# B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

## LAB REPORT

## 23CS3PCOOJ

Submitted in partial fulfillment of the requirements for

Lab Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:


## RAHUL N RAJU

## (1BM22CS215)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019


2023-2024.

# INDEX

12-12-23

WRA 12-12-23

1) Write a program to print "Hello World"

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

output: hello world

2) Write a program to find the area of a rectangle using parseInt()

```
class area {
public static void main (String args [])  {
int length, breadth;
length = Integer.parseInt(args[0]);
breadth = Integer.parseInt(args[1]);
int area = length * breadth;
System.out.println ("length of rectangle = " + length);
System.out.println ("breadth of rectangle = " + breadth);
System.out.println ("area of rectangle = " + area);
}}
```

Output: length of rectangle = 10
breadth of rectangle = 8
area of rectangle = 80

3) Write a program to show example of scanner

```
import java.util.Scanner;
class scanner {
public static void main (String args [])
{
int a; float b; String s;
Scanner in = new Scanner (System.in);
System.out.println ("enter a string: ");
s = in.nextLine ();
System.out.println ("you entered string "+s);
System.out.println ("enter an integer");
a = in.nextInt ();
System.out.println ("you entered integer "+a);
System.out.println ("enter a float");
b = in.nextFloat ();
System.out.println ("you entered float "+b);
}
}
```

Output: enter a string: java
you entered string java
enter an integer 100
you entered integer 100
enter a float 78.6
you entered float 78.6

4) Write a program to know the usage of array

```
class AutoArray {
public static void main (String args[]) {
int month_days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
System.out.println ("April has " + month_days[3] + "days");
}
```

Output: April has 31 days

5) Write a program to find the factorial of a given number

```
class factorial {
 public static void main (String args[])
 {
    int fac = 1;
    System.out.println("Enter a number: ");
    Scanner sc = new Scanner (System.in);
    int n = sc.nextInt();
    for(int i=1; i<=n; i++) {
        fac = fac * i;
    }
    System.out.println("The factorial: "+fac);
 }
}
```

Output: enter a number: 5
the factorial: 120

6) Write a program to check if it is a palindrome

```
import java.util.Scanner;
class palindrome
{
    public static void main (String args[])
    {
        int n, t, rem, rev = 0;
        Scanner sc = new Scanner (System.in);
        System.out.println("enter a 5 digit number: ");
        n = sc.nextInt();
        t = n;
        while (t>0)
        {
            rem = t%10;
            rev = rev * 10 + rem;
            t = t/10;
```

```
        if (rev == n)
        {
            System.out.println("palindrome");
        }
        else
        {
            System.out.println("not palindrome");
        }
    }
}
```

Output: enter a 5 digit number: 12321
palindrome

7) Write a program to find the sum of 5 digits

```
import java.util.*;
class digits {
    public static void main(String args[]) {
        long number, sum;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a 5-digit number:");
        number = sc.nextLong();
        for(sum = 0; number != 0; number = number/10)
            sum = sum + number %10;
        }
        System.out.println("sum of digits: "+sum);
    }
}
```

Output: Enter a 5-digit number: 12345
Sum of digits: 15

12-12-23

LAB-1 Program

1) Quadratic equation

Develop a Java program that prints all real solutions of equation $ax^2+bx+c=0$.

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

```
        else if (d>0)
        {
            r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = "+r1+"Root 2 = "+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("Root2 = "+r1+"-i"+r2);
        }
    }
}
class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("1BM22CS215 Rahul N Raju");
    }
}
```

Output:

i) enter the coefficients of a, b, c
1 2 1
roots are real and equal
root 1 = root 2 = -1.0

ii) enter the coefficients of a, b, c
2 4 5
roots are imaginary
root 1 = -1.0 + i 1.2247448 7139 1589
root 2 = -1.0 - i 1.2247448 7139 1589

iii) enter the coefficients of a, b, c
1 4 1
roots are real and distinct
root 1 = -0.2679 4919 2431 1228
root 2 = -3.7320 5080 75 6887 7

