

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab Bachelor of
Engineering
in
Computer Science and Engineering

Submitted by:

Prakruthi B S

(2023BMS02609)

Department of Computer Science and Engineering,
B.M.S College of Engineering,
Bull Temple Road, Basavanagudi, Bangalore, 560019
2023-2024.

INDEX

Sl.No.	Title Date	Complete scanned Observation Book
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

Lab Program 1
quadratic equation.

import java.util.Scanner;

class quadratic

{

int a, b, c;

double x1, x2, 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)

{

$\alpha_1 = (-b) / (2 * a);$

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

System.out.println("Root1 = Root2 = " + α_1);

}

else if (d > 0)

{

$\alpha_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$

$\alpha_2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$

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

System.out.println("Root1 = " + α_1 + " Root2 = "
+ α_2);

}

else if (d < 0)

{

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

$\alpha_1 = (-b) / (2 * a);$

$\alpha_2 = \text{Math.sqrt}(-d) / (2 * a);$

System.out.println("Root1 = " + α_1 + " + i " + α_2);

System.out.println("Root2 = " + α_1 + " - i " + α_2);

}

}

}

class quadratic main .

{

public static void main (String args[])

{

quadratic q = new quadratic();

q.getd();

q.compute();

}

y.

Output:

Enter the coefficients of a,b,c.

2, 5, 7.

Roots are Imaginary

Root1 = -1.0 + 11.3919410907075054

Root1 = -1.0 - 11.3919410907075054.

9/12/23

PAGE NO.:
DATE:

Lab program 2.

```
import java.util.Scanner;  
class Subject  
{
```

```
    int subjectmarks;
```

```
    int credits;
```

```
    int grade;
```

```
}
```

```
class Student
```

```
{
```

```
    String name;
```

```
    String usn;
```

```
    double CGPA;
```

```
    Scanner s;
```

```
    Subject[] subjects;
```

```
Student()
```

```
{
```

```
    int p;
```

```
    subjects = new Subject[9];
```

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

```
{
```

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

```
}
```

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

```
void getStudentDetails()
```

```
{
```

```
    System.out.println("Enter student name:");
```

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

```
    System.out.println("Enter student usn:");
```

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

```

y
void getMarks()
{
    for (int i=0; i<9; i++)
    {
        System.out.println("Enter marks for"
                            + " subjects" + (i+1) + ":");
        subjects[i].Subjectmarks = sc.nextInt();
        subjects[i].Credits = 4;
        if (subjects[i].Subjectmarks >= 90)
            subjects[i].Grade = 10;
        else if (subjects[i].Subjectmarks >= 75)
            subjects[i].Grade = 9;
        else if (subjects[i].Subjectmarks >= 60)
            subjects[i].Grade = 8;
        else if (subjects[i].Subjectmarks >= 50)
            subjects[i].Grade = 7;
        else if (subjects[i].Subjectmarks >= 40)
            subjects[i].Grade = 6;
        else
            subjects[i].Grade = 0;
    }
}

```

void computeSPPA()

```

{
    double totalCredits = 0;
    double totalGradePoints = 0;
    for (int i=0; i<9; i++)
    {
        totalGradePoints += subjects[i].Grade *

```

subject[i].credits;

totalCredits += subject[i].credits;

}

SGPA = total gradepoints/total credits;

}

}

class main

{

public static void main(String args[])

{

Student s1 = new Student();

s1.getStudentDetails();

s1.getMarks();

s1.computeSGPA();

System.out.println("In Student details");

System.out.println("Name:" + s1.name);

System.out.println("UEN:" + s1.uen);

System.out.println("SGPA:" + s1.SGPA);

}

}

Output:

Enter Student name: "Prakeshthi"

Enter Student UEN: 1202813MS0609

Enter marks for subject 1: 60

Enter marks for subject 2: 80

Enter marks for subject 3: 75

Enter marks for subject 4: 65

Enter marks for subject 5: 85

PAGE NO:
DATE:

Enter marks for subject 6 : 70

Enter marks for subject 2 : 90

Enter marks for subject 8 : 50

Enter marks for subject 9 : 70

Student details

Name: Prakarushi

USN: 2023BMS0609

CGPA: 8.55

06/09/2023
19.12.2023

26/12/23 Lab Program 3

PAGE NO:
DATE:

