
        page = s.nextInt();  
        if (page < 0)  
            throw new Wrongage ("Age cannot be  
negative");  
    }  
    void display () {  
        System.out.println ("Father's age is : " + page);  
    } }
```

```
class Son extends Father {  
    int sage;  
    Son () throws Wrongage {  
        System.out.println ("Enter son's age");  
        Scanner s = new Scanner (System.in);  
        sage = s.nextInt();  
    } }
```

```
if (sage >= fage) {  
    throw new Wrongage ("Son's age cannot be  
    greater than father's age");  
}
```

```
else-if (sage < 0) {  
    throw new Wrongage ("Age cannot be  
    negative");  
}
```

```
}
```

```
void display() {  
    System.out.println ("Son's age is: " + sage);  
}
```

```
}
```

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

```
        try {
```

```
            Son s = new Son();
```

```
            s.display();
```

```
            s.sdisplay();
```

```
        } catch (Wrongage e) {
```

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

```
            System.out.println ("NAME: Shivaraj K. Pujari");
```

~~```
 System.out.println ("USN: 1BM22CS259");
```~~

```
}
```

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

Output : Enter father's age

50

Enter son's age

60

Wrongage : Son's age cannot be greater  
than father's age

Enter father's age

90

Enter son's . age

50

Father's age is : 90

Son's age is : 50

NAME : shivaraj . K . Pujari

USN : IBM 22 CS 259

30/1/2024

6/2/24

LAB-8.

Qn: Write a program which creates 2 threads, one thread displaying "BMS college of engineering" once every ten seconds and another displaying "CSE" once every 2 seconds.

→ class CS extends Thread {

public void run() {

for (int i = 1; i <= 20; i++) {

try {

Thread.sleep(10000);

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

Thread.sleep(10000);

}

catch (InterruptedException e) {

System.out.println("error");

}

}

class IS extends Thread {

public void run() {

for (int i = 1; i <= 20; i++) {

try {

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

Thread.sleep(2000);

}

catch (InterruptedException e) {

System.out.println("error");

}

}

```
public class Tmain {
 public static void main (String args []) {
 CS c = new CS();
 c.start();
 IS i = new IS();
 i.start();
 }
}
```

### OUTPUT:

BMSCE 1

CS 1

:

CS 5

BMSCE 2

CS 6

:

CS 10

BMSCE 3

CS 11

:

CS 15

BMSCE 4

CS 16

:

CS 20

~~BMSCE 5~~

:

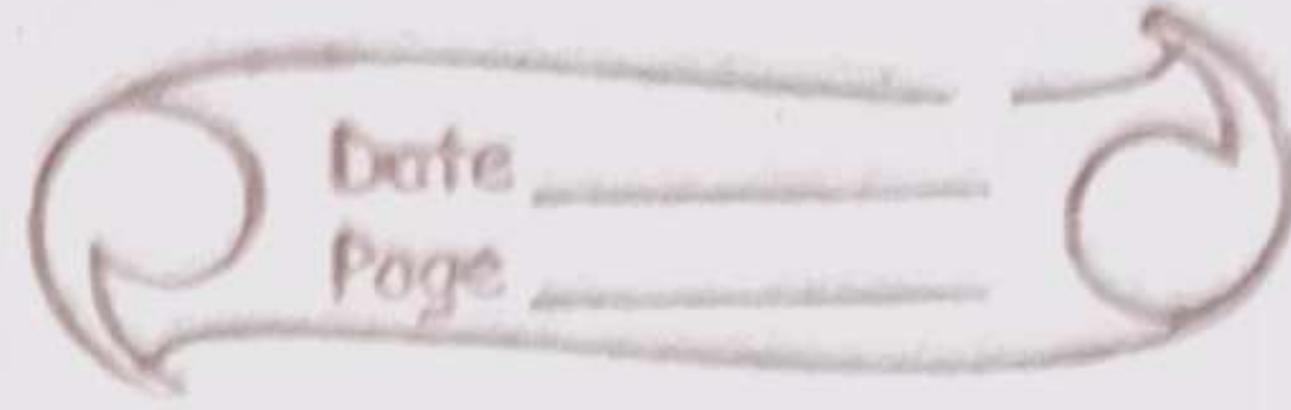
~~BMSCE 20~~

~~6/2/2024~~

4.5G

20/2/21

## LAB-9



Qn: Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 & 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 NumberFormat Exception. 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() {
 JFrame jfrm = new JFrame("Divide App");
 jfrm.setSize(275, 150);
 jfrm.setLayout(new FlowLayout());
 jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 JLabel jlab = new JLabel("Enter the divisor and
divident:");
 JTextField d1tf = new JTextField(8);
 JTextField d2tf = 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(aJff);
jfrm.add(bJff);
jfrm.add(Button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(ansLab);
```