IBM22CS215 Rahul N Raju

19-12-2023

# LAB Program - 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 & display details and a method to calculate SGPA of a student

```
import java.util.Scanner;

class Subject
{
    int subjectMarks;
    int credits;
    String grade;
}

class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];
    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 your name : ");
        name = s.nextLine();
```

```
        System.out.println("enter your usn: ");
        usn = s.nextLine();
    }

    void getMarks()
    {
        int i;
        for(i=0; i<8 ; i++)
        {
            System.out.println("enter the marks & credits for course"+i+":");
            System.out.println("marks: ");
            int marks = s.nextInt();
            System.out.println("credits : ");
            int credit = s.nextInt();
            subject[i].subjectMarks = marks;
            subject[i].credits = credit;
            if (marks>=90 && marks<=100)
            {
                subject[i].grade = "O";
            }
            else if (marks>=80 && marks<90)
            {
                subject[i].grade = "A+";
            }
            else if (marks>=70 && marks<80)
            {
                subject[i].grade = "A";
            }
            else if (marks>=60 && marks<70)
            {
                subject[i].grade = "B+";
            }
            else if (marks>=50 && marks<60)
```

```
        {
            subject[i].grade = "C";
        }
        else if (marks >= 0 && marks < 40)
        {
            subject[i].grade = "F";
        }
    }
}
void computeSGPA()
{
    int i;
    double sgpa;
    double totalcredits = 0;
    double totalgradepoints = 0;
    for (i=0; i<8; i++)
    {
        totalcredits += subject[i].credits;
        switch (subject[i].grade)
        {
            case "O" : totalgradepoints += 10* subject[i].credits;
            break;
            case "A+" : totalgradepoints += 9* subject[i].credits;
            break;
            case "A" : totalgradepoints += 8* subject[i].credits;
            break;
            case "B+" : totalgradepoints += 7* subject[i].credits;
            break;
            case "B" : totalgradepoints += 6* subject[i].credits;
            break;
            case "C" : totalgradepoints += 5* subject[i].credits;
            break;
```

```
                case "F": totalgradepoints += 0*subject[i].credits;
                break;
            }
        }
        sgpa = totalgradepoints / totalcredits;
        System.out.println("Name : "+name);
        System.out.println("USN : "+usn);
        System.out.println("sgpa is : "+sgpa);
    }
}
class sgpa
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}
```

Output:

```
enter your name: dhoni
enter your usn: 215
enter the marks and credits for course 0:
marks : 90
credits : 4
enter the marks and credits for course 1:
marks : 99
credits : 4
enter the marks and credits for course 2:
marks : 94
credits : 3
```

enter the marks and credits for course 3:
marks: 96
credits: 1

name: dhoni
usn: 215
the sgpa is: 10.00

26-12-23

# LAB Program - 3

Create a class Book which contains four members. Include a constructor. Develop a Java program to create n book objects

```
import java.util.Scanner;

class Books
{
    String name;
    String author;
    int price;
    int numPages;
    Books (String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString()
    {
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
        numPages = "Number of pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}

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

```
{
    Scanner s = new Scanner (System.in);
    int n;
    int i;
    String name, author;
    int price, numPages;
    System.out.println ("Enter the number of books : ");
    n = s.nextInt();
    Books b[];
    b = new Books[n];
    for (i=0; i<n; i++)
    {
        System.out.println ("Enter the details of book " + (i+1) +
        System.out.println ("Enter the name of the book : ");
        name = s.next();
        System.out.println ("Enter the author name : ");
        author = s.next();
        System.out.println ("Enter the price : ");
        price = s.nextInt();
        System.out.println ("Enter the number of pages : ");
        numPages = s.nextInt();
        b[i] = new Books (name, author, price, numPages
    }
    System.out.println ("Book Details : ");
    for (i=0; i<n; i++)
    {
        System.out.println (b[i]);
    }
}
}
```

Output:

```
Enter the number of books: 2
Enter the details of book 1:
Enter the name of the book: BhagavadGita
Enter the author name: Krishna
Enter the price: 100
Enter the number of pages: 1000
Enter the details of book 2:
Enter the name of the book: Ramayana
Enter the author name: Valmiki
Enter the price: 200
Enter the number of pages: 2000

Book Details:
Book name: BhagavadGita
Author name: Krishna
Price: 100
Number of pages: 1000

Book name: Ramayana
Author name: Valmiki
Price: 200
Number of pages: 2000
```

26/12/23

2-1-24

# Lab Program-4

Develop a Java Program to create an abstract class named Shape.

