

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



LAB REPORT

23CS3PCOOJ

Submitted in partial fulfillment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

SUJNYAN KINI

1BM22CS340

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-24

INDEX

Sl.No.	Title	Date
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	23/01/2024
8	Lab 7	30/01/2024
9	Lab 8	06/02/2024
10	Lab 9	20/02/2024
11	Lab 10	13/02/2024

Lab Program 1:

Date _____

Page _____

```
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 values for  
        a, b, c");  
  
        a = s.nextInt();  
        b = s.nextInt();  
        c = s.nextInt();  
    }  
  
    void compute()  
    {  
        while (a == 0)  
        {  
            System.out.println ("Not a quad. equa.");  
            System.out.println ("Enter a non zero value");  
            Scanner s = new Scanner (System.in);  
            a = s.nextInt();  
        }  
    }  
}
```

Date _____
Page _____

Output:

Enter
roots
root1
root2
r1
r2

$$d = b * b - 4 * a * c;$$

if ($d == 0$)

$$r1 = (-b) / (2 * a);$$

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

System.out.println("Root1 = Root2 = " + r1);

}

else if ($d > 0$)

{

$$r1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$$

$$r2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2 * a);$$

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

System.out.println("Root1 = " + r1 + " Root2 = " + r2);

}

else if ($d < 0$)

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

$$r1 = (-b) / 2 * a;$$

$$r2 = \text{Math.sqrt}(-d) / (2 * a);$$

System.out.println("Root1 = " + r1 + " + i" + r2);

System.out.println("Root2 = " + r1 + " + i" + r2);

}

)

class Quadratic Main

{ public static void main (String args[])

{ Quadratic q = new Quadratic();

q.getd();

q.compute();

)

Output:

Enter coefficient abc : 1 5 2
roots are real & distinct

$$\text{root}_1 = 4.5615328$$

$$\text{root}_2 = 4.56155281$$

Date _____
Page _____

Check if the given number is prime number or not.

```
import java.util.Scanner;  
class isprime  
{  
    static void isprime (int n)  
    {  
        int i, m= 0, flag = 0;  
        m = n/2;  
        if (n == 0 || n == 1)  
        {  
            System.out.println(n + " is not a prime  
            no");  
        }  
        else  
        {  
            for (i=2; i<m; i++)  
            {  
                if (n % i == 0)  
                {  
                    System.out.println(n + " is not a prime  
                    number");  
                    flag = 1;  
                    break;  
                }  
            }  
            if (flag == 0)  
            {  
                System.out.println(n + " is a prime  
                number");  
            }  
        }  
    }  
}
```

public sta
int i
Scann
Syste
i = s
ispri
y.
y.

Output

Enter
5
It is

LAB

import
class
in
in
S

3
class
x

St
St
d

number

```
public static void main (String args[])
{
    int i;
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the value of
    i: ");
    i = sc.nextInt();
    isprime (i);
}
```

Output:-

Enter the value of i
5

It is a prime number.

LAB PROGRAM②

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

int subjectMarks;

int credits;

String grade;

}

```
class Student
```

{

String name;

String usn;

double SGPA;

Scanner s;

Subject subject [];

Student ()

{

int i
subject = new subject [9]

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

subject [i] = new subject ();

s8 = new Scanner (System.in);

}

void getStudentDetails ()

{

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

name = s.nextLine();

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

usn = s.nextInt();

}

void getMarks ()

{

int i;

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

{

System.out.println ("Enter the marks &
credits for course " + i + ": ");

System.out.println ("marks : ");

int creditmarks = s.nextInt();

System.out.println ("credits : ");

int credits = s.nextInt();

subject [i].~~subject~~subjectMarks = marks;

subject [i].subjectMarks = credit;

if (marks >= 90)

{

subject [i].grade = 'O';

}

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

y
j

void computeSGPA()

{

int i;

double SGPA;

double totalcredits = 0;

double totalgp = 0;

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

{

totalcredits += subject[i].credits;

switch(subject[i].grade)

{

case "O": totalgp += 10 * subject[i].credits;

break;

case "A+": totalgp += 9 * subject[i].credits;

break;

case "A": totalgp += 8 * subject[i].credits;

break;

case "B+": totalgp += 7 * subject[i].credits;

break;