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

{

```
String name;
```

```
String author;
```

```
int price;
```

```
int numPages;
```

```
Book (String name, String author, int price,  
int numPages)
```

{

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

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

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

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

}

```
public String toString()
```

{

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

```
name = "Book name: " + this.name + "\n";
```

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

```
price = "Price: " + this.price + "\n";
```

```
numPages = "Number of pages: " + this.numPages  
          + "\n";
```

```
return name + author + price + numPages;
```

}

}

Class Main

{

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

Scanner s = new Scanner (System. in);

System.out.print ("Enter number of books: ")

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

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

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

```
System.out.print("Enter name of the book: ");
```

```
String name = s.next();
```

```
System.out.print("Enter author of the  
books: \n");
```

```
String author = s.next();
```

```
System.out.print("Enter the price of  
book: \n");
```

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

```
System.out.print("Enter the number of  
pages of book: \n");
```

```
int numPages = s.nextInt();
```

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

3

```
System.out.println("In Book Details: ");
```

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

2

```
System.out.print("In Bo
```

```
("Book") + (i+1) + ":" + books[i]);
```

3

3

Output:

Enter number of books:

2

Enter name of the book: COA.

Enter author of the book: David

Enter the price of book: 600

Enter the number of pages in book: 1200

Enter name of the book: 003

Enter author of the books: John

Enter the price of book: 750

Enter the number of pages in book: 950

BOOK details:

Book 1

Book name: COA

author

Book name: DAVID

Price: 600

Number of Pages: 1200

Book 2

Book name: 003

Author name: JOHN

Price: 750

Number of pages: 950

21/12/21
Lab program 4.

import java.util.Scanner;
abstract class Shape

{

int a, b;

Shape (int a, int b)

{

this.a = a;

this.b = b;

{

abstract void printarea();

{

class Rectangle extends Shape

{

Rectangle (int a, int b)

{

super(a, b);

{

void printarea()

{

System.out.println("Area of rectangle : " + (a * b));

{

{

class Triangle extends Rectangle Shape

{

Triangle

Triangle (int a, int b)

{

super(a, b);

{

void printarea()

{

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

}

{

```
class circle extends Comparable<Shape> {
```

{

```
circle (int a, int b)
```

{

```
super (a, b);
```

}

```
void printarea()
```

{

```
System.out.println("Area of circle :"  
+ (3.14 * a * a));
```

}

{

```
class Main
```

{

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

{

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

```
System.out.println ("Enter length and breadth  
of Rectangle :");
```

```
int length = s.nextInt();
```

```
int breadth = s.nextInt();
```

```
Rectangle rectangle = new Rectangle (length,  
breadth);
```

```
System.out.println ("Enter base and height  
of triangle :");
```

```
int base = s.nextInt();
```

int height = s.nextInt();

Triangle triangle = new Triangle(base, height);

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

int radius = s.nextInt();

Circle circle = new Circle(radius, radius);

rectangle.printArea();

triangle.printArea();

circle.printArea();

3

3

Output:

Enter length and breadth of Rectangle:

2 3

Enter base and height of Triangle:

2 4

Enter radius of circle:

3

Area of rectangle: 6

Area of triangle: 4.0

Area of circle: 28.2599999999.

8
21/12/14

9/01/24 Lab programs

Import java.util.Scanner;

class BankAccount {

protected String customerName;

protected int accountNumber;

protected double balance;

public BankAccount (String customerName,
int accountNumber, double balance)

{

this.customerName = customerName;

this.accountNumber = accountNumber;

this.balance = balance;

}

public void displayAccountDetails()

{

System.out.println("Customer name: " +
+ customerName);

System.out.println("Account number: "
+ accountNumber);

System.out.println("Type of Account: Savings
Account");

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

y

)

class SavingsAccount extends BankAccount

{

private double interestRate;

public SavingsAccount (String customerName,
int accountNumber, double balance, double

InterestRate)

{

super(customerName, accountNumber, balance);
this.interestRate = interestRate;

}

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

}

public void withdraw(double amount) {
balance -= amount;

}

public void computeInterest() {
balance += balance * interestRate / 100;

}

public void displayAccountDetails()
Super.displayAccountDetails();