```
import java.util.Scanner;
class inputScanner
{
    protected Scanner s;
    public inputScanner()
    {
        s = new Scanner(System.in);
    }
    public int getInput(String message)
    {
        System.out.println(message)
        return s.nextInt();
    }
}

abstract class Shape extends inputScanner
{
    protected int a, b;
    public Shape()
    {
        super();
    }
    abstract public void printArea();
}

class Rectangle extends Shape
{
    protected int a, b;
    public Rectangle()
```

```
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter the length : ");
        b = getInput("Enter the breadth : ");
        int area = a*b;
        System.out.println("Area of the rectangle : "+area);
    }
}

class Triangle extends Shape
{
    protected int a, b;
    public Triangle()
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter the side 1 : ");
        b = getInput("Enter the side 2 : ");
        double area = 0.5*a*b;
        System.out.println("Area of the Triangle : "+area);
    }
}

class Circle extends Shape
{
    protected int a;
    public Circle()
    {
        super();
    }
```

```
    public void printArea()
    {
        a= getInput ("Enter the radius: ");
        double area = 3.14 * a * a;
        System.out.println ("Area of the circle: " +area);
    }
}

public class mainArea
{
    public static void main (String args[])
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

Output:

```
Enter the length : 3
Enter the breadth : 4
Area of the Rectangle : 12
Enter the side 1 : 5
Enter the side 2 : 6
Area of the Triangle : 15.0
Enter the radius: 4
Area of the circle : 50.24
```

02/01/24

9-1-24

# Lab Program - 5

Develop a Java program to develop a class Bank with current account & savings account.

```
import java.util.Scanner;
class Account
{
    String name;
    int accno;
    String type;
    double balance;
    account (String name, int accno, String type, double balance)
    {
        this.aname = 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+" accno: "+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: " + 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 min balance,
                service charges imposed: " + serviceCharges);
            balance -= serviceCharges;
            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 amount1, amount2;
        account ac = new account(name, accno, type, balance);
        savAcct sa = new savAcct(name, accno, balance);
        curAcct ca = new curAcct(name, accno, balance);
        while (true)
        {
            if (ac.type.equals("savings"))
            {
                System.out.println("\n Menu \n 1. Deposit 2. withdraw
                                    3. compute interest 4. display ");
                System.out.println("enter the choice: ");
                ch = s.nextInt();
                switch(ch)
                {
                    case 1: System.out.println("enter the amount: ");
                            amount1 = s.nextInt();
                            sa.deposit(amount1);
                            break;
```

```
        case 2: System.out.println("enter the amount:");
                amount2 = s.nextInt();
                sa.withdraw(amount2);
                break;
        case 3: sa.interest();
                break;
        case 4: sa.display();
                break;
        case 5: System.exit(0);
        default: System.out.println("invalid input");
                break;
        }
    }
    else
    {
        System.out.println("\n Menu \n 1.deposit 2.withdraw 3.display");
        System.out.println("enter the choice:");
        ch = s.nextInt();
        switch(ch)
        {
            case 1: System.out.println("enter the amount:");
                    amount1 = s.nextInt();
                    ca.deposit(amount1);
                    break;
            case 2: System.out.println("enter the amount:");
                    amount2 = s.nextInt();
                    ca.withdraw(amount2);
                    ca.checkmin();
                    break;
            case 3: ca.display();
                    break;
            case 4: System.exit(0);
        }
    }
}
```

Output:

Enter the name: Dhoni
Enter the type (current / savings):
current
Enter the account number:
1001
Enter the initial balance:
1000