Date _____
Page _____

case "B": totalgp += 6 * subject[i].credit,
break;

case "C": totalgp += 5 * subject[i].credit,
break;

case "F": totalgp += 0 * subject[i].credit,
break;

}

}

Sgpa = totalgp / total credits;

System.out.println ("The SGPA is : "+ Sgpa);

}

}

class sgpa

public static void main (String [] args){

Student s1 = new Student();

s1.getStudentDetails();

s1.getMarks();

s1.computeSGPA();

}

}

Output:

Copy

Enter your name:

Sujnyan

Enter your ucn:

IBM22EC264

Enter the marks and credits for course 0:

Marks:

90

Credits:

4

Enter the marks and credits for course 1:

Marks :

91

Credits :

4

Enter the

Marks :

91

Credits :

3

Enter t

Marks

93

Credit

2

Enter

Marks

90

Credit

01

Enter

Mar

92

Credit

Marks :

91

Credits :

4

Enter the marks & credits for course 3 :

Marks :

91

Credits :

3

Enter the marks & credit for course 4 :

Marks :

93

Credits :

2

Enter the marks & credits for course 5 :

Marks : 90

90

Credits :

01

Enter the marks & credit for course 6 :

Marks :

92

Credits :

01

Enter the marks & credits for course 7 :

Marks :

90

Credits :

02

The SGPA is 10.000000.

LAB - 3

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

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

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

```
public class Mainbooks
```

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

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

int i;

String name;

String author;

int price;

int numPages;

System.out.println("Enter the number of books")

n = s.nextInt();

Books b[];

b = new Book [n];

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

{

System.out.println("Enter the details of
books" + (i+1) + ":"),

System.out.println("Enter the name of the
book :"),

name = s.next();

System.out.println("Enter the author name"),
author = s.next();

System.out.println("Enter the price"),
price = s.nextInt();

System.out.println("Enter the number of
pages :"),

numPages = s.nextInt();

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

↓

System.out.println("Book Details"),

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

{

System.out.println(b[i]),

}

Output

Enter the number of books:

2

Enter the details of book 1:

Enter the name of the book

Secret

Enter the author name

William

Enter the price:

350

Enter the number of pages

123

Enter the details of book 2:

Enter the name of the book
revolution.

Enter the author name:

Chetan

Enter price:

200

Enter no. of pages:

304

Enter the details of Book 3:

Enter name of Book 3:

secret

Enter author name:

William

Enter price:

300

Enter no. of pages:

150

Book Details

Book name: Revolution

Book author name = Chetan

Price : 200

Pages : 704

Lab Program

import java

class input

{

protected

public

s = n

3

public

*

5

y

}

abstract

{

pro

pub

L

ab

o

l

cl

ss

*

Lab Program 4.

Date _____
Page _____

```
import java.util.Scanner;  
class inputScanner  
{  
    protected Scanner s;  
    public inputScanner()  
    {  
        s = new Scanner(System.in);  
    }  
    public int getInput(String message)  
    {  
        System.out.println(message);  
        return scanner.nextInt();  
    }  
    abstract class Shape extends inputScanner  
{  
        protected int a, b;  
        public Shape()  
        {  
            super();  
        }  
        abstract public void printArea();  
    }  
    class Rectangle extends Shape  
{  
        protected int a, b;  
        public Rectangle()  
        {  
            super();  
        }  
        public void printArea()  
        {  
            a = getInput("Enter the length:");  
            b = getInput("Enter the breadth:");  
        }  
    }
```

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

↓
class Triangle extends Shape

↓
protected int a, b;
public Triangle ()
{
 super ();

↓
public void printArea ()

↓
a = getInput ("Enter the side1 : ");
b = getInput ("Enter the side2 : ");

double area = 0.5 * a * b;

System.out.println ("Area of the Triangle : "
 + area);

↓
class Circle extends Shape

↓
protected int a;
public Circle ()
{
 super ();

↓
public void printArea ()

a = getInput ("Enter the radius : ");
double area = 3.14 * a * a;

↓
System.out.println ("Area of the circle : "
 + area);

1

2

public class MainShape

3

4

public static void main (String [] args)

5 Rectangle r = new Rectangle ();

6 Triangle t = new Triangle ();

7 Circle c = new Circle ();

8 r.printArea();

9 t.printArea();

10 c.printArea();

11 3

12 1

Output:

Enter the length :

12 2

Enter the breadth :

13 4

14 Area of the Rectangle : 8

15 Enter side 1 :

16 2

17 Enter side 2 :

18 2

19 Area of the Triangle : 2.0

20 Enter the radius :

21 5

22 Area of the circle : 78.5

LAB PROGRAM 5:

```
import java.util.Scanner;  
class account  
{  
    String name;  
    int accno;  
    String type;  
    double balance;  
    account (String name, int accno, String  
             type, double balance)  
    {  
        this.name = name;  
        this.accno = accno;  
        this.type = type;  
        this.balance = balance;  
    }
```

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

```
void withdraw (double amount)  
{  
    if ((balance - amount) >= 0)  
    {  
        balance -= amount;  
    }  
    else  
    {  
        System.out.println ("Insufficient balance  
                           Can't withdraw");  
    }  
}
```

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

class savAcct extends account
{
    private static double rate = 5;
    SavAcct (String name, int accno, double
              balance) ,
    {
        super (name, accno, "savings", balance);
    }

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

class currAcct extends account
{
    private double minBal = 500;
    private double serviceCharges = 50;
    currAct (String name, int accno, double
              balance) ,
    {
        super (name, accno, "current", balance);
    }

    void checkmain()
    {
        if (balance < minBal)
    }
}
```

Date _____
Page _____

H

```
System.out.println ("balance is " +  
min balance : service charges imposed : " +  
servicecharges );  
balance -= servicecharges ;  
System.out.println ("balance is :" + balance);
```

3

```
class accountMain  
{  
    public static void main (String[] args)  
    {  
        Scanner s = new Scanner (System.in);  
        System.out.printin ("Enter the name:");  
        String name = s.next();  
        System.out.printin ("Enter the type:");  
        String type = s.next();  
        System.out.printin ("Enter the acc no:");  
        int acno = s.nextInt();  
        System.out.printin ("Enter initial balance:");  
        double balance = s.nextDouble();
```

```
int ch;  
double amt1, amt2;  
Account acc = new account (name, acno,  
                           type, balance)
```

⑧ SavAcct sa = new SavAcc (name, acno,
 balance);
CurrAcct ca = new currAcct (name, acno,
 balance);

while (true)

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

x

System.out.println("\nMenu\n 1. deposit
 2. withdraw 3. Compute Interest 4. display")

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

ch = s.nextInt();

switch (ch)

x

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

amt1 = s.nextInt();

sa.deposit(amt1);

break;

case 2: System.out.println("Enter amount:");

amt2 = s.nextInt();

sa.withdraw(amt2);

break;

case 3: sa.interest();

break;

case 4: sa.display();

break;

case 5: System.exit(0);

default: System.out.println("\ninvalid
input");

y

y

else

x

System.out.println("\nMenu\n 1. deposit
 2. withdraw 3. display");

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

ch = s.nextInt();

switch(h)

{

Case 1: System.out.println ("Enter amount");
amt1 = s.nextInt();
ca.deposit (amt1);
break;

Case 2: System.out.println ("Enter amount");
amt2 = s.nextInt();
ca.withdraw(amt2);
~~break~~ ca.check min();
break;

Case 3: ca.display();
break;

Case 4: System.exit(0);

default: System.out.println ("\n Invalid
Input");

break;

}

}

1

1

1

Output:

Enter the name:

John

Enter the type:

current

Enter the accno :

1

Enter the initial balance:

1000

Menu

1. deposit
2. withdraw
3. display

2 withdraw

3 dis

Enter choice :

2

Enter amount :

600

Menu

1. deposit
2. withdraw
3. display

Enter my choice

3

name : john , accno = 1 type : current
balance 4000

Menu

1. deposit
2. withdraw
3. display

Enter the choice :

4

```

package n1E
import java.util.Scanner
public class Student {
    protected String usn = new String();
    protected String name = new String(),
    protected int sem;
}

```

```

public void inputStudentDetails() {

```

```

    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter the usn!");
    usn = sc.next();
    System.out.println ("Enter name");
    name = sc.next();
    System.out.println ("Enter sem");
    sem = sc.nextInt();
}

```

```

public void displayStudentDetails() {

```

```

    System.out.println ("This usn is :" + usn + "\n The
    name is :" + name + "\n The sem : " + sem);
}

```

Inheritance.java

