B.M.S. College of Engineering
(Autonomous Institution affiliated to VTU, Belagavi)

Department of Computer Science and Engineering

Object Oriented Programming in Java

LAB Report

23CS3PCOOJ

Submitted by

RANJAN DEVI
(1BM22CS219)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

INDEX

1BM22CS219

Date: 11/12/23
Page No:

LAB 1

1. 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("Root 1 = Root 2 =
                + 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("Root 1 = " + 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("Root 2 = " + r1 + "-i" +r2);
    }
  }
}

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

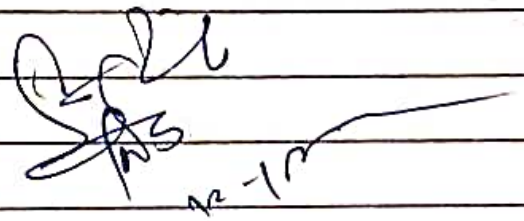
1BM22CS219
Ranjan Devi

Date :
Page No :

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

OUTPUT :

Enter the coefficients of a, b, c : 1 2 1
Roots are real and equal.
Root 1 = Root 2 = -1



1BM22CS219
RANJAN DEVI

Date :
Page No :

LAB - 2

```
import java.util.Scanner;
class Subject
{
    int marks;
    int credits;
    int grade;
}
class Student
{
    Subject sub[];
    String name;
    String usn;
    Scanner s;

    Student()
    {
        int i;
        sub = new Subject[8];
        for(i=0; i<8; i++)
            sub[i] = new Subject();
        s = new Scanner(System.in);
    }

    void getDetails()
    {
        System.out.println("Enter student name:");
        name = s.nextLine();
        System.out.println("Enter student usn:");
        usn = s.nextLine();
    }
```

```
void getMarks()
{
    int i;
    for(i=0 ; i<8 ; i++)
    {
        System.out.println("Enter marks:");
        sub[i].marks = s.nextInt();
        System.out.println("Enter credit:");
        sub[i].credits = s.nextInt();
        if (sub[i].marks < 40)
            sub[i].grade = 0;
        else if (sub[i].marks >=40 && sub[i].
                marks < 60)
            sub[i].grade = 6;
        else if (sub[i].marks >=70 && sub[i].
                    marks < 80)
            sub[i].grade = 7;
        else if (sub[i].marks >= 80) && sub[i].
                        marks < 90)
            sub[i].grade = 8;
        else if (sub[i].marks >= 90) &&
                    sub[i].marks <= 100)
            sub[i].grade = 10;
        else
            sub[i].grade = 0;
    }
}

double computeSGPA()
{
    int i;
    int n-sum = 0, d-sum = 0;
```

```
int numerator;
for(i=0 ; i<8 ; i++)
{
        numerator = sub[i].credits * sub[i].
                                        grade;
        n-sum = n-sum + numerator;
        d-sum = d-sum + sub[i].credits;
}
SGPA = n-sum / d-sum;
return (SGPA);
}}

class mainClass
{
    public static void main (String args[])
    {
        Student s1 = new Student();
        s1.getDetails();
        s1.getMarks();
        System.out.println("SGPA = " + s1.compute
                                        SGPA());
}}
```

1BM22CS219

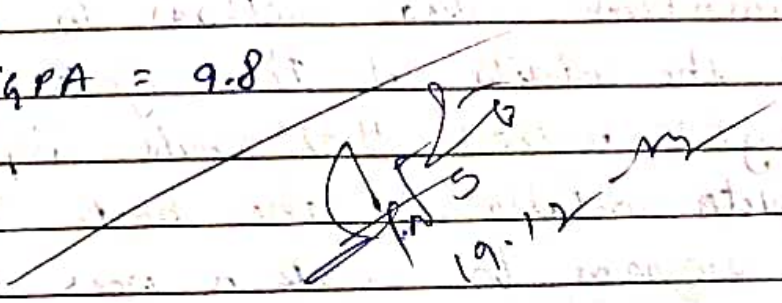
Date :
Page No :

OUTPUT:

```
Enter student name:
Ranjan
Enter student usn:
1BM22CS219
Enter marks:
95
Enter credits:
4
Enter marks:
93
Enter credits:
4
Enter marks:
88
Enter credits:
3
Enter marks:
92
Enter credits:
3
Enter marks:
91
Enter credits:
3
Enter marks:
90
Enter credits:
1
Enter marks:
92
```

Date :
Page No :

