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:

REVANTH K (1BM22CS220)

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

Revanth K

IBM22CS220

①
```
Class helloWorld
{
    public static void main (String [] args)
    {
        System.out.println ("helloWorld");
    }
}
```

O/P :- helloWorld

②
```
class rectangle Area {
    public static void main (String [] args){
        int length, breadth;
        length = Integer.parseInt (args[0]);
        breadth = Integer.parseInt (args[1]);
        int area = length * breadth;
        System.out.println { "length =" + length);
        System.out.println (" breadth =" + breadth);
        System.out.println ("Area=" + Area);
    }
}
```

O/P :-

length = 5

breadth = 5

Area = 25

③ Array :-

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

O/P →   April has 30 days.

④

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

op:

enter a string:

revanth

you entered string revanth

enter an integer

3.

you entered integer 3

- enter a float

3.5

you entered float 2.5

(5)

```java
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");

} 3

}

3

Output:-

Enter a 5 digit number:

12321

Palindrome

Enter a 5 digit number

123u5

not palindrome

(6) Quadratic :-

O/p :-

Output 1:

Enter the coefficients of a, b, c

1

5

2

roots are real and distinct

root1 = 4.5615528128   root 2 = 4.5615528128

Output 2 :- enter the coifficients of a, b, c

$1 \quad 2 \quad 1$

roots are real and equal

root 1 = root 2 = -1.0

Output 3 :-

enter the coefficients of a, b, c

0   4   5

not a quadratic equation

enter a non zero value for a;

1

roots are imaginary

root 1 = -2.0 + i NaN

root 2 = -2.0 - i NaN

⑦

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

}
}

Output :-

Enter a number!

5

The factorial is :

120

⑧

```java
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);
    }
}
```
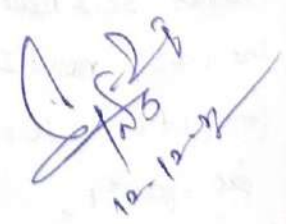
Output :-

Enter a 5-digit number:

12 3 4 5
Sum of digits = 15

Complete all

a)
```
import jara.util.*;

class is prime
{
    Static void is prime (int n)
    {
        int i, m=0, flog=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++)
        {
            System.out.println(n+"is a prime number")
            if (n/i ==0)
            {
                System.out.println (n+" is not a prime number");
                flog= 1;
                break;
            }
            if (flog ==0)
            {
                System.out.println (n+ "is a prime number");
            }
        }
    }
}
```

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

    int i;

    Scanner s = new Scanner (System.in);
    System.out.println ("Enter the value of i: ");
    i = s. nextInt ();
    isprime (i);

}
}
```

Output:-

Enter the value of i

7

It is a prime number

SGPA Lab-2 :-

```java
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();
    }
```

```java
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 = S.nextInt();
    System.out.println ("credits:");
    int credit = S.nextInt();
    Subject [i]. SubjectMarks= marks;
    Subject [i]. credits = credit;

    if (marks >a=0 90 && mark <=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)
    {
```

```cpp
            subject [i].grade ="B";
    }
    else if (marks >=40 && marks <50)
    {
        subject [i].grad = "c";
    }
    else if (marks >= 0 && marks <40)
    {
        subject.[i].grade = "F";
    }
    }
}

void compute SGPA()
{
    int i;
    double sgpa;
    double total credits =0;
    double totalgrade points = 0;

    for (i=0 ; i< 8; i++)
    {
        total credits + = subject [i].credits;
        switch (subject [i].grade)
        {
            case "O": totalgrade points + = 10 * subject [i].credits;
                    break;

            case "A+": total grade points+ = 9 * subject [i].credits;
                    break;

            case "A": total grade points + = 8 * subject [i].credits;
                    break;

            case "B+": total grade points + = 7 * subject [i].credits;
                    break;
```

```
Case "B": totalgrade pouhts = 6 * subject [i]. Credits;
    break;

Case "C": totalgrade pouts + = 5 * subject [i]. credits;
    break;

Case "F": total grade potts + = 0 * subject [i]. credits;
    break;

    }

}

    sgpa = total.grade pouts / total credits;
    System.out.poutln (" the sgpa is : " + sgpa);
    }
}

class sgpa
{
    public static void main (String args[])
    {
        Student S1 = new Student;
        S1.get Student Details ();
        S1.get Marks();
        S1.compute SGPA();
    }
}
```

Output :-

Enter your name:
Revanth k

Enter your USN

1BM22 CS220

Enter the marks and credit for course 0:

marks:

95

Credit:

4

Enter the marks and credits for course 1:

marks:

92

credit

4

Enter the marks and credit for course 2:

Marks:

80

credit:

$1

Enter the marks and Credit for Course 3:

marks:

95 ✓

credits:

$1

Enter the marks for Course 4:

maks:

91

Credits

3

Enter the marks and course 5:

Marks: 91

credits = 4

Entr the marks and credith for cours 7:

Mark : 90

credit :

1

the sgpa is : 10.000

19.12

26|12|23

Lab-3

```java
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 to String ()
    {
```

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

public class Mainbook
{
    public static void main (String args [])
    {
        Scanner s = new Scanner (system.In);
        int n;
        int i;
        String name;
        String author;
        int price;
        int numPages;
System.out.println ("Enter the no of books: ");
        n = s.nextInt();

        Books b[];

        b = new Books [n];
for (i = 0; i<n; i++)
{
        system.out.println (" Details of book" + (i+1) + ":");
        System.out.println ("enter name of book:");
        name = s.next();
        System.out.println ("author name:");
        author = s.next();
```

```java
System.out.println ("price :");
price = s.nextInt();
System.out.println ("no of pages :");
numPages = s.nextInt();

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

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

Output :-

Enter the no of books :

2

Enter the details of book 1 :
Enter the name of book:
harry potter
Enter the author name :
jkrowlings
Enter the price:
1000
Number of pages:
800

Enter the details of book2:
Enter the name of book:
Revolutionary 2020

author name:
Chethan Bagath

price:
2000

number of pages:
1500

Book details:

Book name: harry potter
Author name: jk rowlings

price: 1000
number of pages : 800

Book name: Revolutionary 2020
Author name : Chethan Bagath

price: 2000
number of pages = 1500

26.12.22

Lab-4

```java
import java.util.scanner;

class input Scanner
{
    protected Scanner scanner;
    public input Scanner()
    {
        scanner = new Scanner (System.in);
    }
}

abstract class shape extends input scanner
{
    double a,b;
    public shape()
    {

        super();
        System.out.println ("the area of a:");

        a = scanner.next Double ();
        System.out.println (" the area of b:");
        b= scanner.next Double ());
    }
}

class rectangle extends shape
{
    public rectangle()

    void area ()
    {
        double area = a * b;
        System.out.println (" the area of rectangle is:" + area);
    }
}
```

```java
class triangle extends shape
{
    ~~public triangle()~~

    void area()
    {
        double area = 0.5 * a * b;
        System.out.println("the area of triangle is: "+area);
    }
}

class circle extends shape
{
    void area()
    {
        double area = 3.14 * a * a;
        System.out.println("the area of rectangle is: "+area);
    }
}

public class mainArea
{
    public static void main(String[] args)
    {
        rectangle r = new rectangle();
        triangle t = new triangle();
        circle c = new circle();

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

output:

the area of a:

2

the area of b:

3

the area of a:

4

the area of b:

5

the area of a:

6

the area of b:

7

the area of rectangle is: 6. 0
the area of triangle is: 10.0
the area of circle is: 113.039

2/1/24

Lab - 3 :-   9/01/24

```java
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:" +
            "balance:" + balance);
    }
}
```

```java
}

class SavAcct extends acord
{
        private static double rate =5;
        Sav Acct (String name, int acno, double balance)
        {
            super (name, acno, "savings", balance);
        }

        void intrest ()
        {

            balance += balance* (rate)/10.0;
            System.out.println ("balance" + balance);
        }
}

class currAcct extends account
{
        private double minbal =500;
        private double service charges =50;

        curr Acct (String name, int acno, double balance)
        {
            super (name, acno, "curid", balance);
        }
        void checkmbal()
        {
            if (balance < min Bal)
            {
                System.out.println ("balance is less than min balance, service
                                charges imposed:" + service charges);
```

```java
        balance =- service charges;
        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 (cent/saving):");

        String type = s.next ()
        System.out.println ("enter the accout number:");

        int acno = s.nextInt();
        System.out.println ("enter the initial balance:");
        double balance = s.nextDouble();
        int ch;
        accout acc = new acc (name, acno, type, balance);
        Sav Acct sa = new SavAcct (name, acno, balance);
        Cur Acct ca = new CurAcct (name, acno, balance))
        while (true)
        {
            if (acc.type.equals ("savings"))
            {
                System.out.println ("\n Menu \n 1. deposit 2. withdraw
                                    3. compute interest 4. display");
                System.out.println (" enter the choice:");
                ch = s.nextInt ();
                switch (ch)
```

```java
{
    Case 1: System.out.println ("enter the amt: ");
            amount += S.nextInt();
            sa.deposit (amt);
            break;

    Case 2: System.out.println ("enter the amt");
            amount 2 = S.nextInt();
            sa.withdraw (amt 2);
            break;