```

public class Internals extends Student {

```

```

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

```

```

public void inputInternalsMarks() {
}

```

```

Scanner s = new Scanner (System.in);
int i;
for (i=0; i<5; i++) {

```

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

marks[i] = s.nextInt();

y

3.

External.java

```
package SEE;
import CIF.Intunals;
import java.util.Scanner;
```

```
public class External extends Intunals {
    protected int marks[];
    protected int finalMarks[];
```

```
public External () {
```

marks = new int[5];

finalMarks = new int[5];

y

// Method to see SEE Marks

```
public void input_SEEMarks () {
```

Scanner sc = new Scanner(System.in);

System.out.println ("Enter SEE marks for
each course :");

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

System.out.print ("Course " + (i+1) + ":");

marks[i] = scanner.nextInt();

y

y

// to calculate final Marks:

```
public void calculateFinal Marks () {  
    for (int i=0; i<5; i++) {  
        final Marks[i] = marks[i]/2 +  
        super.internMarks[i];  
    }  
}
```

public void displayFinal Marks () {

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

j.

Main.java

```
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");~~~~System.out.println ("Enter SEE marks");~~

Date _____
Page _____

final Marks[i]. input SEEmarks(),

}

System.out.println("Displaying data:\n");
< for (int i=0; i<num of Students; i++)

final Marks[i]. calculateFinalMarks();
final Marks[i]. displayFinalMarks();

}

1.

~~Output:~~ 23.01.21

Enter USN: 124

Enter Name: Prakash

Enter Semester: 3

Enter CIE Marks:

Enter Internal marks for Prakash

Subject 1 marks : 33

Subject 2 marks : 36

Subject 3 marks : 28

Subject 4 marks : 31

Subject 5 marks : 40

Enter SEE Marks for ~~Prakash~~

Subject 1 marks : 89

Subject 1 : 77

Subject 2 marks : 91

Subject 2 : 81

Subject 3 marks : 78

Subject 3 : 67

Subject 4 marks : 84

Subject 4 : 73

Subject 5 marks : 90

Displaying data:

USN: 124

Name: Prakash

Semester: 3

String:

6)

- 1) BMSCE College
BMSCE Hostel
BMSCE Campus

7).

2)

- The string length is : 4
- The string literal is given by:
Welcome to the Jungle

8), 9)

After concatenation : Welcome to the
Jungle Bangalore.

10).

3. Demonstrate to toString()

Dimensions are 10.0 by 14.0 by 12.0

Box b : Dimensions are 10.0 by 14.0 by
12.0.

11)

- 4). The Extracted part of Welcome to
Bmsce college is :

5)

65

66

67

Welcome to bmsce College.

12)

6) Bmsce equals Bmsce → true
Bmsce equals Bmsce → false
Bmsce equals Bmsce → false
Bmsce equals IgnorCase → true.

7). substring

s1 = "Bmice College"

s2 = "Welcome to Bmice college of Engineering"

8),9) The given String starts with Bmice : True
The given string ends with ege : True.

10). Hello equals Hello == : true
Hello == Hello : false

~~Q&A~~
~~Q&A~~
16.01.24

11) The names in alphabetical order are:

apple
ball

cat

van

watch

12) Sorted Numbers (Ascending order): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

13). Output:

Thwas a was a test. Thwas was, too.

14). hello world.

15. Commeg

16) Hello Friends

17) Enter the number of students : 2

Enter the name : raj

Enter the regno : 1

Enter the cgpa : 9

Enter the name : ram

Enter the regno : 2

Enter the cgpa : 8.

Sorted by name:

name : raj	regno : 1	sem : 0	cgpa = 9.0
name : ram	regno : 2	sem : 0	cgpa : 8.0

18) Output:

Set Length : Hello

Char at Index 1 : e

After setCharAt : Hello

getChar : Hello

After append : Hello How are you?

After insert : Hello How awesome are you?