System.out.println("Interest Rate: " + interestRate
+ "%");

}

{

class MyBank {

public static void main(String[] args) {

{

Scanner scanner = new Scanner(System.in);

SavingsAccount[] accounts = new SavingsAccount[2];
int choice;

String customerName;

int accountNumber;

double balance;

double depositAmount;

double withdrawalAmount;

double interestRate;

```
for(int i=0; i<2; i++)
```

{

```
System.out.println("Enter customer Name:");
```

```
customerName = scanner.nextLine();
```

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

```
accountNumber = scanner.nextInt();
```

```
System.out.println("Enter the deposit amount.");
```

```
balance = scanner.nextDouble();
```

```
accounts[i] = new SavingsAccount(customerName,  
accountNumber, balance, 2);
```

}

do {

```
System.out.println("\n--- MENU ---");
```

```
System.out.println("1. Deposit");
```

```
System.out.println("2. withdraw");
```

```
System.out.println("3. compute Interest for  
Savings Account");
```

```
System.out.println("4. Display Account details");
```

```
System.out.println("5. exit");
```

```
System.out.println("Enter your choice:");
```

```
choice = scanner.nextInt();
```

switch(choice) {

case 1:

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

```
String accountType = scanner.nextLine();
```

```
System.out.println("Enter the deposit  
amount:");
```

```
depositAmount = scanner.nextDouble();
```

account[0].deposit(depositAmount);
break;

case 2:

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

accountType = scanner.nextLine();

System.out.println("Enter the withdrawal
amount:");

wWithDrawalAmount = scanner.nextDouble();
accounts[0].withdraw(wWithDrawalAmount);
break;

case 3:

accounts[0].computeInterest();
break;

case 4:

accounts[0].displayAccountDetails();

break;

case 5:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice.
Please try again.");

} }

while(choice != 5);

Scanner.close();

}

}

Output

Enter customer Name: prakriti

Enter account number: 112

Enter the deposit amount: 20000

Enter customer Name: Rachana.

Enter account number: 213.

Enter the deposit amount: 30000.

- - - MENU - - -

1. deposit

2. withdraw

3. Compute Interest for savings Account

4. display account details

5. exit.

Enter your choice: 1

Enter the type of account: Savings

Enter the deposit amount: 30000

- - - MENU - - -

1. Deposit

2. withdraw

3. Compute Interest for Savings Account

4. display account details

5. exit

Entered your choice: 3.

- - - Menu - - -

1. Deposit

2. withdraw

3. Compute Interest for Savings Account

4. display account details.

c.exit.

Enter your choice : 4

Customer Name : pradeeputhi

Account number : 112

Type of Account : Savings Account

Balance - 53000.0

Interest Rate : 02.0%

1/2/0
09.01.24
09.01.24

23/1/24 Lab program 6

PAGE NO:
DATE:

package ctc;

import java.util.Scanner;
public class Student1.

{

protected String usn = new String();

protected String name = new String();

protected int sem;

public void InputStudentDetails()

{

Scanner s = new Scanner(System.in);

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

usn = s.nextInt();

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

name = s.next();

System.out.println("Enter the student sem:");

sem = s.nextInt();

}

public void displayStudentDetails()

{

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

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

System.out.println("Semester: " + sem);

g

g

```
package IEEE;
import java.util.Scanner;
public class Internal extends Student
{
    protected int marks[] = new int[5];
    public void input(IEEE)
    {
        Scanner s1 = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Enter internal marks");
            marks[i] = s1.nextInt();
        }
    }
}
```

```
package SEE;
import IEEE.Internal;
import java.util.Scanner;
public class External extends Internal
{
    protected int marks[];
    protected int finalmarks[];
    public External()
    {
        marks = new int[5];
        finalmarks = new int[5];
    }
    public void inputSEEmarks()
    {
        Scanner s = new Scanner(System.in);
    }
}
```

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

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

```
System.out.println("Subject" + (i+j) +  
    "marker");
```

marks(r) = segmentEnt(s);

۹

```
public void calculateFinalmarks()  
{
```

25

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

`finalmarks[i] = marks[i]/2 + super_marks[i];`

9

```
public void displayFinalmarks()  
{
```

2

displayStudentDetails();