    Case 3: sa.intrest ();
            break;

    Case 4: sa.display ();
            break;

    Case 5: System.exit (0)

    default : System.out.println ("invalid input");
            break;
    }
}
else
{
    System.out.println ("\n Menu \n 1. deposit 2. withdraw 3. display");

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

    ch = S.nextInt();

    switch (ch)
    {
```

```java
Case 1:  System.out.print ("enter the amt:");
         amount 1 = s.nextInt();
         ca.deposit(amount 1);
         break;

Case 2:  System.out.println ("enter the amt");
         amount 2 = s.nextInt();
         ca.withdraw(amt 2);
         ca.checkmin ();
         break;


Case 3:  ca.display();
         break;

Case 4:  System.exit(0);

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

→ Output:-

Enter the name:
        revanth

enter the account number:
        2201

enter the initial balance:
        5000

menu
1. deposit   2. withdraw   3. display

entu choice:

2

entr amot:

600

Menu

1.depost. 2.withdraw 3.display

ente the choice

3

name: Pavant au no: 2201 type: curnt balan 5600

09.01.24

Lab-6 :-

16/01/24

Question 1 :-

type 1: BMSCE

type 2: BMSCE

type 3: BMSCE

type 4: MS

type 5: abcd

Question 2 :-

length of S1 = 5

concatenation of S1 and S2 : BMSCEBMSCE

question 3 :-

to String ():10

Question 5 :-

65 66 67 68 69 70

B   M   S   C   E

Question 4 :-

The given string is: Welcome to bmse college

The SrcBegin, SrcEnd, and dstBegin or values are : 11, 16 and 0

The value of character array: [b, m, s, c, e, ...]

Question 6 :-

BMSCE equals BMSCE → true

BMSCE equals College → false

BMSCE equals ignorecase BMSCG → true

Question 7 :-

Substring $ matched

$S1 = $ "BMSCE college";

$S2 = $ "welcome to BMSCG college of engineering";

Question 8 :-

true

false

Question 9 :-

False

True

Question 10 :-

Hello equals Hello → True

Hello == Hello → False

Question 11 :-

The names in alphabetical order are:

apple

ball

cat

venn

coatch

Question 12 articles

Sorted numbers : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Question 13 :-

Thuas was a test thuas was, too

Question 14 :-

hello as orld

Question 15 :-

connege

Lab -7

```java
//Student.java
Package.CIE;

import java.util.Scanner;

public class Student {

    protected String usn = new String();
    protected String name = new string();
    protected int sem;

    public void inputStudentDetails() {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        USN = scanner.next();
        System.out.print("Enter Name!");
        name = scanner.next();
        System.out.print("Enter Semester:");
        sem = scanner.nextInt();

    }

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

    }
}
```

```java
// Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];

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

```java
// Externals.java
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];

    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }
```

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

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

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

```java
//Main.java
import SEE.Externals;

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

        int num of Students = 2;
        . Externals final Marks[] = new Externals [num of students];

        for (int i=0; i < num of students; i++) {
            final Marks[i] = new Externals();
            final Marks[i].input Student Details();
            System.out.println("Enter CIE marks");
            final Marks[i].input CIE marks();
            System.out.println("Enter SEE marks");
            final Marks[i].input SEE marks();
        }

        System.out.println("Displayy data :\n");
        for(int i = 0; i < num of students; i++) {
            final Marks[i].Calculate Final Marks();
            final Marks[i].display final Marks();
        }
    }
}
```

Output :-

Enter USN: 1BM22CS201

Enter Name : A

Enter Semester : 2

Enter CIE marks.
Enter Internal mark for A

Sub 1 : 45

Sub 2 : 46

Sub 3 : 42

Sub 4 : 43

Sub 5 : 44

Enter for SEE marks

Enter SEE marks for A

Sub 1 : 85

Sub 2 : 89

Sub 3 : 87

Sub 4 : 88

Sub 5 : 87

Enter USN: 1BM22CS220

Enter Name = B

Enter Semester : 2

Enter CIE marks

Enter Internal Marks for B

Sub 1 = 45

Sub 2 = 47

Sub 3: 40

Sub 4 : 43

Sub 5 : 44

Extra SEE marks

Extra .SEE marks for B

Sub 1 : 90

Sub 2 : 78

Sub 3 : 79

Sub 4 : 89

Sub 5 : 80

10 Displaying data :

USN : IBM22CS201

Name : A f

Semester : 2

Sub 1 : 87

Sub 2 : 90

Sub 3 : 85

Sub 4 : 87

Sub. 5 : 87

USN : 1BM22CS220

Name : B

Semester : 2

Sub 1 : 90

Sub 2 : 86

Sub 3 : 76

Sub 4 : 88

Sub 5 : 75

28·01·24

