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:

RUSHIL MAGAZINE (1BM22CS225)

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

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

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	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-1

Quadratic

```
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();
    }
    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("Roots are = 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("Root2 = "+r1+" - i"+r2);
}
}
}
```

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

Output:

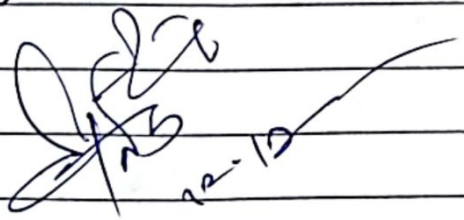
```
Rushil    1BM22CS225
Enter coefficients of a,b,c
4 5 6
Roots are imaginary
Root1 = 0.0+i1.0532687216470449
Root2 = 0.0-i1.0532687216470449

Rushil    1BM22CS225
Enter coefficients of a,b,c
1 -2 1
Roots are real and equal
Root1 = Root2 = 1.0

Rushil    1BM22CS225
Enter coefficients of a,b,c
1 -3 2
Roots are real and distinct
Root1 = 2.0 and Root2 = 1.0
```

- To Find area of rectangle

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

Output

```
10 8
length of rectangle = 10
breadth of rectangle = 8
Area of rectangle = 80
```



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

Q.) Find sum of 5 digit number

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

Output:

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

Q.) Program For palindrome

```
import java.util.Scanner;
class palindrome
{
    public static void main (String args[])
    {
        int n, t, rem, rev=0;
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter a 5 digit
        number:");
        n = sc.nextInt();
        t=n;
        while (t>0)
        {
            rem = t%10;
            rev = rev *10+rem;
            t= t/10;
        }
        if(rev==n)
        {
            System.out.println("palindrome");
        }
        else
        {
            system.out.println (" not palindrome");
        }
    }
}
```

Output

Enter a 5 digit number:
12321
palindrome

Enter a 5 digit number:
94821
not palindrome

(Q:) Program for prime number

```
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
          number");
      }
      else
      {
        for (i=2; i<=m; i++)
        {
        if(n%i==0)
```

```
{
    System.out.println(n+" is not a prime
        number");
    break;
}
if (flag==0)
{
    System.out.println(n+" is a prime number");
}
}
}
public static void main(String args[])
{
int i;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the value of n:");
i= sc.nextInt();
isprime(i);
}
}
```

Lab-2

7.

```
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");
    System.out.println("Enter usn of the student");
}
```

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

```
else if (marks >= 50 && marks < 60)
{
    subject[i].grade = "C";
}
else if(marks >=40
{
    subject[i].grade = 'F';
}
}
void computeSGPA()
{
  for (i=1; i<9; i++)
  {
    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;
     case 'B': totalgp += 6*subject[i].credits;
     break;
     case 'C': totalgp += 5*subject[i].credits;
     break;
     case 'F': totalgp += 4*subject[i].credits;
     break;
```

```
        }
    }
    SGPA = totalGradePoints / totalCredits;
    System.out.println
}
public class Sgpa
{
    public static void main(String[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA;
        s1.displayResult();
    }
}
```

Output

```
Enter your name;
Rushil
Enter your usn;
225;
Enter marks and credits For course 0;
```

Lab-3

```
import java.util.scanner;
class Books {
        String name;
        String author;
        int price;
        int num_pages;

    Books(string name, string author, int price, int
          num_pages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    public String toString (){

        String V="";
        V+ = "Book name :"+ this.name+"\n".
        V+ = "Author :" +this.author +"\n";
        V+ = "Price"+ this.price + "\n";
        V+ = "Number of Pages" + this.num_pages
        +"\n";
        return V;
    }
}
```

```
public class BookDemo{
    public static void main(String arg[])
    {
        int n;
        Book books[];
        String name,author;
        int price,num_pages;

        Scanner s=new Scanner(System.in);
        System.out.println("Enter number of
        books");
        n = s.nextInt();
        b= new Book[n]
        for(int i=0;i<n;i++)
        {
            system.out.println("Enter details
            of Book"+(i+1)+"\n";
            system.out.print("Name");
            name = s.next();
            system.out.print("Author");
            Author = s.next();
            System.out.print("Price");
            price = s.nextInt();
            system.out.print("Number of Pages");
            num_pages = s.nextInt();
            b[i] = new Books(name, author, price,
            num_pages);
    }
```

```
System.out.println("\n Displaying book details : \n");
for (int i=0; i<n; i++)
{
    System.out.println(b[i].toString());
}
}
```