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

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

۳

g

Import SITEextensible;

class mates

2

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

9

~~Per n numero studenti = 2^o~~

~~external finalMarker[]) = new~~

~~ew.~~ conflict

Q3d

QUESTION
DATE:

```
external [numofStudents];  
for (int i=0; i<numofStudents; i++)  
{  
    finalmarks[i] = new external();  
    finalmarks[i].InputStudentDetails();  
    System.out.println("Enter SEE marks " + (i+1));  
    finalmarks[i].InputC8Marks();  
    System.out.println("Enter SEE marks " + (i+1));  
    finalmarks[i].InputSEEmarks();  
    System.out.println("Displaying data: ");  
    for (int i=0; i<numofStudents; i++)  
    {  
        finalmarks[i].calculateFinalMarks();  
        finalmarks[i].displayFinalMarks();  
    }  
}
```

Output

Enter the student USn: 2023BMS02609

Enter the student name: Rakesh.

Enter the student sem: 3.

Enter CSE marks: 1

Enter Internal marks of CSE: 1

24

Enter Internal marks of CSE: 2

35

Enter Internal marks of CSE: 3

33

Enter Internal marks of CSE: 4

27.

Enter external marks of CSE:5

46.

Enter SIE marks 1 34

Subject 1 marks: 34

Subject 2 marks: 86

Subject 3 marks: 48

Subject 4 marks: 35

Subject 5 marks: 36.

Enter the student vno: 2023BMS02608

Enter the student name: Rachana

Enter the student sem: 3

Enter CSE marks 2.

Enter internal marks of CSE:1

24

Enter internal marks of CSE:2

25

Enter internal marks of CSE:3

26

Enter internal marks of CSE:4

27

Enter internal marks of CSE:5.

28

Enter SIE marks 2

Subject 1 marks: 32

Subject 2 marks: 34

Subject 3 marks: 33

Subject 4 marks: 26

Subject 5 marks: 38.

~~Displaying data~~

USN: 2023BMS02609

Name: Prakasanthi

Semester: 3

Subject 1: 41

Subject 2: 63

Subject 3: 55

Subject 4: 44

Subject 5: 64

USN: 2093BMS02608

Name: Rachana

Semester: 3 *

Subject 1: 40

Subject 2: 42

Subject 3: 42

Subject 4: 40

Subject 5: 43.

✓ Gen

30/1/20

Lab program 7

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

```
    public WrongAge()
```

```
    {
```

```
        super("Age error");
```

```
    }
```

```
    public WrongAge(String message)
```

```
    {
```

```
        super(message);
```

```
    }
```

```
}
```

```
class InputScanner  
{
```

```
    public Scanner S;
```

```
    public InputScanner()
```

```
    {
```

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

```
    }
```

```
}
```

```
class Father extends InputScanner  
{
```

```
    public int fatherage;
```

```
    public Father() throws WrongAge  
{
```

```
        System.out.println("Enter Father's Age:");
```

```
        fatherage=S.nextInt();
```

```
        if(fatherage<0)
```

```
    {
```

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

negative");
}

}

public void display()
{

System.out.println("Father's Age: "
+fatherage);

}

}

class Son extends Father

{

private int sonage;

public Son() throws wrongAge

{

System.out.println("Enter Son's Age:");

sonage = s.nextInt();

if (sonage > fatherage)

{

System.out.println("Son's age cannot be
greater than Father's age");

}

else if (sonage < 0)

{

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

}

public void display()
{

System.out.println("Son's Age: " + sonage);

}

3
class Main.

{
public static void main(String args[])

{
try {

Son son = new Son();

son.display();

}

catch (WrongAge e)

{

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

}

g

Output

Enter Father's Age: 23

Enter Son's Age: 45

Son's age cannot be greater than
father's age.