```
ActionListener i = new ActionListener(){
```

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

```
aJff.addActionListener(i);
```

```
bJff.addActionListener(i);
```

```
Button.addActionListener(new ActionListener(){
```

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

```

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

```

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

```

```
 int ans = a/b;

```

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

```

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

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

}

};

```
jfrm.setVisible(true);
```

}

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

~~SwingUtilities.invokeLater(new Runnable(){~~~~public void run(){~~~~new SwingDemo();~~

}

};

}

};

Output: NAME: SHIVARAJ K PUSARI  
USN: IBM22CS259

Enter the divisor and dividend:

[ 20 ]    [ 2 ]

calculate    A=20    B=2    Ans=10

\* Functions & definitions:

1) JFrame: The javax.swing.JFrame class is a type of container which inherits the java.awt.class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

2) getSize(): Sets the size of Dimension object to the specified width & height. This method is included for completeness, to parallel the getSize() method of component.

3) setLayout(): After a container has been created, you can set its layout manager using setLayout method.

4) getDefaultCloseOperation(): This method is used to specify one of several options for close button.

5) Label : 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.

6) ActionListener () : The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent.

7) Action Performed () : This method is invoked automatically whenever you click on registered component.

8) SwingUtilities : The SwingUtilities class has 2 important static methods, invoke and wait() and invokeLater() to use to put references to blocks of code onto event queue.

9) invokeLater () : It can be used to perform a task asynchronously in AWT Event dispatcher thread.

10) setVisible () : It is a method that has return type boolean. It sets the currently made attributes to be visible on the screen GUI application mainly Java Swing

28  
20/2/2024

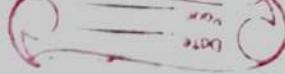
13/12/24

## LAB - 10

\* Demonstrate Interprocess Communication.  
On: Implementation of a producer & consumer.

```
→ import java.util.*;
class Q{
 int n;
 boolean valueset = false;
 synchronized int get(){
 while(!valueset)
 try {
 System.out.println("Consumer Waiting\n");
 wait();
 }
 catch(InterruptedException e){
 System.out.println("Interrupted Exception
caught in");
 }
 System.out.println("Got; " + n);
 valueset = false;
 System.out.println("Notify Producer\n");
 notify();
 return n;
 }
 synchronized void put(int n){
 while(valueset)
 try {
 System.out.println("Producer waiting\n");
 wait();
 }
 }
```

F54 50



```

catch (InterruptedException e) {
 System.out.println("Interrupted exception\n");
}
this. n = n;
valueset = true;
System.out.println("Put: " + n);
System.out.println("Intimate consumer\n");
notify();
}
}

```

```

class Producer implements Runnable {
 Queue q;
 Producer(Q q) {
 this. q = q;
 new Thread(this, "Producer").start();
 }
 public void run() {
 int i = 0;
 while (i < 5) {
 q.put(i++);
 }
 }
}

```

```

class Consumer implements Runnable {
 Queue q;
 Consumer(Q q) {
 this. q = q;
 new Thread(this, "Consumer").start();
 }
}

```

```
public void run(){
 int i=0;
 while (i<5){
 int r=q.get();
 System.out.println ("Consumed : "+r);
 i++;
 }
}
```

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

Output:

Put : 0  
Intimate consumer

Producer waiting

Press Ctrl-C to stop

Crot : 0  
Intimate producer

Part : 1  
Intimate Consumer  
Producer waiting

Consumed : 0

Crot : 1  
Intimate Producer

Consumed : 1

Part : 2  
Intimate Consumer

Producer waiting

Crot : 2

Intimate Producer

Consumed : 2

Part : 3  
Intimate Consumer  
Producer waiting

Crot : 3

Intimate Producer

Consumed : 3

Part : 4  
Intimate Consumer

Crot : 4

Intimate Producer

Consumed : 4

13/2/14

## \* Deadlock

Only Demonstrate Deadlocks.

→ 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 Deadlocks 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 mainthread");
```

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

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

```
System.out.println("Back in otherthread");
```

}

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

```
new Deadlock();
```

\*

```
System.out.println("NAME: Shivaraj K Pujari");
```

\*

```
System.out.println("OSN: IBM 22CS259");
```

}

Output:

Main Thread entered A.foo

Racing Thread entered B.bar

Main Thread trying to call B.last()

Inside A.last

Back in main thread

Racing Thread trying to call A.last()

Inside A.last

Back in other thread.