Output:

```
Enter the number of Books: 2

Enter details of Book 1

Name : Rushil
Author : Dave
Price : 160
Number of pages: 320

Enter details of Book 2

Name : Adnan
Author : Chilkesh
Price : 180
Number of Pages: 420
```

Lab 4

```
import java.util.Scanner;

class InputScanner
{
    protected Scanner scanner;
    public InputScanner()
    {
        scanner = 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 length");
   b = getInput("Enter breadth");
   System.out.println("Area of rectangle" +(a*b));
  }
}
class Triangle extends Shape
{
   protected int a, b;
   public Triangle()
   {
      super();
   }
   public void printArea()
   {
     a = getInput("Enter side1");
     b = getInput("Enter side2"); Triangle"
     System.out.println("Area of side
      +(0.5*a*b));
   }
```

```
class circle extends Shape
{
    protected int a;
    public circle()
    {
        super();
    }
    public void printArea()
    {
        a = getInput("Enter the radius");
        System.out.println("Area of circle "+(3.14*a
    }
}
public class MainShape
{
    public static void main(String[] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

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

classmate
Date
Page

Output

Enter the length:
2
Enter the breadth:
4
Area of Rectangle: 8

Enter side1:
4
Enter side2:
8
Area of triangle: 16

Enter radius: 5
Area of circle: 78.5

2/11/24

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");
  }
```

```
void display()
{
System.out.println("name:"+name+"accno:"+accno+
"type:"+type+"balance"+balance);
}
class savAcct extends account
{
private savAcct static double rate=5;
savAct (string name, int, accno, double, balance)
{
super (name, accno, "savings", balance);
}

void interest()
{
balance += balance*(rate)/100;
System.out.println("balance:"+balance);
}
class curAct extends account
{
private double minBal=500;
private double serviceCharges=50;

curAct (string name, int accno, double balance)
{
super(name, accno, "current", balance);
}
```

```
void checkBal()
{
    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 BankAccountMain
{
    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 initial balance");
    double balance = s.nextDouble();
```

```
int ch;
double amount1, amount2;
Account acc = new Account (name, acno, type, balance);
SavAcct sa = new SavAcct (name, acno, balance);
CurAcct ca = new CurAcct (name, acno, balance);
while (true)
{
 if (acc.type.equals ("savings"))
{
  System.out.println ("In Menu \n1. deposit
  2. withdraw 3. compute 4. 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();
   break;
   case 2: System.out.println ("enter the amount");
   amount2 = s.nextInt();
   break;
   case 3: sa.interest ();
   break;
   default: System.out.println ("invalid input");
  }
```

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

16/01/2024

Lab-6

2) Output

str = "Hello World"
string length = 12

first name = "John"
lastname = "Cena"
concat = "John Cena"

4) Output : Bmsce

8) String 1 : Hello, World
Prefix : Hello
StartsWith() Result = True

String 2 : Java Programming
Prefix : Python
StartsWith() Result : False

9) Output →

String 1 : Hello World
Postfix : World
endswith() Result : True

String 2 : Java Programming
Postfix : Python
endswith() Result : False

6) Output:

Bmsce equals Bmsce → True
Bmsce equals College → False
Bmsce equals BMSCE → False
Bmsce equalsIgnoreCase BMSCE → True

10)

Output:

String str1 = new String ("Hello");
String str2 = new String ("Hello");

Using equals(): True
Using ==: False

String str3 = "Java";
String str4 = "Java";

Using equals(): True
Using ==: True

11) Output :

apple
ball
cat
dog
ent
Free
gun
her
ice
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree
umbrella
Van
watch
xmas
yatch
zee

12) Output:

Sorted Numbers (Descending Order):

10
9
8
7
6
5
4
3
2
1

13) Output:

Original string: Thwas was a test. Thwas was, too

Modified string: Thus us a test. Thus us, too

14) Output

String s1 = "Hello"

String s2 = "World"

concatenated string = HelloWorld

1) Hello, World!
Hello
Hello
Java.

3) Dimensions are 10.0 by 14.0 by 12.0!

5) Byte-Array.

Bytes - 72, 101 108 111 44 32 87
117, 14 108 100 33.