Son's Age: 45

✓
Date
15/01/2014

6/2/24

PAGE NO :
DATE :

Lab program 8.

class college extends thread
{

 public void run()

 { for(int i=0; i<6; i++) {

 System.out.println("BMS college of
 engineering");

 try

 {

 Thread.sleep(10000);

 }

 catch (InterruptedException e)

 {

 e.printStackTrace();

 }

 }

}

class Dept extends Thread

{

 public void run()

 {

 for(int i=1; i<=10; i++)

 {

 System.out.println("CSF");

 try {

 Thread.sleep(3000);

 }

 catch (InterruptedException e)

 {

```
e. print stacktrace();
```

{
y
y
y

```
class Main
```

{
public static void main(String args[]){
 college c1 = new college();
 c1.start();
 Dept d1 = new Dept();
 d1.start();
}
y.

```
college c1 = new college();
```

```
c1.start();
```

```
Dept d1 = new Dept();
```

```
d1.start();  
}  
y.
```

Output

BMS college of engineering

CSE

CSE

CSE

CSE

BMS college of engineering

CSE

CSE

CSE

CSE

CSE

BMS college of engineering

BMS college of engineering

Lab program 9

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

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

        JLabel qlab = new JLabel("Enter the divisor  
and dividend:");
        JTextField qtf = new JTextField(8);
        JTextField btf = new JTextField(8);
        JButton button = new JButton("calculate");
        JLabel eror = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        ifrm.add(qtf);
        ifrm.add(qlab);
        ifrm.add(btf);
        ifrm.add(button);
        ifrm.add(alab);
        ifrm.add(blab);
        ifrm.add(eror);
    }
}
```

```
frm.add(anslab);
```

```
ActionListener l = new ActionListener()
{
```

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

```
        System.out.println("Action event from a
text field");
```

```
}
```

```
g;
```

```
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("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 (ArithmaticException e)

{

alab.setText(" ");

blab.setText(" ");

anslab.setText(" ");

esn.setText("B should be Non zero.");

{

{

});

if (fm.setVisible(true));

{

public static void main (String args[])

{

SwingUtilities.invokeLater (new Runnable())

{

public void run()

{

new swingDemo();

{

};

{

}.

Output

Enter only integers!

Enter the divisor and dividend:

12

01

calculate.

B should be now zero!

Enter the divider and dividend:

12

0

calculate.

Enter the divider and dividend:

12

3

calculate. $A=12$ $B=3$ Ans = 4.

Lab program 10

Incorrect implementation of a producer and consumer.

class Q {

 int n;

 synchronized int get() {

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

 return n;

}

 synchronized void put(int n) {

 this.n = n;

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

}

}

class Producer implements Runnable {

 Q q;

 Producer(Q q) {

 this.q = q;

 new Thread(this, "Producer").start();

}

 public void run() {

 int i = 0;

 while(i < 15) {

 q.put(i++);

}

}

class Consumer implements Runnable {

 Q q;

 Consumer(Q q) {

this. q = q;

new thread (this, "consumer"), starts;

}

public void run() {

int i = 0;

while (i < 5) {

int x = q.get();

i++;

}

}

}

class PC {

public static void main(String args[])

{

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

Put: 1

Put: 2

Put: 3

Put: 4

Put: 5

Put: 6

put: 7

put: 8

put: 9

put: 10

put: 11

put: 12

put: 13

put: 14.

ejot: 14

ejot: 14

ejot: 14

ejot: 14

~~ejot: 14~~

13/02/15

PAGE NO.:

DATE:

contd Lab Program 10.

class Q,

{

int n;

boolean valueset = false;

synchronized int get()

{

while (!valueset)

try {

System.out.println("Consumer waiting");

wait();

g

catch (InterruptedException e)

{

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

g

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

valueset = false;

System.out.println("Estimate Producer");

notify();

return n;

g

synchronized void put(int n)

{

while (valueset)

try {

System.out.println("Producer waiting");

wait();

g

catch (InterruptedException e)

{

System.out.println("An interrupted exception
caught");

}

this.n=n;

Value.set=true;

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

System.out.println("Intimate consumer");

notify();

}

}

Class Producer implements Runnable

{

q=q;

Producer(q)

{

this.q=q;

new Thread(this, "Producer").start();

}

public void run()

{

int i=0;

while(i<15)

{

q.put(i++);

}

}

}

Class Consumer implements Runnable

{

q=q;

Consumer(q)

{

this.q = q;

new Thread(this, "consumer").start();

y

public void run()

{

int i=0;

while(i<15)

{

int u=q.get();

System.out.println("consumed "+u);

i++;

y

g

g

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.");

y

g

Output

Press control-c to stop.