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

```
public class WrongAge extends Exception {
```

```
    public WrongAge () {
```

```
        super ("error message");
```

```
}
```

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

```
        super (message);
```

```
}
```

```
}
```

```
public class InputScanner {
```

~~Scanner sc;~~~~public InputScanner () {~~~~sc = new Scanner (System.in);~~~~static Scanner sc = new Scanner~~~~y~~~~(System.in)~~

```
public class Father extends InputScanner {
```

~~int fatherage;~~

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

~~super ();~~~~System.out.println ("Enter father's~~~~age");~~~~fatherage = sc.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();  
System.out.println ("Enter Son's age.");  
sonage = sc.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);  
}
```

```
public class MainException {  
    public static void main (String [] args) {  
        try {  
            Son s = new Son();  
            s.display();  
        } catch (WrongAge e) {  
            System.out.println ("Exception : " +
```

e.getMessage());

}

Output:

Enter Father's age = 47

Enter Son's age = 50

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

Enter Father's age = 47

Enter Son's age = -3

Exception: Age cannot be negative.

Enter Father's age = -57

Exception: Age cannot be negative.

~~30.01.2019~~

```
class BMSThread extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println ("BMS College  
of Engineering");  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```
class CSEThread extends Thread {  
    public void run() {  
        while (true) {  
            System.out.println ("CSE");  
            try {  
                Thread.sleep(2000);  
                // sleep for 2 sec  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

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

2
// to create & start first thread

BMSThread bmsThread = new BMSThread();

bmsThread.start();

// Create & start the second thread

CSEThread cseThread = new CSEThread();

,

↓

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

LAB Program 10.

Date / /

Page

ms Thread
();
read
d();

int n;
boolean valueSet = false;
synchronized int get()
{
 while (!valueSet)
 try
 {
 System.out.println("In Consumer waiting(n);");
 wait();
 }
 catch (InterruptedException e)
 }
 System.out.println("InterruptedException caught");
 System.out.println("Got: " + n);
 valueSet = true;
 System.out.println("In intimate Producer (n);");
 notify();
 return n;
}

Synchronized void put(int n)
{
 while (valueSet)
 try
 {
 System.out.println("In Producer Waiting(n);");
 wait();
 }
 catch (InterruptedException e)
 }
 System.out.println("InterruptedException caught");
}

white (valueSet)

try

System.out.println("In Producer Waiting(n);");
wait();

catch (InterruptedException e)

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

this.n = n;

valueSet = true;

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

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

notify();

}

}

class Producer implements Runnable

{

Q q;

Producer(Q q)

{

this.q = q;

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

public void run()

{

int i = 0;

while (i < 5)

{

 q.put(i++);

}

}

class Consumer implements Runnable

{

Q q;

Consumer(Q q)

{

this.q = q;

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

public void run()

{

```
int i=0;  
while(i<5)
```

```
    int r=q.get();
```

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

```
    i++;
```

```
}
```

```
}
```

```
}
```

```
class PCFixed
```

```
{
```

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

```
    { q = new Q();
```

```
        new Producer (q);
```

```
        new Consumer (q);
```

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

```
y
```

```
l
```

Output:

Put : 0

Intimate Consumer

Producer waiting

Get : 0

Intimate Producer

Put : 1

Intimate Consumer

Producer waiting

Consumed : 1

Got : 1

Intimate Producer

Consumed : 1

Put : 2

Intimate Consumer

Producer waiting

Got : 2

Intimate Producer

Consumed : 2

Put : 3

Intimate Consumer

Producer waiting

Got : 3

Intimate Producer

Consumed : 3

Put : 4

Intimate Consumers

Date _____
Page _____

Qnt : 4

Intimate Producer

consumed : 4

~~Intimate
Producer~~

Java GUI program :-

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

class UserInterface extends UserInterface {
    // create JFrame container
    JFrame jfrm = new JFrame ("Divide App");
    jfrm.setSize (275, 150);
    jfrm.setLayout (new FlowLayout ());
    // to terminate on close jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```

// text label

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

// add text field for both numbers

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

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

// calc button

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

// label

 ~~JLabel eor = new JLabel ();~~ ~~JLabel alab = new JLabel ();~~ ~~JLabel blab = new JLabel ();~~ ~~JLabel anslab = new JLabel ();~~