Char Array:

7) Substring we matched

15) Orginal string: Welcome to College
Modified String: Welcome to Commage

16) Orginal String : 'Hello friend'
Trimmed string : 'Hello friends'

17) Enter details for student 1:
Registration Number: 476
Full Name: Rushil
Semester: 3
CGPA: 8.9
Enter details for student 2:
Registration Number: 44
Full Name: Adnan
Semester: 3
CGPA: 9.2
Enter details for student 3:
Registration number: 1702
Full Name: Dsuv
Semester: 3
CGPA: 7.2

19) After setLength(5): Hello
charAt(1): After setChar(1, 'a'):
Hallo getChars(0, 5, charArray, 0): Hello
After operand: Hello World: After
insert(6, "Java"): Hello, Java World!

reverse: !dlroW avaJ ,olleH
delete(5, 11): !dlro, olleH
After deleteCharAt(1): !lro, olleH
After replace(7, 12, "world"):
Hello, world Substring(7, 12): world

20) Circle:
Area: 78.5421
Perimeter: 31.4515

Triangle
Area: 6.0
Perimeter: 12.0

Lab -7

Student.Java

```
package ai;
public class students {
        public string name, USN;
        public int sem;
};
```

Internal.java

```
package ai;
import.java.util scanner;
public class internal extends student{
        public int marks[] = new int [5];
        public void input marks()
        {
        scanner sc = new Scanner (system.in);
        For (int i=0;i<5;i++){
                System.out.println("Enter
                subject"+(i+1)+"marks");
                marks[i] = sc.nextint();
        }
}
public void display marks(){
        for (int i=0; i<5; i++){
                System.out.println("Subject"+(i+1)+
                "marks" + marks[i]);
}
```

```
}
```

D)

Externals.java

```
package see;
import ae.Student;
import java.util.Scanner;

public class Externals extends Students {
    public int marks[] = new int[5];
    public void input marks(){
        scanner sc = new Scanner(System.in);
        for(int i=0; i<5; i++){
            System.out.println("enter subject"
            +(i+1)+"marks");
            marks[i] = sc.nextInt();
        }
    }
    public void display marks(){
        for(int i=0; i<5; i++){
            System.out.println("subject"+(i+1)+"marks"+marks[i]);
        }
    }
}
```

Main.java

```
import ue.Student;
import ie.Internals;
import ue.Externals;
import java.util.Scanner;

class Main{
    public static void main (String args[]){
        int no=2;
        Externals final marks[] = new
        Externals[100];
        Internals int marks[]= new
        Internals(100);
        for(int i=0;i<100;i++){
        final marks[i] = new Externals()
        int marks[i] = new Internals();
        final marks[i] = input marks[];
        int marks[i]. input marks[];
        }
        for(int i=0;i<100;i++){
            System.out.println("cie ");
            int marks[i] . display marks();
            System.out.println("SEE:");
            Final marks[i]. display marks();
        }
    }
```

Output :

Enter Subject 1 Marks : 30
Enter Subject 2 Marks : 50
Enter Subject 3 Marks : 40
Enter Subject 4 Marks : 20
Enter Subject 5 Marks : 10

Enter Subject 1 Marks : 30
Enter Subject 2 Marks : 70
Enter Subject 3 Marks : 60
Enter Subject 4 Marks : 80
Enter Subject 5 Marks : 90

CIE

Subject 1 Marks : 30
Subject 2 Marks : 50
Subject 3 Marks : 40
Subject 4 Marks : 20
Subject 5 Marks : 10

SEE

Subject 1 Marks : 30
Subject 2 Marks : 70
Subject 3 Marks : 60
Subject 4 Marks : 80
Subject 5 Marks : 90