Put: 0

Intimate consumer

Producers 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

Put: 4. consumed: 3

Put: 4

Intimate consumer

Producer waiting

got: 4

Intimate producer

consumed: 4

Put: 5

Intimate consumer

Producer waiting

got: 5

Intimate producer consumed: 5

10b) Deadlocks

class A {

 synchronized void foo(B b)

 {

 String name = Thread.currentThread().getName();

 System.out.println(name + " entered A.foo");

 try

 {

 Thread.sleep(1000);

 }

 catch (Exception e)

 {

 System.out.println("A interrupted");

 }

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

 b.last();

 }

 void last()

 {

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

 }

 class B

 {

 synchronized void bar(A a)

 {

 String name = Thread.currentThread().getName();

 System.out.println(name + " entered B.bar");

 try {

 Thread.sleep(1000);

{

catch (Exception e)

{

System.out.println("B interrupted");

{

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

a.last();

{

void last()

{

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

{

{

class 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.pool(b);

System.out.println("Back in Main Thread");

{

public void run()

{

b.back(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.

Main thread trying to call B.last();

Inside A.last

Racing thread trying to call A.last();

Back in main thread.

Inside A.last

Back in other thread.

8/13/2021

WEEK 1

- 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.**

Program:

```
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 + " 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();
    }
}
```

WEEK2

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.

Program:

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

```
class Subject
```

```
{
```

```
    int subjectmarks;
```

```
    int credits;
```

```
    int grade;
```

```
}
```

```
class Student
```

```
{
```

```
    String name;
```

```
    String usn;
```

```
    double SGPA;
```

```
    Scanner s;
```

```
    Subject[] subjects;
```

```
Student()
```

```
{
```

```
    int i;
```

```
    subjects=new Subject[9];
```

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

```
{
```

```
subjects[i]=new Subject();
}

s=new Scanner(System.in);
}

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

void getMarks(){
for(int i=0; i<9; i++)
{
System.out.println("Enter marks for subject"+(i+1)+":");
subjects[i].subjectmarks=s.nextInt();
subjects[i].credits=4;
if(subjects[i].subjectmarks>=90){
    subjects[i].grade=10;}
else if(subjects[i].subjectmarks>=75){
    subjects[i].grade=9;}
else if(subjects[i].subjectmarks>=60){
    subjects[i].grade=8;}
else if(subjects[i].subjectmarks>=50){
    subjects[i].grade=7;}
else if(subjects[i].subjectmarks>=40){
    subjects[i].grade=6;}
else{
```

```
subjects[i].grade=0;
}
}
}

void computeSGPA()
{
    double totalcredits=0;
    double totalgradepoints=0;
    for(int i=0; i<9; i++){
        totalgradepoints += subjects[i].grade*subjects[i].credits;
        totalcredits += subjects[i].credits;
    }
    SGPA= totalgradepoints/totalcredits;
}

class Main
{
    public static void main(String args[])
    {
        Student s2=new Student();
        s1.getStudentdetails();
        s1.getMarks();
        s1.computeSGPA();

        System.out.println("\n Student details");
        System.out.println("Name:"+s1.name);
        System.out.println("usn:"+s1.usn);
        System.out.println("SGPA:"+s1.SGPA);
    }
}
```

}

WEEK3

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.

Program:

```
import java.util.Scanner;
class Book
{
    String name;
    String author;
    int price;
    int numPages;

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

    public String toString()
    {
        String name, author, price, numPages;
```

```
name="Book name:"+this.name+"\n";
author="Author name:"+this.author+"\n";
price="Price:"+this.price+"\n";
numPages="Number of pages:"+this.numPages+"\n";
return name+author+price+numPages;
}
}
```

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

            books[i]=new Book(name,author,price,numPages);
        }
    }
}
```

```
System.out.println("\n Book Details:");
for(int i=0; i<n; i++)
{
    System.out.println("Book" +(i+1)+":\n"+ books[i]);
}

}
```

WEEK4

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.

Program:

```
import java.util.Scanner;
abstract class Shape1
{
    int a,b;
    Shape1(int a,int b)
    {
        this.a=a;
        this.b=b;
    }
    abstract void printarea();
}
class Rectangle extends Shape1
{
    Rectangle(int a, int b)
    {
        super(a,b);
    }
    void printarea()
    {
```