~~NAME: Shivraj K Pujari~~

~~USN: 1BM22CS259~~

~~28  
13/2/2023~~

**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.*;
class Quadratic
{
int a, b, c;
double r1, r2, d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the coefficients of a,b,c");
a = s.nextInt();
b = s.nextInt();
c = s.nextInt();
}
void compute()
{
while(a==0)
{
System.out.println("Not a quadratic equation");
System.out.println("Enter a non zero value for a:");
Scanner s = new Scanner(System.in);
a = s.nextInt();
}
d = b*b-4*a*c;
if(d==0)
{
r1 = (-b) / (2*a);
System.out.println("Roots are real and equal");
System.out.println("Root1 = Root2 = " + r1);
}
else if(d>0)
{
r1 = ((-b)+(Math.sqrt(d))) / (double) (2*a);
r2 = ((-b)-(Math.sqrt(d))) / (double) (2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root1 = " + r1 + " Root2 = " + r2);
}
else if(d<0)
{
System.out.println("Roots are imaginary");
r1 = (-b) / (2*a);
r2 = Math.sqrt(-d) / (2*a);
System.out.println("Root1 = " + r1 + " + i" + r2);
System.out.println("Root1 = " + r1 + " - i" + r2);
}
}
```

```

}

class Quadraticmain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();
q.compute();
System.out.println("NAME:Shivaraj K Pujari");
System.out.println("USN:1BM22CS259");
}
}

```

**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.**

```

//NAME:Shivaraj K Pujari

import java.util.*;
class Subject{
 int submarks;
 int credits;
 int grade;
}

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

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

 void getstudentdetails()
 {
 System.out.println("enter name and usn");
 this.name=s.nextLine();
 this.USN=s.nextLine();
 }
}
```

```

void getmarks()
{
int i;
for(i=0;i<8;i++)
{
 System.out.println("Enter the marks of " + (i+1)+ " Subject");
 subject[i].submarks=s.nextInt();
 System.out.println("Enter the credits of " + (i+1)+ " Subject");
 subject[i].credits=s.nextInt();
 subject[i].grade=(subject[i].submarks/10)+1;
 if(subject[i].grade>10){
 subject[i].grade=10;
 }
 if(subject[i].grade<4){
 subject[i].grade=0;
 }
}
}

void computeSGPA()
{
 int totalcredits=0;
 int sum=0;
 int i;
 for(i=0;i<8;i++)
 {
 sum=sum+subject[i].grade * subject[i].credits;
 totalcredits=totalcredits+subject[i].credits;
 }
 this.SGPA=(double) sum/totalcredits;
}

public class Main
{
 public static void main(String args[])
 {
 Student s1=new Student();
 s1.getstudentdetails();
 s1.getmarks();
 s1.computeSGPA();
 System.out.println("NAME:"+s1.name);
 System.out.println("USN:"+s1.USN);
 System.out.println("SGPA:"+s1.SGPA);
 System.out.println("NAME:Shivaraj K Pujari");
 System.out.println("USN:1BM22CS259");

 }
}

```

**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.*;

class Book{
String name;
String author;
int price;
int numpages;

Book(String name, String author, int price, int numpages)
{
 this.name=name;
 this.author=author;
 this.price=price;
 this.numpages=numpages;
}

String getname()
{
 return name;
}
String getauthor()
{
 return author;
}
int getprice()
{
 return price;
}
int getnumpages()
{
 return numpages;
}

void setname(String name)
{
 this.name=name;
}
void setauthor(String author)
{
 this.author=author;
}
void setprice(int price)
{
 this.price=price;
```

```

}

void setnumpages(int numpages)
{
 this.numpages=numpages;
}

public String toString()
{
 String name,author;
 String 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;
}
};

public class BooksMain{
 public static void main(String args[])
 {
 Scanner s=new Scanner(System.in);
 int n,price,numpages;
 String name, author;
 System.out.println("Enter n");
 n=s.nextInt();
 Book b[];
 b=new Book[n];
 System.out.println("By toString method");
 for(int i=0;i<n;i++)
 {
 System.out.println("Enter the book name");
 name=s.next();
 System.out.println("Enter the author name");
 author=s.next();
 System.out.println("Enter the price of book");
 price=s.nextInt();
 System.out.println("Enter the number of pages");
 numpages=s.nextInt();
 b[i]=new Book(name,author,price,numpages);
 }
 for(int i=0;i<n;i++)
 {
 String Bookdetails=b[i].toString();
 System.out.println(Bookdetails);
 }
 System.out.println("*****");
 System.out.println("By get and set methods");
 for(int i=0;i<n;i++)
 {

 System.out.println("enter book name:");