Enter credits:
1
Enter marks:
94
Enter credits:
1
SGPA = 9.8



26/12/23 1BM22CS219 Date : Page No :

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 title;
    String author;
    int price;
    int numpages;

    Books(String title, String author, int price,
          int numpages)
    {
        this.title = title;
        this.author = author;
        this.price = price;
        this.numpages = numpages;
    }
```

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

class Main
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int n, i;
        String title, author;
        int price, numpages;

        System.out.println("Enter the no. of
                books: ");
        n = s.nextInt();

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

Date :
Page No :

```
for(i=0 ; i<n ; i+1)
{
    System.out.println("Enter the title,
    author name , price and
    number of pages of book: ");
    title = s.next();
    author = s.next();
    price = s.nextInt();
    numpages = s.nextInt();
    b[i] = new Books(title, author,
                price, numpages);
}

for(i=0 ; i<n ; i+1)
{
    System.out.println(b[i]);
}
}
}
```

1BM22CS219

OUTPUT :

```
Enter the no. of books:
3
Enter the title, author name, price and
number of pages of book:
Hamlet WilliamShakespeare 500 480
Enter the title, author name, price and
number of pages of book:
HarryPotter JKRowling 550 580
Enter the title, auther name, price and
number of pages of book:
1984 George Orwell 600 520

Book name: Hamlet
Author name: WilliamShakespeare
Price: 500
Number of pages: 480

Book name: HarryPotter
Author name: JKRowling
Price: 550
Number of pages: 580

Book name: 1984
Author name: George Orwell
Price: 600
Number of pages: 520
```

2/1/2024

LAB-4

Develop a Java program to create abstract class shape which finds area of different shapes.

```
import java.util.Scanner;

abstract class Shape
{
    int dim1;
    int dim2;
    Scanner s = new Scanner(System.in);
    abstract void printArea();
    abstract void input();
}

class Rectangle extends Shape
{
    void input()
    {
        System.out.println("Enter length and
            breadth:");
        dim1 = s.nextInt();
        dim2 = s.nextInt();
    }
    void printArea()
    {
        System.out.println("Area of rectangle:
            "+(dim1 * dim2)+" sq units");
    }
}
```

```
class Triangle extends Shape
{
    void input()
    {
        System.out.println("Enter base and
            height : ");
        dim1 = s.nextInt();
        dim2 = s.nextInt();
    }
    void printArea()
    {
        System.out.println("Area of triangle = "
            + (dim1 * dim2 /2) + " sq units");
    }
}

class Circle extends Shape
{
    void input()
    {
        System.out.println("Enter radius: ");
        dim1 = s.nextInt();
    }
    void printArea()
    {
        System.out.println("Area of circle = " +
            (3.14 * dim1 * dim1) + " sq units");
    }
}
```

1BM22CS219

Date :
Page No :

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

OUTPUT::

```
Enter length and breadth:
4 5
Area of rectangle = 20 sq units
Enter base and height:
6.3
Area of triangle = 9 sq units
Enter radius:
4
Area of circle = 50.24 sq units
```

1BM22CS219

Date : 9/1/2024
Page No :

LAB - 5

~~import~~

Develop a Java program to create a class Bank and respective subclasses Savings and Current account.

```
import java.util.*;
class Bank
{
    Scanner s=new Scanner(System.in);
    String customer;
    String accno;
    void get()
    {
        System.out.println("Enter customer name:");
        customer = s.next();
        System.out.println("Enter account number:");
        accno = s.next();
    }
}

class Cur_acct extends Bank
{
    double bal = 0;
    double dep;
    void issue_cheque()
    {
        System.out.println("Cheque book issued");
    }
```

```
void deposit1()
{
    System.out.println("Enter the amount
        to be deposited: ");
    dep = s.nextInt();
    bal += dep;
}
void check()
{
    if(bal < 1000)
    {
        System.out.println("Minimum
            balance must be 1000");
        bal = bal - 5;
        System.out.println("Service charge
                imposed");
    }
}
void display()
{
    System.out.println("Balance = "+bal);
}
}

class Sav_acct extends Bank
{
    double bal=0;
    double dep, draw;
    void deposit2()
    {
        System.out.println("Enter the amount to
            be deposited: ");
```