```
System.out.println("Area of rectangle:"+ (a*b));
}

}

class Triangle extends Shape1

{
    Triangle(int a, int b)
    {
        super(a,b);
    }

    void printarea()
    {
        System.out.println("Area of trangle:"+(0.5*a*b));
    }
}

class Circle extends Shape1

{
    Circle(int a, int b)
    {
        super(a,b);
    }

    void printarea()
    {
        System.out.println("Area of circle:"+(3.14*a*a));
    }
}

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

```
{  
    Scanner s=new Scanner(System.in);  
    System.out.println("Enter length and breadth of Rectangle: ");  
    int length = s.nextInt();  
    int breadth = s.nextInt();  
    Rectangle rectangle = new Rectangle(length, breadth);  
    System.out.println("Enter base and height of Triangle: ");  
    int base = s.nextInt();  
    int height = s.nextInt();  
    Triangle triangle = new Triangle(base, height);  
    System.out.println("Enter radius of Circle: ");  
    int radius = s.nextInt();  
    Circle circle = new Circle(radius, radius);  
    rectangle.printarea();  
    triangle.printarea();  
    circle.printarea();  
}  
}
```

WEEK5

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.

Program:

```
import java.util.Scanner;

class BankAccount {
    protected String customerName;
    protected int accountNumber;
    protected double balance;

    public BankAccount(String customerName, int accountNumber, double balance) {
        this.customerName = customerName;
```

```
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public void displayAccountDetails() {
        System.out.println("Customer name: " + customerName);
        System.out.println("Account number: " + accountNumber);
        System.out.println("Type of Account: Savings account");
        System.out.println("balance - " + balance);
    }
}

class SavingsAccount extends BankAccount {
    private double interestRate;

    public SavingsAccount(String customerName, int accountNumber, double balance,
double interestRate) {
        super(customerName, accountNumber, balance);
        this.interestRate = interestRate;
    }

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

    public void withdraw(double amount) {
        balance -= amount;
    }
}
```

```
public void computeInterest() {  
    balance += balance * interestRate / 100;  
}  
  
@Override  
public void displayAccountDetails() {  
    super.displayAccountDetails();  
    System.out.println("Interest Rate: " + interestRate + "%");  
}  
}  
  
class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        SavingsAccount[] accounts = new SavingsAccount[2];  
        int choice;  
        String customerName;  
        int accountNumber;  
        double balance;  
        double depositAmount;  
        double withdrawalAmount;  
        double interestRate;  
  
        for (int i = 0; i < 2; i++) {  
            System.out.print("Enter customer name: ");  
            customerName = scanner.next();  
            System.out.print("Enter account Number: ");  
            accountNumber = scanner.nextInt();  
            System.out.print("Enter the deposit amount: ");  
        }  
    }  
}
```

```
balance = scanner.nextDouble();
accounts[i] = new SavingsAccount(customerName, accountNumber, balance,
2);
```

```
}
```

```
do {
    System.out.println("\n-----MENU-----");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Compute interest for Savings Account");
    System.out.println("4. Display account details");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = scanner.nextInt();
```

```
switch (choice) {
    case 1:
        System.out.print("Enter the type of account: ");
        String accountType = scanner.next();
        System.out.print("Enter the deposit amount: ");
        depositAmount = scanner.nextDouble();
        accounts[0].deposit(depositAmount);
        break;
```

```
    case 2:
        System.out.print("Enter the type of account: ");
        accountType = scanner.next();
        System.out.print("Enter the withdrawal amount: ");
        withdrawalAmount = scanner.nextDouble();
        accounts[0].withdraw(withdrawalAmount);
```

```
        break;

    case 3:
        accounts[0].computeInterest();
        break;

    case 4:
        accounts[0].displayAccountDetails();
        break;

    case 5:
        System.out.println("Exiting...");
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
    }

}

while (choice != 5);
scanner.close();
}
```

WEEK6

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.