```

```

 name=s.next();
 b[i].setname(name);

 System.out.println("enter author name:");
 author=s.next();
 b[i].setauthor(author);

 System.out.println("enter book price:");
 price=s.nextInt();
 b[i].setprice(price);

 System.out.println("enter number of pages:");
 numpages=s.nextInt();
 b[i].setnumpages(numpages);
 }
 for(int i=0;i<n;i++)
 {
 System.out.println("Book name:"+b[i].getname());
 System.out.println("Author name:"+b[i].getauthor());
 System.out.println("Book price:"+b[i].getprice());
 System.out.println("Number pages:"+b[i].getnumpages());
 }
 System.out.println("NAME:Shivaraj K Pujari");
 System.out.println("USN:1BM22CS259");
}
}

```

**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 ofthe given shape.**

```

import java.util.Scanner;

class InputScanner{
 Scanner s;
 InputScanner(){
 s = new Scanner(System.in);
 }
}
abstract class Shape extends InputScanner{
 double a;
 double b;

 abstract void getInput();
 abstract void displayArea();
}

class Rectangle extends Shape{

```

```

 void getInput(){
 InputScanner is = new InputScanner();
 System.out.println("Enter the length and breadth of the
rectangle :");
 a = is.s.nextDouble();
 b = is.s.nextDouble();
 }
 void displayArea(){
 System.out.println("The area of the rectangle is :" +(a*b));
 }
 }

 class Triangle extends Shape{

 void getInput(){
 InputScanner is = new InputScanner();
 System.out.println("Enter the base and height of the triangle
:");
 a = is.s.nextDouble();
 b = is.s.nextDouble();
 }
 void displayArea(){
 System.out.println("The area of the triangle is :" +(a*b*0.5));
 }
 }

 class Circle extends Shape{

 void getInput(){
 InputScanner is = new InputScanner();
 System.out.println("Enter radius of the Cirlce :");
 a = is.s.nextDouble();
 }
 void displayArea(){
 System.out.println("The area of the Circle is :" +(3.14*a*a));
 }
 }

 public class AbstractMain{
 public static void main(String args[]){
 System.out.println("HI");
 Rectangle rect = new Rectangle();
 rect.getInput();
 rect.displayArea();

 Triangle triangle = new Triangle();
 triangle.getInput();
 triangle.displayArea();
 }
 }
}

```

```

 Circle circle = new Circle();
 circle.getInput();
 circle.displayArea();
 System.out.println("NAME : SHIVARAJ K PUJARI");
 System.out.println("USN : 11BM22CS259");
 }
}

```

**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.*;
class Account{
 String name;
 int accno;
 String acctype;
 double balance;
 Account(String name,int accno,String acctype,double balance)
 {
 this.name=name;
 this.accno=accno;
 this.acctype=acctype;
 this.balance=balance;
 }
 void deposit(double amt){
 balance+=amt;
 }
 void withdraw(double amt){
 if((balance-amt)>=0){
 balance-=amt;
 }
 else{
 System.out.println("Insufficient balance");
 }
 }
}

```

```

 void display(){
 System.out.println(" Name:" + name + " accnor:" + accno + " "
 account_type:" + acctype + " balance:" + balance);
 }
 }
 class Savingaccount extends Account{
 private static double rate=5;
 private double minbal=500;
 Savingaccount(String name,int accno,double balance) {
 super(name,accno,"savings",balance);
 }
 void interest(){
 balance+=balance*(rate)/100;
 System.out.println("Balance:" + balance);
 }
 void checkmin(){
 if(balance<minbal){
 System.out.println("balance is less than minimum
 balance,insufficient balance");
 }
 }
 }

 class Curracc extends Account{
 private double minbal=500;
 private double charge=50;
 Curracc(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:" + charge);
 balance-=charge;
 System.out.println("Balance is:" + balance);
 }
 }
 }

 class Bank{
 public static void main(String args[])
 {
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the name:");
 String name=sc.nextLine();
 System.out.println("Enter the type of account:");
 String type=sc.nextLine();
 System.out.println("Enter the account number:");
 int accno=sc.nextInt();
 System.out.println("Enter the Balance:");
 double balance=sc.nextDouble();
 int ch;

```