```
        dep = s.nextInt();
        bal += dep;
    }
    void withdraw()
    {
        System.out.println("Enter the amount
                of withdrawal :");
        draw = s.nextInt();
        bal -= draw;
    }
    void compute_interest()
    {
        bal = bal + (0.06 * bal);
        System.out.println("Current balance ="
                + bal);
    }
}

class Account
{
    public static void main(String args[])
    {
        int ch, type;
        Scanner sc = new Scanner(System.in);
        Bank b = new Bank();
        b.get();
        Curr_acct c = new Cur_acct();
        Sav_acct a = new Sav_acct();
        System.out.println("Enter the account
            type (1.savings / 2.current) :");
        type = sc.nextInt();
```

```
if (type==1)
{
    do
    {
        System.out.println("--Main menu--");
        System.out.println("1.Deposit
        2.Withdrawal 3.Compute Interest
        4.Exit");
        System.out.println("Enter your choice");
        ch=sc.nextInt();
        switch(ch)
        {
            case 1:
                a.deposit2();
                break;
            case 2:
                a.withdraw();
                break;
            case 3:
                a.compute_interest();
                break;
        }
    }while(ch!=4);
    System.out.println("Exited");
}

else
{
    do
    {   System.out.println("1.Deposit 2.Cheque
        issue 3.Display 4.Exit");
        System.out.println("Enter your choice:");
```

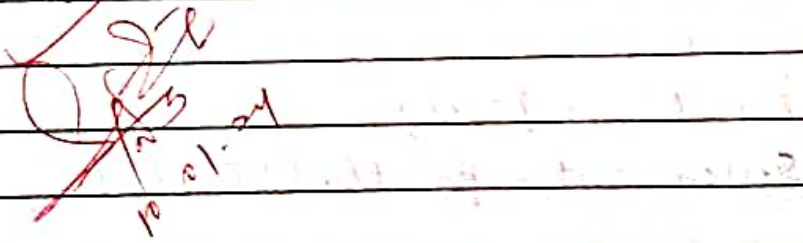
Date :
Page No :

```
ch = sc.nextInt();
switch(ch)
{
    case 1:
        c.deposit1();
        break;
    case 2:
        c.issue_cheque();
        break;
    case 3:
        c.display();
        c.check();
        break;
    }
}while(ch!=4);
System.out.println("Exited");
}
}
}
```

OUTPUT :-

```
Enter customer name:
Ranjan
Enter account number:
5785413
Enter the account type (1.savings/2.current):
1
--Main menu--
1.Deposit 2.Withdrawal 3.Compute Interest 4.Exit
Enter your choice: 1
Enter the amount to be deposited: 5000
```

--Main menu--
1. Deposit 2. Withdrawal 3. Compute Interest 4 Exit
Enter your choice : 2
Enter the amount of withdrawal :
200
--Main menu--
1. Deposit 2. Withdrawal 3. Compute Interest 4. Exit
Enter your choice : 3
Current balance = 5088.0
--Main menu--
1. Deposit 2. Withdrawal 3. Compute Interest 4. Exit
Enter your choice : 4
Exited



16/1/2024

Date :
Page No :

STRINGS

1. Demonstrate various string constructor with proper java programs.

O/p -
s = javaprogramming
s1 = programming

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

Length of s = 15
Length of s1 = 11
Length of literal (Java) = 4
This is Lab 6 program

3. Demonstrate tostring();

Dimensions are 3.0 by 5.0 by 4.0
Box b: Dimensions are 3.0 by 5.0 by 4.0

4. Using getchars(), extract Bmsce from "Welcome to Bmsce college".

Extracted :
BMSCE

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

The String (JAVA) after conversion is :
74,97,118,97
The String (JAVA) after toCharArray is :
JAVA

6.

Bmsce equals Bmsce -> true
Bmsce equals College -> false
Bmsce equals BMSCE -> false
Bmsce equalsIgnoreCase BMSCE -> true

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

String : Welcome to Bmsce College of Engineering
Substring is matched.

8. Demonstrate startswith().

String : Welcome
Startswith 'Wel' = true

9. Demonstrate endswith()
String : Welcome
Endswith 'ome' = true
Endswith 'W' = false

Date :
Page No :

10. Demonstrate the output for equals() versus ==

o/p -

s1 = "Java"
s2 = "Java"
s1 equals s2 -> true
s1 == s2 -> false

11. Write a java program to perform sorting using compareTo()

apple
ball
cat
dog
ent
free
gun
hen
ice
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree

umbrella
van
watch
xmas
yatch
zee

12. Write sorting of numbers from 10 to 1 using compare to ()

Sorted numbers are:
1 2 3 4 5 6 7 8 9 10

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

o/p :-
Thwas was a test. Thwas was, too
This was a test. Thwas was, too
This is a test. Thwas was, too
This is a test. This was, too
This is a test. This is, too.

14. WAP to demonstrate concat().

o/p -
s1 = hello
s2 = world
Concatenated = helloworld.

23/1/24

1BM22CS219

Date :
Page No :

LAB - 6

Create two packages CIE & SEE to calculate marks.

→ Student.java code -