Menu:
1. Deposit 2. withdraw 3. display
1
Enter the amount: 1000

Menu:
1. Deposit 2. Withdraw 3. display
2
Enter the amount: 500

Menu:
1. Deposit 2. Withdraw 3. display
3

name: Dhoni accno: 1001 type: current balance: ~~2005.00~~ 1500.0

9/1/24

23-1-24

# Lab - Program - 7

```
package CIE
import java.util.Scanner;
public class Student
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails (){
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter USN: ");
        usn = s.next();
        System.out.println ("Enter name: ");
        name = s.next();
        System.out.println ("Enter Semester: ");
        sem = s.nextInt();
    }
    public void displayStudentDetails(){
        System.out.println ("USN: "+usn);
        System.out.println ("Name: "+name);
        System.out.println ("semester: "+sem);
    }
}

//internals
package CIE;
import java.util.Scanner;
public class internals extends Student
{
    protected int marks[] = new int[5];
    public void inputCIEmarks()
    {
```

```
        Scanner s = new Scanner (System.in);
        System.out.println ("enter the interal marks for "+name
        for (int i=0; i<5; i++)
        {
            System.out.println ("enter the "+(i+1)+" internals mark
            marks[i] = s.nextInt();
        }
    }
}

//Externals
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.println ("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.Externals;
class Main
{
    public static void main(String args[])
    {
        int n=2;
        Externals finalMarks[] = new Externals[n];
        for(int i=0; i<n; 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<n; i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Output:

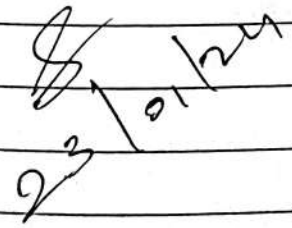
Enter USN: 777 Enter name: dhoni Enter sem: 2
Enter CIE marks:
Enter the interval marks for dhoni
enter the 1 internals marks 46
enter the 2 internals marks 43
enter the 3 internals marks 49
enter the 4 internals marks 42
enter the 5 internals marks 43
enter SEE marks:
subject 1 marks: 87
subject 2 marks: 88
subject 3 marks: 76
subject 4 marks: 98
subject 5 marks: 99

Enter USN: 18 Enter name: Virat Enter sem: 3
Enter CIE marks
enter the internal marks for virat
enter the 1 internals marks 87
enter the 2 internals marks 78
enter the 3 internals marks 98
enter the 4 internals marks 69
enter the 5 internals marks 87

enter SEE marks
subject 1 marks : 97
subject 2 marks : 88
subject 3 marks : 68
subject 4 marks : 78
subject 5 marks : 97

displaying data :
USN : 777 Name : dhoni sem : 2
Subject 1 : 90
subject 2 : 92
Subject 3 : 94
Subject 4 : 97
subject 5 : 100

USN : 18 Name : Virat sem : 3
Subject 1 : 94
Subject 2 : 95
Subject 3 : 96
Subject 4 : 96
Subject 5 : 100



30-1-24

# Lab- Program -8

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

```
import java.util.Scanner;
class WrongAge extends Exception
{
    public WrongAge(String s){
        super(s);
    }
}

class Father
{
    protected int fatAge;
    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.println("enter father's age:");
        fatAge = s.nextInt();
        if (fatAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    public void displayfat(){
        System.out.println("father's age is : " + fatAge);
    }
}

class Son extends Father
{
    private int sonAge;
    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.println("enter the son's age : ");
        sonAge = s.nextInt();
```

```
        if(sonAge >= fatAge)
            throw new WrongAge("son's age is more than or equal to father's age");
        else if(sonAge < 0)
            throw new WrongAge("age cannot be negative");
    }
    public void display() {
        System.out.println("son's age is : "+sonAge);
    }
}
public class MainFatSon
{
    public static void main(String args[])
    {
        try {
            Son son = new Son();
            son.displayfat();
            son.display();
        }
        catch(WrongAge e) {
            System.out.println("error: "+e.getMessage());
        }
    }
}
```

Output:

1) enter father's age: 45
enter the son's age: 18
father's age: 45
son's age is : 18

2) enter father's age : 23
enter the son's age: 54
error: son's age is more than or equal to father's age

3) enter father's age : -87
error: Age cannot be negative

20/01/24

# Program - 8

WAP which creates 2 threads, one displaying "BMS college of Engineering" once every 10 secs and another displaying "CSE" once every 2 secs

```
class BMSthreads extends Thread
{
    @Override
    public void run()
    {
        while (true) {
            System.out.println ("BMS college of Engineering");
            try {
                Thread.sleep (10000);
            }
            catch (InterruptedException e) {
                e.printStackTrace() ; }
        }
    }
}

class CSEThread extends Thread {
    @Override
    public void run() {
        while (true) {
            System.out.println ("CSE");
            try {
                Thread.sleep (2000) ; }
            catch (InterruptedException e) {
                e.printStackTrace ();
            }
        }
    }
}
```