```

double amt1,amt2;
Account ac=new Account(name,accno,type,balance);
Savingaccount sa=new Savingaccount(name,accno,balance);
Curracc ca=new Curracc(name,accno,balance);
System.out.println("NAME:Shivaraj K Pujari");
System.out.println("USN:1BM22CS259");
while(true)
{
 if(ac.acctype.equals("savings"))
 {
 System.out.println("\nMENU\n 1.deposit 2.withdraw
3.compute interest 4.display 5.exit");
 System.out.println("Enter the choice");
 ch=sc.nextInt();
 switch(ch)
 {
 case 1:System.out.println("Enter the
amount");
 amt1=sc.nextInt();
 sa.deposit(amt1);
 break;
 case 2:System.out.println("Enter the
amount");
 amt2=sc.nextInt();
 sa.withdraw(amt2);
 sa.checkmin();
 break;
 case 3:sa.interest();
 break;
 case 4:sa.display();
 break;
 case 5:System.exit(0);

 default:System.out.println("Inavlid input");
 break;
 }
 }
 else{
 System.out.println("\n MENU \n 1.deposit 2.withdraw
3.display ");
 System.out.println("Enter the choice");
 ch=sc.nextInt();
 switch(ch){
 case 1:System.out.println("Enter the
amount:");
 amt1=sc.nextInt();
 ca.deposit(amt1);
 break;
 case 2:System.out.println("Amount to be
withdraw");
 amt2=sc.nextInt();
 ca.withdraw(amt2);
 }
 }
}

```

```

 ca.checkmin();
 break;
 case 3:ca.display();
 break;
 case 4:System.exit(0);
 }
}
System.out.println("NAME:Shivaraj K Pujari");
System.out.println("USN:1BM22CS259")

}
}

```

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

```

package CIE;
import java.util.*;
public class Student{
 protected String usn=new String();
 protected String name =new String();
 protected int sem;

 public void inputStudentDetails(){
 Scanner s=new Scanner(System.in);
 this.usn=s.nextLine();
 this.name=s.nextLine();
 this.sem=s.nextInt();
 }

 public void displayStudentDetails(){
 System.out.println(this.usn+" "+this.name+" "+this.sem);
 }
}

package CIE;
import java.util.Scanner;
public class Internals extends Student{

```

```

protected int marks[] = new int[5];
public void inputCIEmarks() {
 Scanner s = new Scanner(System.in);
 for (int i=0; i<5; i++) {
 marks[i] = s.nextInt();
 }
}

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

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

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

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

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

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

import SEE.*;

class Main1 {
 public static void main(String args[]) {
 int num=2;
 Externals finalMarks[] = new Externals[num];
 for (int i=0; i<num; i++) {
 finalMarks[i] = new Externals();
 }
 }
}

```

```

 finalMarks[i].inputStudentDetails();
 System.out.println("Enter CIE marks");
 finalMarks[i].inputCIEmarks();
 System.out.println("Enter SEE marks");
 finalMarks[i].inputSEEmarks();
 }
 System.out.println("Displaying Data:\n");
 for(int i=0;i<num;i++) {
 finalMarks[i].calculateFinalMarks();
 finalMarks[i].displayFinalMarks();
 }
 System.out.println("NAME:Shivaraj K Pujari");
 System.out.println("USN:1BM22CS259");
}

}
}

```

**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.*;
class Wrongage extends Exception{
 public Wrongage(String s)
 {
 super(s);
 }
}
class Father{
 int fage;
 Father () throws Wrongage
 {
 System.out.println("enter father's age");
 Scanner s=new Scanner(System.in);
 fage=s.nextInt();
 if(fage<0)
 {
 throw new Wrongage("Age cannnot be negative");
 }
 }
 void display()
 {
 System.out.println("Father's Age is :" + fage);
 }
}
class Son extends Father{

```

```

int sage;
Son () throws Wrongage
{
 System.out.println("enter son's age");
 Scanner s=new Scanner(System.in);
 sage=s.nextInt();
 if(sage>fage)
 {
 throw new Wrongage("Son's age cannot be greater than
 father's age");
 }
 else if(sage==fage)
 throw new Wrongage("Age cannnot be same");
 }
 else if(sage<0)
 {
 throw new Wrongage("Age cannnot be negative");
 }
}
void sdisplay()
{
 System.out.println("Son's Age is :" + sage);
}
}
class Spmain{
 public static void main(String args[])
 {
 try{
 Son s=new Son();
 s.display();
 s.sdisplay();
 }
 catch(Wrongage e){
 System.out.println(e);
 }
 System.out.println("NAME: Shivaraj K Pujari");
 System.out.println("USN:1BM22CS259");
 }
}