```
package CIE;
import java.util.Scanner;
public class student
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputDetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter student usn:");
        usn = s.next();
        System.out.print("Enter student name:");
        name = s.next();
        System.out.print("Enter student sem:");
        sem = s.nextInt();
    }
    public void displayDetails()
    {
        System.out.println("Student Details:\n");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Sem: " + sem);
    }
}
```

→ internals.java code

```
package CIE;
import java.util.Scanner;
public class internals extends student
{
    protected int marks[] = new int[5];
    public void CIEmarks()
    {
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.print("Subject" + (i+1)
                    + "marks: ");
            marks[i] = s.nextInt();
        }
    }
}
```

→ externals.java code

```
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 SEEMarks()
{
    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 calMarks()
{
    for(int i=0 ; i<5 ; i++)
    {
        finalMarks[i] = marks[i]/2
            +super.marks[i];
    }
}

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

→ Main.java code

```
import SEE.externals;
class Main
{
    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].inputDetails();
            System.out.println("Enter CIE marks:");
            finalMarks[i].CIEmarks();
            System.out.println("Enter SEE marks:");
            finalMarks[i].SEEMarks();
        }
        System.out.println("Displaying data:");
        for(int i=0; i<num; i++)
        {
            finalMarks[i].calMarks();
            finalMarks[i].displayDetails();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

→ Command Prompt:

file_path > javac -d . student.java
file_path > javac -d . internals.java
file_path > javac -d . externals.java
file_path > javac -d . Main.java

1BM22CS219

OUTPUT:

Enter student usn: 1BM22CS208
Enter student name: Prethi
Enter sem: 3
Enter CIE marks:
Subject 1 marks: 36
Subject 2 marks: 37
Subject 3 marks: 38
Subject 4 marks: 34
Subject 5 marks: 39
Enter SEE marks:
Subject 1 marks: 90
Subject 2 marks: 95
Subject 3 marks: 94
Subject 4 marks: 93
Subject 5 marks: 91
Enter student usn: 1BM22CS212
Enter student name: Rachana
Enter sem: 3
Enter CIE marks:
Subject 1 marks: 36
Subject 2 marks: 37
Subject 3 marks: 38
Subject 4 marks: 32
Subject 5 marks: 37
Enter SEE marks:
Subject 1 marks: 97
Subject 2 marks: 90
Subject 3 marks: 92
Subject 4 marks: 94
Subject 5 marks: 93

Date :
Page No :

Student Details :
USN : 1BM22CS208
Name : Preethi
Sem : 3
Subject 1 : 81
Subject 2 : 84
Subject 3 : 85
Subject 4 : 80
Subject 5 : 84

Complete the page

Student Details :
USN : 1BM22CS212
Name : Rachana
Sem : 3
Subject 1 : 84
Subject 2 : 82
Subject 3 : 84
Subject 4 : 79
Subject 5 : 83

23.01.24

30/1/2024

Date :
Page No :

LAB - 7

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

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

class input {
{   int f_age;
    int s_age;
    Scanner sc = new Scanner (System.in);
}

class father extends input
{
    father() throws WrongAge
    {
        System.out.println("Enter father's age:");
        f_age = sc.nextInt();
        if (f_age < 0)
            throw new WrongAge ("Age cannot
                be negative");
    }
    void display1()
    {   System.out.println("Father's age is "+f_age);
    }
}
```

```
class son extends father
{
    son() throws WrongAge
    {
        System.out.println("Enter son's age:");
        s_age = sc.nextInt();
        if(s_age >= f_age)
        {
            throw new WrongAge("Son's age
            cannot be greater than father's age");
        }
        else if(s_age < 0)
        {
            throw new WrongAge("Age cannot
                be negative");
        }
    }
    void display2()
    {
        System.out.println("Son's age is: "+s_age);
    }
}

class Main
{
    public static void main(String args[])
    {
        try
        {
            son s = new son();
            s.display1();
        }   s.display2();
```

```
        catch (WrongAge e)
        {
            System.out.println("Error: "+e);
        }
}
}
```