// add in order :)

```
jfrm.add (err); // to display error message.
```

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

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

Action-Listener calculateListener = new Action-
Listener() {

```
    public void  
    actionPerformed(ActionEvent evt) {  
        try {  
            int a = Integer.parseInt(aijf.  
            getText());  
            int b = Integer.parseInt(bijf.  
            getText());  
            if (b == 0) {  
                throw new ArithmeticException();  
            }  
            int ans = a / b;  
            alab.setText("In A = " + a);  
            blab.setText("In B = " + b);  
            anslab.setText("In Ans = " + ans);  
            err.setText(""); // Clear any previous error  
            message.  
        }  
    }
```

```
    catch (NumberFormatException e) {  
        displayErrorMessage("Enter Only Integers!");  
    }  
    catch (ArithmeticException e) {  
        displayErrorMessage("B should be non-  
        zero!");  
    }  
}.
```

20/2/24

Date _____
Page _____

```
private void displayErrorMessage (String message) {
    alab.setText ("");
    blab.setText ("");
    anslab.setText ("");
    err.setText (message);
}

button.addActionListener (calculateListener);
// display frame
jfrm.setVisible (true);

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

// create frame on event dispatching thread

```
SwingUtilities.invokeLater (new Runnable () {
    public void run () {
        new UserInterface ();
    }
});
```

Output:

Enter the divisor and dividend :

6

3

Calculate

A=6

B=3

Ans = 2

Some definitions:

- ① JFrame \Rightarrow It is an essential component of Java Swing. JFrame is a class that allows its create and manage a top-level window in a Java Application
- ② setSize \Rightarrow Sets the size of this Dimension object to the specified width & height
- ③ setLayout \Rightarrow allows you to set the layout of the container often a JPanel, to say flowLayout, BorderLayout, GridLayout or whatever layout desired
- ④ set default close operation \Rightarrow method provided by java swing
- ⑤ JLabel \Rightarrow is a built in Java Swing class that lets you display information on a JFrame
- ⑥ JTextField \Rightarrow a lightweight component that allows the editing of a single of text
- ⑦ add ActionListener \Rightarrow an interface in java.awt.event package
- ⑧ setText \Rightarrow substitutes the character SText for the text in the text field

- ⑦ add Frame: → used to add a new frame inside
the existing one

~~new frame~~
new frame

Lab 10. (Deadlock)

class A {

synchronized void foo(B b) {

String name = Thread.currentThread().get₂Name();

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

try {

Thread.sleep(1000);

Catch (Exception e) {

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

↳ "trying to call

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.B.last();

a.last(); }

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

```
Class Deadlock {  
    implements Runnable  
    A a = new A();  
    B b = new B();  
    Deadlock() {  
        Thread.currentThread().setName("Main Thread");  
        Thread t = new Thread(this, "Racing Thread");  
        t.start();  
        a.foo(b); // get a lock on a in this  
        thread  
        System.out.println("Back in main thread");  
    }  
    public void run() {  
        b.bar(a); // get a lock on b in other  
        thread.  
    }  
    public static void main(String args[]) {  
        new Deadlock();  
    }  
}
```

~~Main Thread entered~~

Outkut.

Main Thread entered A. foo

Racing Thread entered B. bar

Main Thread trying to call B.last()

Inside A.last

Back in main thread

Racing Thread trying to call A.last()

Inside A.last

Back in other thread

~~See
Ans 3
3.02.21~~

USN:1BM22CS340 SUJNYAN KINI

LAB-1

```
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("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 1 =" +r1+"-i"+r2);
}

}

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

```

USN:1BM22CS340 SUJNYAN KINI

LAB-2

```
import java.util.Scanner;

class Subject

{
    int subjectMarks;
    int credits;
    String grade;
}

class Student

{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];

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

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

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

void getMarks()
{
    int i;
    for(i=0;i<8;i++)
    {
        System.out.println("enter the marks and credits for course " + (i+1) + ":");

        System.out.println("marks : ");
        int marks = s.nextInt();

        System.out.println("credits : ");
        int credit = s.nextInt();

        subject[i].subjectMarks = marks;
        subject[i].credits = credit;

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

```

        }

        else if(marks>=50 && marks<60)

        {

            subject[i].grade = "B";

        }

        else if(marks>=40 && marks<50)

        {

            subject[i].grade = "C";

        }

        else if(marks>=0 && marks<40)

        {

            subject[i].grade = "F";

        }

    }

}

void computeSGPA()

{

    int i;

    double sgpa;

    double totalcredits = 0;

    double totalgradepoints = 0;

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

    {

        totalcredits += subject[i].credits;

        switch(subject[i].grade)

        {

            case "O" : totalgradepoints += 10*subject[i].credits;

            break;

            case "A+" : totalgradepoints += 9*subject[i].credits;

            break;