```
public class ThreadExample {
    public static void main(String args[]) {
        BMSThreads bmsthread = new BMSThreads();
        bmsthread.start();
        CSEThread cseThread = new CSEThread();
        cseThread.start();
    }
}
```

Output:-

```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
```

6/02/24

20-2-24

# Lab - 9

Write a program that creates a user interface to perform integer divisions.

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

class UserInterface {
    UserInterface() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and dividend: ");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel arslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(arslab);
```

```
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 static void main
    public void actionPerformed(ActionEvent evt){
        try{
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            alab.setText("In A = "+a);
            blab.setText("In B = "+b);
            anslab.setText("In Ans = "+ans);
        }catch(NumberFormatException e){
            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);
}
```

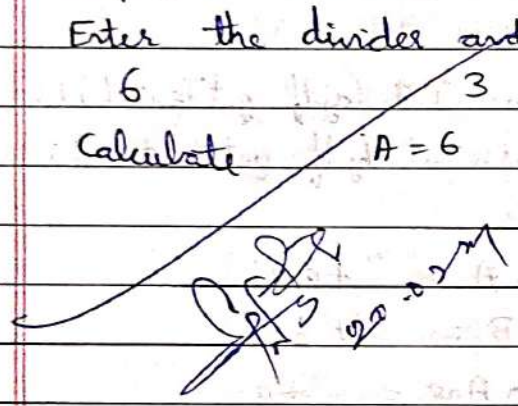
```
public static void main (String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new UserInterface();
        }
    });
}
}
```

Output:

Enter the divider and dividend

6 3

Calculate A = 6 B = 3 Ans = 2



Functions used

JFrame: It is a top level container in Java String that represents a window with a title bar, border and optional menubar.

13-2-24

# Lab Program - 10 (10M)

Demonstrate Inter process communication and deadlock

```
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("Interrupted Exception 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");
            wait();
        } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
        }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        System.out.println("\nIntimate Consumer\n");
        notify();
    }
}
```

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

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
      this.q = q;
      new Thread(this, "consumer").start();
    }
    public void run() {
      int i = 0;
      while (i < 5) {
        int r = q.get();
        System.out.println("Consumed : " + r);
        i++;
      }
    }
}

class PCFixed {
    public static void main(String args[]) {
      ~~PC~~ Q q = new Q();
```

```
        new Producer(q);
        new Consumer(q);
        System.out.println("Press control-C to stop.");
    }
}
```

Output:

```
Press Control-C to stop
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer

Producer waiting
consumed: 0
Got: 1
Intimate Producer
consumed: 1
Put: 2
Intimate Consumer

Producer waiting
Got: 2
Intimate Producer
consumed: 2
Put: 3
Intimate Consumer

Producer waiting
Got: 3
Intimate Producer
consumed: 3
Put: 4
Intimate Consumer
Got: 4
Intimate Producer
consumed: 4
```

Program (10 A)
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("Main Thread");
        Thread t = new Thread(this, "Racing Thread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run()
    {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[])
    {
        new Deadlock();
    }
}
```

Output:

Main Thread entered A.foo

Racing Thread entered B.bar

Mainthread trying to call B.last()

Racing Thread trying to call A.last()