OUTPUT -

Enter father's age:
-10
Error: WrongAge : Age cannot be negative

OUTPUT -
Enter father's age:
45
Enter son's age:
50
Error: WrongAge : Son's age cannot be greater than father's age

OUTPUT -
Enter father's age:
40
Enter son's age:
-20
Error: WrongAge : Age cannot be negative

30.01.24

Date : 6/2/24
Page No :

LAB - 8

```
import java.util.*;
class NewThread implements Runnable
{
    String name;
    Thread t;
    NewThread (String threadname)
    {
        name = threadname;
        t = new Thread(this, name);
        t.start();
    }
    public void run()
    {
        try{
        while(true)
        {
            System.out.println(name);
            Thread.sleep(2000);
        }}
        catch (InterruptedException e)
        {
            System.out.println(name + "Interrupted");
        }
    }
}
```

1BM22CS219

Date :
Page No :

```
class MainThread
{
  public static void main (String args[])
  {
    new NewThread ("Computer Science Engineering");
    try {
    while (true)
    { System.out.println("BMS College of
                          Engineering");
      Thread.sleep (10000);
      System.out.print ("\n");
    }
    }
    catch (InterruptedException e)
    {
      System.out.println(" BMS Interrupted");
    }
  }
}
```

OUTPUT:

```
BMS College of Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering

BMS College of Engineering
Computer Science Engineering
Computer Science Engineering
```

Date :
Page No :

Computer Science Engineering
Computer Science Engineering
Computer Science Engineering

BMS College of Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering
Computer Science Engineering

BMS College of Engineering

20/2/24

1BM22CS219

Date :
Page No :

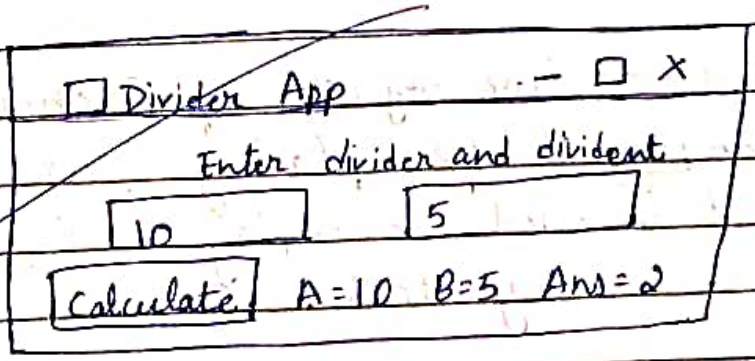
LAB 9

WAP that creates a user interface to perform integer divisions.

```
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());
    jfrm.setDefaultCloseOperation(JFrame.
                    EXIT_ON_CLOSE);
    JLabel jlab = new JLabel("Enter the
          divider and divident :");

    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);

    JButton button = new JButton("calculate");

    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslab = new JLabel();
```

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

ActionListener l = new ActionListener(){
    public void actionPerformed(ActionEvent
                                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("\n A = " + a);
            blab.setText("\n B = " + b);
            anslab.setText("\n Ans = " + ans);
        }
```

```
        catch (NumberFormatException e){
            alab.setText(" ");
            blab.setText(" ");
            anslab.setText(" ");
            err.setText("Enter Only Integers!");
        }
    }
});
    jfrm.setVisible(true);
}

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



Date :
Page No :

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

Date :
Page No :

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

6/2/24 IBM22CS219

Date :
Page No :

LAB 10 (a)

Demonstrate Inter process Communication.

```
class Q
{
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
        while(!valueSet){
        try{
            System.out.println("\n Consumer waiting");
            wait();
        }
        catch(InterruptedException e){
            System.out.println("Interruption caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
    }

synchronized void put(int n)
{
    while(valueSet){
    try{
        System.out.println("Producer waiting");
        wait();
    }catch(InterruptedException e){
```

```
            System.out.println("Interrupted");
        }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        System.out.println("Intimate Consumer");
        notify();
    }
}

class Producer implements Runnable
{
    Q q;
    Producer(Q q)
    {   this.q = q;
        new Thread(this, "Produces").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[])
    {
        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println(
    }
}
```

OUTPUT :

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

Got : 1
Intimate Producer
Put : 2
Intimate Consumer
Producer waiting
Got : 2
Intimate Producer
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
Put : 4
Intimate Consumer
Producer waiting
Got : 4
Intimate Producer

13/2/24

Date :
Page No :

LAB - 10 (b)