```

```

        case "A" : totalgradepoints += 8*subject[i].credits;
        break;
        case "B+" : totalgradepoints += 7*subject[i].credits;
        break;
        case "B" : totalgradepoints += 6*subject[i].credits;
        break;
        case "C" : totalgradepoints += 5*subject[i].credits;
        break;
        case "F" : totalgradepoints += 0*subject[i].credits;
        break;
    }
}

sgpa = totalgradepoints/totalcredits;
System.out.println("the sgpa is : "+sgpa);
}

}

class sgpa_Main
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}

```

USN:1BM22CS340 SUJNYAN KINI

LAB-3

```
import java.util.Scanner;

class Books

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

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

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

public class Books_Main
```

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

```
{  
    System.out.println(b[i]);  
}  
}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-4

```
import java.util.Scanner;

class inputScanner

{

    protected Scanner s;

    public inputScanner()

    {

        s = new Scanner(System.in);

    }

    public int getInput(String message)

    {

        System.out.println(message);

        return s.nextInt();

    }

}

abstract class Shape extends inputScanner

{

    protected int a,b;

    public Shape()

    {

        super();

    }

}
```

```
abstract public void printArea();

}

class Rectangle extends Shape

{

protected int a,b;

public Rectangle()

{

super();

}

public void printArea()

{

a=getInput("Enter the length:");

b=getInput("Enter the breadth:");

int area= a*b;

System.out.println("Area of the Rectangle:" +area);

}

}

class Triangle extends Shape

{

protected int a,b;

public Triangle()

{

super();

}

public void printArea()

{
```

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

}

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

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

}

public class Shape_Main
{
    public static void main(String[] args)
    {
        Rectangle r=new Rectangle();
```

```
Triangle t=new Triangle();
Circle c=new Circle();

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

}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-5

```
import java.util.Scanner;

class account

{
    String name;
    int accno;
    String type;
    double balance;

    account(String name,int accno,String type,double balance)
    {
        this.name=name;
        this.accno=accno;
        this.type=type;
        this.balance=balance;
    }

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

    void withdraw(double amount)
    {
        if((balance-amount)>=0)
        {
            balance-=amount;
        }
        else
        {
            System.out.println("insufficient balance,cant withdraw");
        }
    }
}
```

```
}

void display()
{
    System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance);
}

class savAcct extends account
{

    private static double rate=5;

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

    }

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

class curAcct extends account
{

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

```
curAcct(String name,int accno,double balance)
{
    super(name,accno,"current",balance);

}

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

class account_Main
{
    public static void main(String a[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("enter the name :");
        String name=s.next();
        System.out.println("enter the type(current/savings):");
        String type=s.next();
    }
}
```

```

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

System.out.println("enter the intial balance:");
double balance=s.nextDouble();

int ch;

double amount1,amount2;

account acc=new account(name,accno,type,balance);

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

curAcct ca=new curAcct(name,accno,balance);

while(true)

{

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

    {

        System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
interest 4.display");

        System.out.println("enter the choice:");

        ch=s.nextInt();

        switch(ch)

        {

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

                    amount1=s.nextInt();

                    sa.deposit(amount1);

                    break;

            case 2:System.out.println("enter the amount:");

                    amount2=s.nextInt();

                    sa.withdraw(amount2);

                    break;

            case 3:sa.interest();

                    break;

            case 4:sa.display();

                    break;
        }
    }
}

```

```

        case 5:System.exit(0);

        default:System.out.println("invalid input");

            break;

    }

}

else

{

    System.out.println("\nMenu\n1.deposit 2.withdraw 3.display");

    System.out.println("enter the choice:");

    ch=s.nextInt();

    switch(ch)

    {

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

                amount1=s.nextInt();

                ca.deposit(amount1);

                break;

        case 2:System.out.println("enter the amount:");

                amount2=s.nextInt();

                ca.withdraw(amount2);

                ca.checkmin();

                break;

        case 3:ca.display();

                break;

        case 4:System.exit(0);

        default:System.out.println("invalid input");

            break;

    }

}

}

```

USN:1BM22CS340 SUJNYAN KINI

LAB-6

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

```
package CIE;

import java.util.Scanner;

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

    public Internals() {}

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