Inside A.last

Back in other thread

Inside A.last

Back in main thread

8/3/2/24

LAB - 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

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

```
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + r1 + " Root2 = " +
r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1 = " + r1 + " + i"+r2);
            System.out.println("Root1 = " + r1 + " - i"+r2);
        }
    }
}

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

LAB - 2

Sgpa calculator.

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;
    String grade;
}

class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];
    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 your name : ");
        name = s.nextLine();
        System.out.println("enter your usn : ");
        usn = s.nextLine();
    }
```

```
    void getMarks()
    {
        int i;
        for(i=0;i<8;i++)
        {
            System.out.println("enter the marks and credits for course " +
(i+1) + ":");
            System.out.println("marks : ");
            int marks = s.nextInt();
            System.out.println("credits : ");
            int credit = s.nextInt();
            subject[i].subjectMarks = marks;
            subject[i].credits = credit;

            if(marks >= 90 && marks<=100)
            {
                subject[i].grade = "O";
            }
            else if(marks>=80 && marks<90)
            {
                subject[i].grade = "A+";
            }
            else if(marks>=70 && marks<80)
            {
                subject[i].grade = "A";
            }
            else if(marks>=60 && marks<70)
            {
                subject[i].grade = "B+";
            }
            else if(marks>=50 && marks<60)
            {
                subject[i].grade = "B";
            }
            else if(marks>=40 && marks<50)
            {
                subject[i].grade = "C";
            }
            else if(marks>=0 && marks<40)
            {
                subject[i].grade = "F";
            }
```

```
        }
    }
    void computeSGPA()
    {
        int i;
        double sgpa;
        double totalcredits = 0;
        double totalgradepoints = 0;

        for(i=0;i<8;i++)
        {
            totalcredits += subject[i].credits;
            switch(subject[i].grade)
            {
                case "O" : totalgradepoints += 10*subject[i].credits;
                break;
                case "A+" : totalgradepoints += 9*subject[i].credits;
                break;
                case "A" : totalgradepoints += 8*subject[i].credits;
                break;
                case "B+" : totalgradepoints += 7*subject[i].credits;
                break;
                case "B" : totalgradepoints += 6*subject[i].credits;
                break;
                case "C" : totalgradepoints += 5*subject[i].credits;
                break;
                case "F" : totalgradepoints += 0*subject[i].credits;
                break;
            }
        }
        sgpa = totalgradepoints/totalcredits;
        System.out.println("the sgpa is : "+sgpa);
    }
}
class sgpa
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
```

```
    }
}
```

```
                                    LAB - 3

Create a class Book which contains four members: name, author, price,
num_pages. Include a constructor to set the values for the members. Include
methods to set and get the details of the objects. Include a toString( )
method that could display the complete details of the book. Develop a Java
program to create n book objects.

import java.util.Scanner;

class Books
{
    String name;
    String author;
    int price;
    int numPages;

    Books(String name,String author,int price,int numPages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.numPages=numPages;
    }

    public String toString()
    {
        String name,author,price,numPages;
        name="Book name:" +this.name+ "\n";
        author="Author name:" +this.author+ "\n";
        price="Price:" +this.price+ "\n";
        numPages="Number of pages:" +this.numPages+ "\n";
        return name+author+price+numPages;
    }
}

public class Mainbook
```

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

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

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

        for(i=0;i<n;i++)
        {
            System.out.println("Enter the details of book" + (i+1) + ":");
            System.out.println("Enter the name of the book:");
            name=s.next();
            System.out.println("Enter the author name:");
            author=s.next();
            System.out.println("Enter the price:");
            price=s.nextInt();
            System.out.println("Enter the number of pages:");
            numPages=s.nextInt();

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

        System.out.println("Book Details:");
        for(i=0;i<n;i++)
        {
            System.out.println(b[i]);
        }
    }
}
```

LAB - 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape

```
import java.util.Scanner;

class inputScanner
{

    protected Scanner s;

    public inputScanner()
    {
        s = new Scanner(System.in);
    }

    public int getInput(String message)
    {
        System.out.println(message);
        return scanner.nextInt();
    }



}

abstract class Shape extends inputScanner
{
    protected int a,b;

    public Shape()
    {
        super();
    }

    abstract public void printArea();
}

class Rectangle extends Shape
{
```

```
    protected int a,b;
    public Rectangle()
    {
        super();
    }

    public void printArea()
    {

        a=getInput("Enter the length:");
        b=getInput("Enter the bredth:");
        int area= a*b;
        System.out.println("Area of the Rectangle:" +area);
    }
}

class Triangle extends Shape
{
    protected int a,b;
    public Triangle()
    {
        super();
    }

    public void printArea()
    {
        a=getInput("Enter the side1:");
        b=getInput("Enter the side2:");
        double area=0.5*a*b;
        System.out.println("Area of the Triangle:" +area);
    }

}

class Circle extends Shape
{
    protected int a;
    public Circle()
    {
        super();
    }
```

```
    public void printArea()
    {
        a=getInput("Enter the radius:");
        double area=3.14*a*a;
        System.out.println("Area of the Circle:" +area);

    }

}

public class MainShape
{
    public static void main(String[] args)
    {
        Rectangle r=new Rectangle();
        Triangle t=new Triangle();
        Circle c=new Circle();

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

LAB - 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class account
{
    String 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,cant withdraw");
        }
```