Demonstrate Deadlock

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

class B
{
    synchronized void veggies(A a)
    {
        String name = Thread.currentThread().
                                    getName();
        System.out.println (name + " entered
                            B.veggies ");
        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,
                  "Running Thread");
      t.start();
      a.fruits(b);
      System.out.println("Back in main
                  thread");
    }
    public void run(){
      b.veggies(a);
      System.out.println("Back in other
                  thread");
    }
```

Date :
Page No :

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

OUTPUT -

```
Main Thread entered A.fruits
Running Thread entered B.veggies
MainThread trying to call B.last()
Inside A.last
Back in main thread
RunningThread trying to call A.last()
Inside A.last
Back in other thread
```

13/2/24

LAB 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class Quadratic
{
	int a, b, c;
	double r1, r2, d;
	void getd()
	{
		Scanner s = new Scanner(System.in);
		System.out.println("Enter the coefficients of a,b,c;");
		a = s.nextInt();
		b = s.nextInt();
		c = s.nextInt();
	}
	void compute()
	{
		while(a==0)
		{
			System.out.println("Not a quadratic equation");
			System.out.println("Enter a non zero value for a:");
			Scanner s = new Scanner(System.in);
		a = s.nextInt();
		}
		d = b*b-4*a*c;
		if(d==0)
		{
			r1 = (-b)/(2*a);
			System.out.println("Roots are real and equal");
			System.out.println("Roo1 = Root2 ="+ r1);
```

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

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

LAB 2

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

```
import java.util.Scanner;
class Subject
{
	int marks;
	int credits;
	int grade;
}
class Student
{
	Subject sub[];
	String name;
	String usn;
	double SGPA;
	double n_sum=0;
	double d_sum=0;
	Scanner s;
	Student()
	{
		int i;
		sub=new Subject[8];
		for(i=0;i<8;i++)
			sub[i]=new Subject();
		s=new Scanner(System.in);
	}
	void getDetails()
	{
		System.out.println("Enter student name:");
		name=s.nextLine();
```

```
                System.out.println("Enter student usn:");
                usn=s.nextLine();
        }
        void getMarks()
        {
                int i;
                int numerator;
                for(i=0;i<8;i++)
                {       System.out.println("Enter marks:");
                        sub[i].marks = s.nextInt();
                        System.out.println("Enter credits:");
                        sub[i].credits = s.nextInt();
                        sub[i].grade=sub[i].marks//10+1;
                        if(sub[i].grade<4 || sub[i].grade>10)
                                sub[i].grade=0;
                        numerator= sub[i].credits * sub[i].grade;
                        n_sum = n_sum + numerator;
                        d_sum = d_sum + sub[i].credits;
                }
        }
        void computeSGPA()
        {
                SGPA= n_sum / d_sum;
                System.out.println("SGPA= "+SGPA);
        }
}
class mainClass{
 public static void main(String args[]){
        Student s1 = new Student();
        s1.getDetails();
        s1.getMarks();
        System.out.println("SGPA= "+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 title;
	String author;
	int price;
	int num_pages;

	Books(String title,String author, int price, int num_pages)
	{
		this.title=title;
		this.author=author;
		this.price=price;
		this.num_pages=num_pages;
	}

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

```
class Main
{
	public static void main(String args[])
	{
		Scanner s= new Scanner(System.in);
		int n,i;
		String title, author;
		int price, num_pages;

		System.out.println("Enter the no. of books: ");
		n= s.nextInt();

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

		for( i=0 ; i<n ; i++)
		{
			System.out.println("Enter the title, author name, price and number of pages of book:");
			title= s.nextLine();
			author= s.nextLine();
			price= s.nextInt();
			num_pages= s.nextInt();
			b[i] = new Books(title,author,price,num_pages);
		}