06 | 02 | 24

Lab-1 Week-8

```
import java.util.Scanner;

class WrongAge extends RuntimeException {
    public WrongAge() {
        super("Age cannot be negative");
    }
    public WrongAge(String message) {
        super("message");
    }
}

class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }
    public int nextInt() {
        return scanner.nextInt();
    }
}
class Father extends InputScanner {
    protected int FatherAge;

    public Father() {
        System.out.println("Enter Fathers age");
        FatherAge = super.nextInt();
```

```
        if (fatherAge<0){
            throw new WrongAge ("Age cannot be negative");
        }
    }
    public void display(){
        System.out.println("Fathers Age "+fatherAge);
    }
}

class Son extends Father{
    private int sonAge;

    public Son(){
        super();
        System.out.println("Enter son's age");
        sonAge = super.nextInt();

        if (sonAge > fatherAge){
            throw new WrongAge ("Sons age cannot be
            greater than fathers age");
        }
        else if(sonAge<0){
            throw new WrongAge ("Age cannot be negative");
        }
    }
```

```
    public void display(){
        super.display();
        System.out.println("Sons age "+ sonAge);
    }
}

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

Output

Enter fathers age : 20

Enter sons age : 40

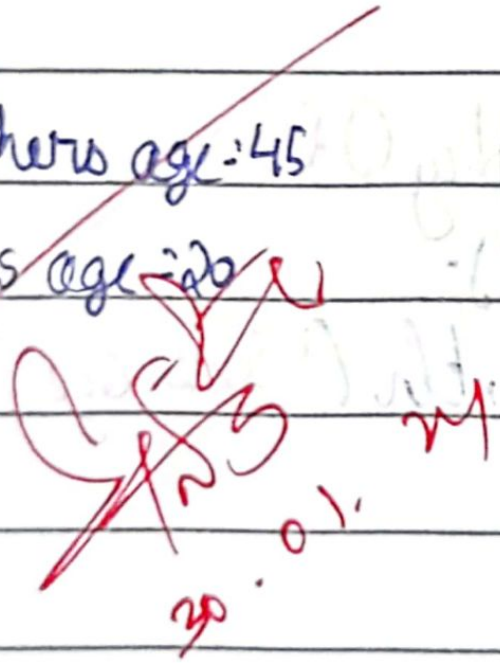
Sons age cannot be greater than father's age

Enter fathers age : -10

Age cannot be negative

Enter Fathers age: 45

Enter sons age: 20



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.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();
        }
}
```

Output →

BMS- College of Engineering
CSE 1
CSE 2
CSE 3
CSE 4
CSE 5

BMS -College of Engineering
CSE 1
CSE 2
CSE 3
CSE 4
CSE-5

Lab-10

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

Output

Press Control C to Stop

Put :0
Intimate Consumer
Producer Waiting

Pal
Got:0

Intimate Producer
Put :1

Intimate Consumer

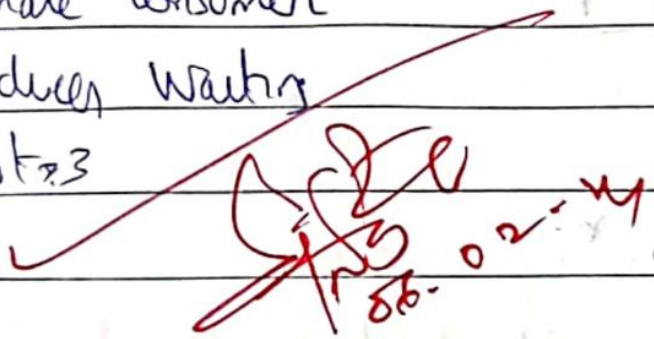
Producer Waiting

Consumed :0
Got :1

Intimate Producer
Consumed :1
~~Got Got :1~~
Put :2

Intimate Consumer
Producer Waiting
Got :2

Intimate Consumer
Produces Waiting
Cot=3



Lab-9 (Week-10)

```
import java.io.*;

class B extends Thread {
    public void run() {
        try {
            for (int i=0; i<3; i++) {
                System.out.println("BMS");
                Thread.sleep(10000);
            }
        catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}

class C extends Thread {
    public void run() {
        try {
            for (int i=0; i<3; i++) {
                System.out.println("CSE");
                Thread.sleep(20000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

```
class Threadmain {
    public static void main(String args[])
    {
        Bb = new B();
        Cc = new C();
        b.start();
        c.start();
    }
}
```

Deadlock.java

```
class A {
    Synchronized void foo (B b){
        String name = Thread.currentThread().getName();
        System.out.println(name + "entered A");
        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() {
    S.O.P ("Inside A.last");
    }
class B{
    Synchronized void bar (A a)
    String name = Thread.currentThread().getNam
    System.out.println (name+"extend B.bar");
    try{
        Thred.sleep (1000);
    } catch (Exception e)
        System.out.println ("B interrupted");
    }
    System.out.println (name + " trying to call A.l
        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
        t.start();
        a.foo(b);
```

```
    S.O.P ("Back in Thread");
}
    new Deadlock;
}
```

Output:

```
Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main Thread
Racing Thread trying to call A.last()
Inside A.last
Back in other Thread
```

Pro.con Java

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while(!valueSet)
        try {
            System.out.println("Consumed waiting");
            wait();
        } catch (InterruptedException e) {
            S.O.ph(e);
        }
        S.O.Ph(" " + n);
        valueSet = false;
        S.O.ph("Test produces");
        notify();
    }
    return n;
}
}

Class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "producer").start();
    }
}
```

```
public void (1) {
        int i=0;
        while (i<3) {
            q.put (i++);
        }
    }
}
class consumer implements Runnable {
        Q q;
        Consumer (Q q) {
            this.q = q;
        new Thread (this, "Consumer").start();
    }
    public void in {
        int i=0;
        while (i<3) {
            int n = g.get();
            S.O.Ph ("Consumed:"+n);
            i++;
        }
    }
}
```

Output

Put:1
got:1
Put:2
got:2
Put:3
got:3

classmate
Date
Page

Lab-9

```
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 dividant");

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

        JButton = 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 ae){
      S.O.println("Action event from a text field");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);

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

        alab.setText("A = "+a);
        blab.setText("B = "+b);
        anslab.setText("Ans = "+ans);
```

```
catch (NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Enter only Integers");
}
catch (ArithmeticException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be non zero");
}
}
jfrm.setVisible(true);
}
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

OUTPUT:

Enter the dividend and divisor:

25 5

Calculate A=25 B=5 Ans=5

Date
Page

JFrame: is a class in Java Swing that represents a top level container for a GUI application.

setSize():

The setSize(int width, int height) method is used to set the size of a JFrame in pixels.

setLayout():

The setLayout(LayoutManager manager) method is used to set the layout manager for the frame

setDefaultCloseOperation():

setDefaultCloseOperation(int operation) is used to the set the default close operation for the JFrame.

JLabel:

'JLabel' is a Swing component used to display a non-editable text or image.

JTextfield:

'JTextfield' is a Swing component that allows the user to enter and edit a single line of text.

add(Frame):

'add(Component comp)' as a method used to add components to a container, such as a JFrame.

addActionListener():

'addActionListener(ActionListener listener)' as a method used with components like buttons to register an ActionListener.

setText():

'setText(String text)' as a method used with components like JTextField or JLabel to set the text content.

LAB-1(QUADRATIC)

```
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);
                        r1=((-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();
        }
}
```

Lab-2

```
import java.util.Scanner;
class Subject
{
	int marks;
	int credits;
	int grade;
}
class Student
{
	Subject sub[];
	String name;
	String usn;
	double SGPA;
	double n_sum=0;
	double d_sum=0;
	Scanner s;

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

```
}

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

void getMarks()
{
        int i;
        int numerator;
        for(i=0;i<8;i++)
        {
                System.out.println("Enter marks:");
                sub[i].marks = s.nextInt();
                System.out.println("Enter credits:");
                sub[i].credits = s.nextInt();
                sub[i].grade=sub[i].marks//10+1;
                if(sub[i].grade<4 || sub[i].grade>10)
                        sub[i].grade=0;
                numerator= sub[i].credits * sub[i].grade;
                n_sum = n_sum + numerator;
                d_sum = d_sum + sub[i].credits;
        }
}
```

```
	void computeSGPA()
	{
		SGPA= n_sum / d_sum;
		System.out.println("SGPA= "+SGPA);
	}
}

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

/*OUTPUT:
Enter student name:
Rushil
Enter student usn:
1BM22CS225
Enter marks:
94
Enter credits:
4
Enter marks:
92
Enter credits:
```

4

Enter marks:

93

Enter credits:

3

Enter marks:

88

Enter credits:

3

Enter marks:

92

Enter credits:

3

Enter marks:

91

Enter credits:

1

Enter marks:

92

Enter credits:

1

Enter marks:

93

Enter credits:

1

SGPA= 9.85 */

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 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 Books[n];
      for (i = 0; i < n; i++)
      {
              System.out.println("Enter the details of book" + (i+1) + ":");

         System.out.println("Enter the name of the book:");
              name = s.next();
              System.out.println("Enter the author name:");
         author = s.next();
              System.out.println("Enter the price:");
         price = s.nextInt();
              System.out.println("Enter the number of pages:");
         numPages = s.nextInt();

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

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

LAB-4

```
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 bredth:");
```

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

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

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

}

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

```
	}

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

	}

}

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

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

```
import java.util.Scanner;
class account
{
	String name;
	int accno;
	String type;
	double balance;

	account(String name,int accno,String type,double balance)
	{
		this.name=name;
		this.accno=accno;
		this.type=type;
		this.balance=balance;
	}
	void deposit(double amount)
	{
		balance+=amount;
	}
	void withdraw(double amount)
	{
		if((balance-amount)>=0)
		{
			balance-=amount;
		}
		else
		{
			System.out.println("insufficient balance,cant withdraw");
		}
```