```

**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 CS extends Thread
{
public void run()
{

```

```

for(int i=1; i<=20; i++) {
 try{
 System.out.println("BMS College of Engineering" + i);
 Thread.sleep(10000);
 }
 catch(InterruptedException e){
 System.out.println("thread error");
 }
}
}

class IS extends Thread
{
public void run()
{
for(int i=1; i<=20; i++) {
 try{
 System.out.println("Computer Science " + i);
 Thread.sleep(2000);
 }
 catch(InterruptedException e){
 System.out.println("thread error");
 }
}
}

public class Tmain {
public static void main(String args[])
{
CS c1 = new CS();
c1.start();

IS i1 = new IS();
i1.start();
System.out.println("NAME:SHIVARAJ KALLAPPA PUJARI");
System.out.println("USN :1BM22CS259");
}
}

```

**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();

 // add in order :
 jfrm.add(err); // to display error bois
 jfrm.add(jlab);
 jfrm.add(ajtf);
 jfrm.add(bjtf);
 jfrm.add(button);
 jfrm.add(alab);
 jfrm.add(blab);
 jfrm.add(anslab);

 ActionListener l = new ActionListener() {
 public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text field");
 }
 };
 ajtf.addActionListener(l);
 bjtf.addActionListener(l);

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

```

```

 int ans = a/b;

 alab.setText("\nA = " + a);
 blab.setText("\nB = " + b);
 anslab.setText("\nAns = "+ ans);
 }
 catch(NumberFormatException e){
 alab.setText("");
 blab.setText("");
 anslab.setText("");

 err.setText("Enter Only Integers!");
 }
 catch(ArithmetricException e){
 alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("B should be NON zero!");
 }
}
);

// display frame
jfrm.setVisible(true);
}

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

 System.out.println("NAME:SHIVARAJ KALLAPPA PUJARI");
 System.out.println("USN:1BM22CS259");
}
}
}

```

## **10)Demonstrate Inter process Communication and deadlock**

### **a)Inter process Communication**

```

class Q {
 int n;
 boolean valueSet = false;

 synchronized int get() {
 while (!valueSet)

```

```

 try {
 System.out.println("\nConsumer waiting\n");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 System.out.println("Got: " + n);
 valueSet = false;
 System.out.println("\nIntimate Producer\n");
 notify();
 return n;
 }

synchronized void put(int n) {
 while (valueSet)
 try {
 System.out.println("\nProducer waiting\n");
 wait();
 } catch (InterruptedException e) {
 System.out.println("InterruptedException caught");
 }
 this.n = n;
 valueSet = true;
 System.out.println("Put: " + n);
 System.out.println("\nIntimate Consumer\n");
 notify();
}
}

class Producer implements Runnable {
 Q q;

 Producer(Q q) {
 this.q = q;
 new Thread(this, "Producer").start();
 }

 public void run() {
 int i = 0;
 while (i < 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 < 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.");
 System.out.println("NAME:SHIVARAJ K PUJARI");
 System.out.println("USN:1BM22CS259");
 }
}

```

### b)Deadlock

```

class A {
 synchronized void foo(B b) {
 String name = Thread.currentThread().getName();
 System.out.println(name + " entered A.foo");
 try {
 Thread.sleep(1000);
 } catch (Exception e) {
 System.out.println("A Interrupted");
 }
 System.out.println(name + " trying to call B.last()");
 b.last();
 }

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

class B {
 synchronized void bar(A a) {
 String name = Thread.currentThread().getName();
 System.out.println(name + " entered B.bar");
 try {
 Thread.sleep(1000);
 } catch (Exception e) {
 System.out.println("B Interrupted");
 }
 }
}

```

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

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

class Deadlock implements Runnable {
 A a = new A();
 B b = new B();

 Deadlock() {
 Thread.currentThread().setName("MainThread");
 Thread t = new Thread(this, "RacingThread");
 t.start();
 a.foo(b);
 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();
 System.out.println("NAME:SHIVARAJ K PUJARI");
 System.out.println("USN:1BM22CS259");
}
}
```