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

LAB 4

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

```
import java.util.Scanner;
abstract class Shape
{	int dim1;
	int dim2;
	Scanner s= new Scanner(System.in);
	abstract void printArea();
	abstract void input();
}
class Rectangle extends Shape
{
	void input()
	{	System.out.println("Enter length and breadth:");
		dim1=s.nextInt();
		dim2=s.nextInt();
	}
	void printArea(){
		System.out.println("Area of rectangle= "+(dim1*dim2)+ " sq units");
}}
class Triangle extends Shape
{
	void input()
	{	System.out.println("Enter base and height:");
		dim1=s.nextInt();
		dim2=s.nextInt();
	}
	void printArea(){
```

```
                System.out.println("Area of triangle= "+(dim1*dim2/2)+ " sq units");
        }
}
class Circle extends Shape
{
        void input()
        {       System.out.println("Enter radius:");
                dim1=s.nextInt();
        }
        void printArea(){
                System.out.println("Area of circle= "+(3.14*dim1*dim1)+ " sq units");
        }
}
class Area
{
public static void main (String args[]){
                Rectangle r= new Rectangle();
                Triangle t= new Triangle();
                Circle c= new Circle();
                Shape ref;
                ref=r;
                ref.input();
                ref.printArea();
                ref=t;
                ref.input();
                ref.printArea();
                ref=c;
                ref.input();
                ref.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 Bank
{
        String customer;
        String accno;
        Scanner s=new Scanner(System.in);
        void get()
        {
                System.out.println("Enter the customer name:");
                customer=s.next();
                System.out.println("Enter the account number:");
                accno=s.next();
        }
}
class Cur_acct extends Bank
{
```

```
        double bal=0;
        double dep;
        void deposit1()
        {       System.out.println("Enter the amount to be deposited");
                dep=s.nextInt();
                bal += dep;
                System.out.println("Amount"+dep+" is succesfully deposited");
        }
        void issue_cheque()
        {       System.out.println("The cheque book is issued succesfully");
        }
        void check()
        {
             if(bal<1000)
            {
                System.out.println("The minimum amount must be 1000");
                bal=bal-5;
                System.out.println("Service charges are imposed");
             }
         }
        void display()
        {       System.out.println("Balance = "+bal);
        }
}
class Sav_acct extends Bank
{
        double bal=0;
        double dep , draw;
        int rate=6;
        void deposit2()
        {       System.out.println("Enter the amount to be deposited");
                dep=s.nextInt();
                bal  += dep;
```

```
                System.out.println("Amount"+dep+" is succesfully deposited");
        }
        void withdrawal()
        {       System.out.println("Enter the amount to withdraw");
                draw=s.nextInt();
                bal -= draw;
                System.out.println("The balance amount is"+bal);
        }
        void comp_interest()
        {       bal=bal+bal*0.06;
                System.out.println("The balance amount is"+bal);
        }
}
class Account
{
        public static void main(String args[])
        {
                Scanner sc=new Scanner(System.in);
                int ch, type;
                Bank b=new Bank();
                b.get();
                Sav_acct a=new Sav_acct();
                Cur_acct c=new Cur_acct();
                System.out.println("Enter the account type (1.savings/2.current):");
                type=sc.nextInt();
                if(type==1)
                {
                     do
                    {
                        System.out.println("-----Menu-----");
                        System.out.println("1:Deposit  2:Withdrawal  3.Compound interest  4.Exit");
                        System.out.println("Enter your choice:");
                        ch=sc.nextInt();
```

```
                    switch(ch)
                    {
                        case 1: a.deposit2();
                                break;
                        case 2: a.withdrawal();
                                break;
                        case 3: a.comp_interest();
                                break;
                    }
                } while(ch!=4);
            }
            else
            { do {
                    System.out.println("-----Menu-----");
                    System.out.println("1:Deposit  2:Cheque issue  3.Display  4.Exit");
                    System.out.println("Enter your choice:");
                    ch=sc.nextInt();
                    switch(ch)
                    {
                        case 1: c.deposit1();
                                break;
                        case 2: c.issue_cheque();
                                break;
                        case 3: c.check();
                                c.display();
                                break;
                    }
                  }while(ch!=4);
            }
        }
}
```

LAB 6

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

```
// Package CIE
package CIE;
import java.util.Scanner;
public class student
{
	protected String usn = new String();
	protected String name = new String();
	protected int sem;
	public void inputDetails()
	{
		Scanner s=new Scanner(System.in);
		System.out.println("Enter student usn:");
		usn = s.next();
		System.out.println("Enter student name:");
		name = s.next();
		System.out.println("Enter student sem:");
		sem = s.nextInt();
	}
	public void displayDetails()
	{	System.out.println("Student Details:\n");
		System.out.println("USN: "+usn);
		System.out.println("Name: "+name);
		System.out.println("Sem: "+sem);
	}
```

```
}
public class internals extends student
{
	protected int marks[] = new int[5];
	public void CIEmarks()
	{	Scanner s= new Scanner(System.in);
		for(int i=0;i<5;i++)
		{
			System.out.println("Subject "+(i+1)+" marks");
			marks[i] = s.nextInt();
		}
	}
}

// Package SEE
package SEE;
import CIE.internals;
import java.util.Scanner;
public class externals extends internals
{
	protected int marks[];
	protected int finalMarks[];
	public externals()
	{	marks = new int[5];
		finalMarks = new int[5];
	}
	public void SEEMarks()
	{	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 calMarks()
        {       for(int i=0;i<5;i++)
                        finalMarks[i] = marks[i]/2 + super.marks[i];
        }
        public void displayFinalMarks()
        {       for(int i=0;i<5;i++)
                        System.out.println("Subject "+(i+1)+": "+finalMarks[i]);
        }
}

// Main Class
import SEE.externals;
class Main
{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].inputDetails();
                        System.out.println("Enter CIE marks:");
                        finalMarks[i].CIEmarks();
                        System.out.println("Enter SEE marks:");
                        finalMarks[i].SEEMarks();
                }
                System.out.println("Displaying data:\n");
                for (int i=0;i<num;i++)
                {       finalMarks[i].calMarks();
                        finalMarks[i].displayDetails();
                        finalMarks[i].displayFinalMarks();
                }
        }
}
```