Program:

STUDENT FILE

```
package CIE;
import java.util.Scanner;
public class Student1
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the Student usn:");
        usn=s.next();
```

```
System.out.println("Enter the Student name: \n");
name=s.nextLine();
System.out.println("Enter the Student sem: \n");
sem=s.nextInt();
}
public void displayStudentDetails()
{
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Semester: " + sem);
}
}
```

INTERNAL FILE

```
package CIE;
import java.util.Scanner;
public class internal extends Student1
{
    protected int marks[] = new int[5];
    public void inputCIEmarks()
    {
        Scanner s1=new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Enter internal marks of
```

```
CIE:"+(i+1));  
    marks[i]=s1.nextInt();  
}  
  
}  
}
```

EXTERNAL FILE

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

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

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

MAIN FILE

```
import SEE.external;
class Main
{
    public static void main(String args[])
    {
        int numOfStudents = 2;
        external finalMarks[] = new
        external[numOfStudents];
        for(int i=0;i<numOfStudents; i++)
        {
            finalMarks[i] = new external();
            finalMarks[i].inputStudentDetails();
```

```
System.out.println("Enter CIE marks"+(i+1));
finalMarks[i].inputCIEmarks();
System.out.println("Enter SEE marks"+(i+1));
finalMarks[i].inputSEEmarks();
}
System.out.println("Displaying data:\n");
for(int i=0;i<numOfStudents; i++)
{
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
}
}
}
```

WEEK7

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.

Program:

```
import java.util.Scanner;

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

class Father {
    int fatherAge;

    Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}
```

```
    }

void display() {
    System.out.println("Father's age is: " + fatherAge);
}

}
```

```
class Son extends Father {
```

```
    int sonAge;
```

```
    Son() throws WrongAge {
```

```
        super();
```

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

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

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

```
        if (sonAge >= fatherAge) {
```

```
            throw new WrongAge("Son's age cannot be greater than father's age");
```

```
        } else if (sonAge < 0) {
```

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

```
        }
```

```
}
```

```
    void display() {
```

```
        super.display();
```

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

```
}
```

```
}
```

```
class Main {
```

```
public static void main(String[] args) {  
    try {  
        Son s = new Son();  
        s.display();  
    } catch (WrongAge e) {  
        System.out.println(e.getMessage());  
    }  
}
```

WEEK8

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.

Program:

```
class College extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            System.out.println("BMS COLLEGE OF ENGINEERING");  
            try {  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
class Dept extends Thread {  
    public void run() {  
        for (int i = 1; i <= 25; i++) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
    }
}

}

}

class Main {
    public static void main(String args[]) {
        College c1 = new College();
        c1.start();
        Dept d1 = new Dept();
        d1.start();
    }
}
```

WEEK9

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

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

Program:

```
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);
        jfrm.add(jlab);
        jfrm.add(ajtf);
```

```

jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

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

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

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

            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
        }
        catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        }
        catch(ArithmaticException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});

```

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

WEEK10

10. Demonstrate Inter process Communication and deadlock

Program:

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

B.InterprocessCommunication:

```
class Q {  
  
    int n;  
  
    boolean valueSet = false;  
  
    synchronized int get() {  
  
        while(!valueSet)  
  
        try {  
  
            System.out.println("\nConsumer waiting\n");  
  
            wait();  
  
        } catch(InterruptedException e) {  
  
            System.out.println("InterruptedException caught");  
        }  
    }  
}
```

```
}

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

valueSet = false;

System.out.println("\nIntimate Producer\n");

notify();

return n;

}

synchronized void put(int n) {

while(valueSet)

try {

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

wait();

} catch(InterruptedException e) {

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

}

}
```

```
this.n = n;  
  
valueSet = true;  
  
System.out.println("Put: " + n);  
  
System.out.println("\nIntimate Consumer\n");  
  
notify();  
  
}  
  
}  
  
class Producer implements Runnable {  
  
    Q q;  
  
    Producer(Q q) {  
  
        this.q = q;  
  
        new Thread(this, "Producer").start();  
  
    }  
  
    public void run() {  
  
        int i = 0;
```

```
while(i<15) {  
  
    q.put(i++);  
  
}  
  
}  
  
}  
  
class Consumer implements Runnable {  
  
    Q q;  
  
    Consumer(Q q) {  
  
        this.q = q;  
  
        new Thread(this, "Consumer").start();  
  
    }  
  
    public void run() {  
  
        int i=0;  
  
        while(i<15) {  
  
            int r=q.get();
```

```
System.out.println("consumed:"+r);

i++;

}

}

class PCFixed {

public static void main(String args[]) {

Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}

}
```