```
    }

    void display()
    {

System.out.println("name:"+name+"accno:"+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:"+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 min balance,service
charges imposed:"+serviceCharges);
            balance-=serviceCharges;
            System.out.println("balance is:"+balance);
        }

    }

}
class accountMain
{
    public static void main(String a[])
    {
        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 intial balance:");
        double balance=s.nextDouble();
        int ch;
        double amount1,amount2;
        account acc=new account(name,accno,type,balance);
        savAcct sa=new savAcct(name,accno,balance);
        curAcct ca=new curAcct(name,accno,balance);
        while(true)
        {
            if(acc.type.equals("savings"))
            {
                System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
interest 4.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
                {
                    case 1:System.out.println("enter the amount:");
                        amount1=s.nextInt();
```

```
                        sa.deposit(amount1);
                        break;
                    case 2:System.out.println("enter the amount:");
                        amount2=s.nextInt();
                        sa.withdraw(amount2);
                        break;
                    case 3:sa.interest();
                        break;
                    case 4:sa.display();
                        break;
                    case 5:System.exit(0);
                    default:System.out.println("invalid input");
                        break;
                }
            }
            else
            {
                System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
                {
                    case 1:System.out.println("enter the amount:");
                        amount1=s.nextInt();
                        ca.deposit(amount1);
                        break;
                    case 2:System.out.println("enter the amount:");
                        amount2=s.nextInt();
                        ca.withdraw(amount2);
                        ca.checkmin();
                        break;

                    case 3:ca.display();
                        break;
                    case 4:System.exit(0);
                    default:System.out.println("invalid input");
                        break;
                }
            }
        }
    }
}
```

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.

```
// Internals.java
package CIE;

import java.util.Scanner;

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



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

```
// Student.java
package CIE;

import java.util.Scanner;

public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
```

```
    public void inputStudentDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = scanner.next();
        System.out.print("Enter Name: ");
        name = scanner.next();
        System.out.print("Enter Semester: ");
        sem = scanner.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
// Externals.java
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 scanner = new Scanner(System.in);
        System.out.println("Enter SEE Marks for " + name);
        for (int i = 0; i < 5; i++) {
```

```
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.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]);
    }
}
```

LAB - 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;

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

class Father
{
    protected int fatAge;
    public Father() throws WrongAge
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter father's age : ");
        fatAge = s.nextInt();
        if(fatAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    public void displayfat()
    {
            System.out.println("father's age is : "+fatAge);
    }
}

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

```
        Scanner s = new Scanner(System.in);
        System.out.println("enter the son's age : ");
        sonAge = s.nextInt();
        if(sonAge >= fatAge)
        {
            throw new WrongAge("son's age is more than or equal to father's
age");
        }
        else if(sonAge<0)
        {
            throw new WrongAge("age cannot be negative");
        }
    }
    public void display()
    {
        System.out.println("son's age is : "+sonAge);
    }
}

public class Mainfatson
{
    public static void main(String args[])
    {
        try
        {
            Son son = new Son();
            son.displayfat();
            son.display();
        }
        catch(WrongAge e)
        {
            System.out.println("error: "+e.getMessage());
        }
    }
}
```

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 BMSThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("BMS college of engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSEThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class threadEx {
    public static void main(String[] args) {
        BMSThread bms = new BMSThread();
        bms.start();
        CSEThread cse = new CSEThread();
        cse.start();
    }
}
```

LAB - 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

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

class UserInterface {
    UserInterface() {
        // 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 (ArithmeticException e) {
                    alab.setText(" ");
                    blab.setText(" ");
                    anslab.setText(" ");
                    err.setText("B should be NON zero!");
                }
            }
        });

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

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

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("\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<2) {
    q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
    this.q = q;
    new Thread(this, "Consumer").start();
    }
    public void run() {
    int i=0;
    while(i<5) {
    int r=q.get();
    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.");
    }
}
```

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

        System.out.println(name + " trying to call B.last()");
        b.last();

    }

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

class B
{
    synchronized void bar(A a)
    {
        String name =Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");

        try
        {
            Thread.sleep(1000);
        }
        catch(Exception e)
        {
```

```
            System.out.println("B Interrupted");
        }

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

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

class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock()
    {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this,"RacingThread");
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
    }

    public void run()
    {
        b.bar(a); // get lock on b in other thread.
        System.out.println("Back in other thread");
    }

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