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.*;
class wrongAge extends Exception
{
	wrongAge(String s)
	{	super(s);
	}

}
class input
{
	int f_age;
	int s_age;
	Scanner sc= new Scanner(System.in);
}
class father extends input
{
	father() throws wrongAge
	{
		System.out.println("Enter father's age:");
		f_age = sc.nextInt();
		if(f_age < 0)
			throw new wrongAge("Age cannot be negative");
	}
	void display1(){
		System.out.println("Father's age is : "+f_age);
```

```
        }
}
class son extends father
{
        son() throws wrongAge
        {
                System.out.println("Enter son's age:");
                s_age = sc.nextInt();
                if (s_age > f_age)
                {
                        throw new wrongAge("Son's age cannot be greater than father's age");
                }
                else if (s_age <0)
                        throw new wrongAge("Age cannot be negative");
        }
        void display2(){
                System.out.println("Son's age is : "+s_age);
        }
}
class Main
{
public static void main (String args[]){
        try
        {
                son s =new son();
                s.display1();
                s.display2();
        }
        catch(wrongAge e){
                System.out.println("Error : "+e);
        }
}
}
```

LAB 8

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
import java.util.*;
class NewThread implements Runnable
{
String name;
Thread t;
NewThread(String threadname)
{
        name = threadname;
        t = new Thread(this , name);
        t.start();
}
public void run()
{
        try {
        while(true)
        {       System.out.println(name);
                Thread.sleep(2000);
        } }
        catch (InterruptedException e) {
        System.out.println(name + " Interrupted");
        }
}
}
class MainThread
{
public static void main(String args[])
{
        new NewThread("Computer Science Engineering");
```

```
        try
        {
                while(true){
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
                Syste m.out.print("\n");
        }
        }
        catch (InterruptedException e) {
        System.out.println("BMS Interrupted");
        }
System.out.println("Main thread exiting.");
}
}
```

LAB 9

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

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
   SwingDemo(){
      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 divident:");
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);
      JButton button = new JButton("Calculate");
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();
      jfrm.add(err);  // 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!");
                }} });
        jfrm.setVisible(true);
    }
    public static void main(String args[]){
        SwingUtilities.invokeLater(new Runnable(){
            public void run(){
                new SwingDemo();
            }});
}}
```

LAB 10

Demonstrate Inter process Communication and deadlock

```
// InterProcess Communication
class Q
{
        int n;
        boolean valueSet = false;
        synchronized int get()
        {
                while(!valueSet)
                try {
                System.out.println("Consumer waiting");
                wait();
        } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
        }
        synchronized void put(int n) {
        while(valueSet)
        try {
        System.out.println("Producer waiting");
        wait();
        } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
        }
        this.n = n;
        valueSet= true;
```

```
        System.out.println("Put: " + n);
        System.out.println("Intimate Consumer");
        notify();
        }
}
class Producer implements Runnable {
        Q q;
        Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
        }
        public void run() {
        int i = 0;
        while(i<5) {
        q.put(i++);
        }
        }
}
class Consumer implements Runnable {
        Q q;
        Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
        }
        public void run() {
        int i=0;
        while(i<5) {
        int r=q.get();
        i++;
        }
}}
class 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 fruits(B b)
        {
                String name = Thread.currentThread().getName();
                System.out.println(name + " entered A.fruits");
                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 veggies(A a)
        {
                String name = Thread.currentThread().getName();
                System.out.println(name + " entered B.veggies");
```

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