```
        }

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

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

        }

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


}
class curAcct extends account
{

        private double minBal=500;
```

```
	private double serviceCharges=50;

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

	}

	void checkmin()
	{

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

	}

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

```
String name=s.next();
System.out.println("enter the type(current/savings):");
String type=s.next();
System.out.println("enter the account number:");
int accno=s.nextInt();
System.out.println("enter the intial balance:");
double balance=s.nextDouble();
int ch;
double amount1,amount2;
account acc=new account(name,accno,type,balance);
savAcct sa=new savAcct(name,accno,balance);
curAcct ca=new curAcct(name,accno,balance);
while(true)
{
	if(acc.type.equals("savings"))
	{
		System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute interest 4.display");
		System.out.println("enter the choice:");
		ch=s.nextInt();
		switch(ch)
		{
			case 1:System.out.println("enter the amount:");
				amount1=s.nextInt();
				sa.deposit(amount1);
				break;
			case 2:System.out.println("enter the amount:");
				amount2=s.nextInt();
				sa.withdraw(amount2);
```

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

                case 3:ca.display();
                    break;
```

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

LAB-7

```
package CIE;

public class Internals {
   private int[] internalMarks = new int[5];


   public Internals() {

   }


   public void setInternalMarks(int[] internalMarks) {
      this.internalMarks = internalMarks;
   }
```

```
    public int[] getInternalMarks() {
        return internalMarks;
    }
}

package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;


    public Student() {
        this("", "", 0);
    }


    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }


    public void setUsn(String usn) {
        this.usn = usn;
    }
```

```
    public void setName(String name) {
        this.name = name;
    }

    public void setSem(int sem) {
        this.sem = sem;
    }


    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {
        return sem;
    }
}

package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new int[5];
public External() {
```

```
        this("", "", 0, new int[5]);
    }
  public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
public void setSeeMarks(int[] seeMarks) {
        this.seeMarks = seeMarks;
    }
 public int[] getSeeMarks() {
        return seeMarks;
    }
}
import CIE.Student;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Allow the user to enter the number of students
        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];
```

```
        // Initialize students, internals, and externals
        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            System.out.print("Enter USN for student " + (i + 1) + ": ");
            students[i].setUsn(scanner.next());

            System.out.print("Enter name for student " + (i + 1) + ": ");
            students[i].setName(scanner.next());

            System.out.print("Enter semester for student " + (i + 1) + ": ");
            students[i].setSem(scanner.nextInt());

            internals[i] = new Internals();
            // Assuming a simple method to input internal marks with validation
            internals[i].setInternalMarks(inputMarksWithValidation("internal", i, scanner, 0, 50));

            externals[i] = new External(students[i].getUsn(), students[i].getName(), students[i].getSem(), new
int[5]);
            // Assuming a simple method to input external marks with validation
            externals[i].setSeeMarks(inputMarksWithValidation("external", i, scanner, 0, 100));

            // Calculate final marks for the ith student and display
            int[] finalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = internals[i].getInternalMarks()[j] + externals[i].getSeeMarks()[j] / 2;
            }

            System.out.println("Student " + (i + 1) + " Final Marks: " +
```

```
            finalMarks[0] + ", " + finalMarks[1] + ", " + finalMarks[2] + ", " +
            finalMarks[3] + ", " + finalMarks[4]);
        }

        scanner.close();
    }

    private static int[] inputMarksWithValidation(String type, int studentIndex, Scanner scanner, int min,
int max) {
        int[] marks = new int[5];
        System.out.println("Enter " + type + " marks for student " + (studentIndex + 1) + ": ");
        for (int i = 0; i < 5; i++) {
            int mark;
            do {
                System.out.print("Subject " + (i + 1) + ": ");
                mark = scanner.nextInt();
                if (mark < 0 || mark > max) {
                    System.out.println("Invalid input. " + type + " marks should be between 0 and " + max + ".
Please try again.");
                }
            } while (mark < 0 || mark > max);

            marks[i] = mark;
        }
        return marks;
    }
}
```

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();
    }
}
```

LAB-9(DEADLOCK)

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

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

	}

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

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

		try
		{
			Thread.sleep(1000);
		}
		catch(Exception e)
		{
			System.out.println("B Interrupted");
		}

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

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

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

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

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

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

}
```

```
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();
            }
        });
    }
}
```

LAB-10(IPC)

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

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


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

```
                while(i<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.");
	}
}
```