```
package SEE;

import CIE.Internals;

import java.util.Scanner;

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

```
protected int finalMarks[];  
  
public Externals()  
{  
    marks = new int[5];  
    finalMarks = new int[5];  
}  
  
public void inputSEEmarks()  
{  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter SEE Marks for " + name);  
    for (int i = 0; i < 5; i++)  
    {  
        System.out.print("Subject " + (i + 1) + " marks: ");  
        marks[i] = s.nextInt();  
    }  
}  
  
public void calculateFinalMarks()  
{  
    for (int i = 0; i < 5; i++)  
        finalMarks[i] = marks[i] / 2 + super.marks[i];  
}  
  
public void displayFinalMarks()  
{  
    displayStudentDetails();  
    for (int i = 0; i < 5; i++)  
        System.out.println("Subject " + (i + 1) + ":" + finalMarks[i]);  
}
```

```
import SEE.Externals;

public class Main_264

{

    public static void main(String args[])

    {

        int numOfStudents=2;

        Externals finalMarks[]=new Externals[numOfStudents];



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

        {

            finalMarks[i]=new Externals();

            finalMarks[i].inputStudentDetails();

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

            finalMarks[i].inputCIEmarks();

            System.out.println("Enter SEE marks:");

            finalMarks[i].inputSEEmarks();

        }





        System.out.println("Displaying data:\n");

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

        {

            finalMarks[i].calculateFinalMarks();

            finalMarks[i].displayFinalMarks();

        }

    }

}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-7

```
import java.util.Scanner

class WrongAge extends Exception {

    public WrongAge(){
        super("Error message");
    }

    public WrongAge(String message) {
        super(message);
    }

    class inputScanner {
        static Scanner sc = new Scanner(System.in);
    }

    class Father extends inputScanner {
        int fatherage;

        Father() throws WrongAge {
            System.out.println("Enter Father's age");
            fatherAge = sc.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();  
  
        System.out.println("Enter son's age");  
        sonage = sc.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() {  
        System.out.println("Son's age is " + sonage);  
    }  
}
```

```
public class Mainexception {  
    public static void Main(String args[]) {  
        try{  
            Son son= new Son();  
            son.display();  
        }  
        catch(WrongAge e){  
            System.out.println("Exception "+ e.getMessage());  
        }  
    }  
}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-8

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

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

```
        }

        catch(InterruptedException e)

        {

            e.printStackTrace();

        }

    }

}
```

```
public class ThreadExample

{

    public static void main(String[] args)

    {

        BMSThread bmsThread=new BMSThread();

        bmsThread.start();

        CSEThread cseThread=new CSEThread();

        cseThread.start();

    }

}
```

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("MainThread");
        Thread t = new Thread(this,"RacingThread");
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
    }

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

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

}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-9

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

class UserInterface {
    UserInterface() {
        // create JFrame container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and dividend:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

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

```

// add in order

jfrm.add(err); // to display error message

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

ActionListener calculateListener = new ActionListener() {

    public void actionPerformed(ActionEvent evt) {

        try {

            int a = Integer.parseInt(ajtf.getText());

            int b = Integer.parseInt(bjtf.getText());

            if (b == 0) {

                throw new ArithmeticException();

            }

            int ans = a / b;

            alab.setText("\nA = " + a);

            blab.setText("\nB = " + b);

            anslab.setText("\nAns = " + ans);

            err.setText(""); // Clear any previous error message

        } catch (NumberFormatException e) {

            displayErrorMessage("Enter Only Integers!");

        } catch (ArithmeticException e) {

            displayErrorMessage("B should be non-zero!");

        }

    }

}

```

```
private void displayErrorMessage(String message) {  
    alab.setText("");  
    blab.setText("");  
    anslab.setText("");  
    err.setText(message);  
}  
};  
  
button.addActionListener(calculateListener);  
  
// display frame  
jfrm.setVisible(true);  
}  
  
public static void main(String args[]) {  
    // create frame on event dispatching thread  
    SwingUtilities.invokeLater(new Runnable() {  
        public void run() {  
            new UserInterface();  
        }  
    });  
}
```

USN:1BM22CS340 SUJNYAN KINI

LAB-10

```
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<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("Press Control-C to stop.");}}
```