```java
import java.util.Scanner;

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

class InputScanner
{
    protected Scanner S;
    public InputScanner ()
    {
        S = new Scanner (System.in);
    }
}

class Father extends InputScanner
{
    protected int fatherAge;
    public Father () throws WrongAge
    {
        System.out.println ("Enter Father's Age :");
        fatherAge = S.nextInt ();

        if (fatherAge < 0)
        {
            throw new WrongAge ("Age cannot be negative :"));
        }
    }
}
```

```java
public void display()
{
    System.out.println("Father's Age :" + fatherAge);
}
}


class Son extends Father
{
    private int sonAge;
    public Son() throws WrongAge
    {
        super();
        System.out.println("Enter Son's Age :");
        sonAge = s.nextInt();
        if (sonAge > fatherAge)
        {
            throw new WrongAge("Son's Age can't be greater than
                                father's age");
        }
        else if (sonAge < 0)
        {
            throw new WrongAge("Age can't be negative");
        }
    }
    public void display()
    {
        super.display();
        System.out.println("Son's Age :" + sonAge);
    }
}
```

```
public class FatherSonAge
{
    public static void main (String args[])
    {
        try
        {
            Son Son = new Son();
            Son.display();
        }
        catch (wrong Age)
        {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:-

Enter Father's Age :

    50

Enter Son's Age :

    25

Father's Age :50

Son's Age :25

Enter Father's Age :

    10

Enter Son's Age :

    20

Error: Son's age cannot be greater than father's age

Enter Father's Age:

    -1

Error: Age cannot be negative

Lab - 9 :-

```java
Class BMS Thread extends Thread {
    @override
    public void run() {
        while (true) {
            System.out.println ("BMS College of Engineering");
            try {
                Thread.sleep (10000);
            } Catch ( Interrupted Exception e) {
                e.printStackTrace ();
            }
        }
    }
}

Class CSEThead extends Thread {
    @override
    public void run () {
        while (true) {
            System.out.println ("CSE");
            try {
                Thread.sleep (2000);
            } Catch (Interrupted Exception e) {
                e.printStackTrace ();
            }
        }
    }
}
```

```java
public class Thread Example {
    public static void main (String [] args) {
        BMS Thread bms Thread = new BMSThread();
        bms Thread.start();

        CSE Thread cse Thread = new CSE Thread();
        cse Thread.start();
    }
}
```

Output:
BMS College of Engineering
 CSE
 CSE
 CSE
 CSE

BMS college of Engineering
CSE
CSE
CSE
CSE

Lab-10 :-

Class Q
{

    int n;
    boolean valueSet = false;
    Synchronized int get() {
            while (!valueSet)
            try {
            System.out.println ("\n Consumer waiting \n");
            wait ();
            } catch (Interrupted Exception e) {

                    System.out.println (" Interrupted Exception caught ")
            }

    System.out. println (" Got: " + n);
    value Set = false;
    System.out. println ("\n\n Intimate Producer \n") notify ();

    wait (); return n; }

synchronized void put (int n) {

    while (valueset)

    try {
        System.out. println ("\n Producer waiting \n");

    wait ();

    } catch (Interrupted Exception e) {
                System.out. println (" Intitputed Exception caught)
}

```java
        this. n = u;
        value. Set = true;
        System. out. print ("Put :" + u);
        System. out. print ("\n Internals 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 ()
```

```java
            System.out.prinln ("command "+R));
            i++;
        }
      }
    }
}

Class-PCforced{
    public static void main (String args[]){

        Q q = new Q();
        new Produn (q);
        new Consum (q);
        System.out.prinln ("Press Control-C to stop.");

    }
}
```

output :-

put: 0

intimate consum

Produn waiting

Got : 0

Intimate Produn

Put : 1

Intimate Consum

Produn waiting

Consumd : 0

Got = 1

Intimate produn

consumd : 1

put : 2

Intimat . produ

consumed : 2 ,

deadlock:-

```java
Class A
{
    synchronized void foo (B b)
    {
        String name = Thread.Current().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");
```

```java
    try
    {
        Thread.sleep(1000);
    }
    Catch (Exception e)
    {
        System.out.println("B Interrupted");
    }
    . Catch (Exception e)
    {
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
}
void last()
{
    System.out.println("Inside A.last");
}
}

Class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock()
    {
        Thread.currentThread().setName("Main Thread");
        Thread t = new Thread(this, "Kachy Thread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
```

```java
public void run()
{
    b.bar(a);
    System.out.println("Bcuk in other thread");
}

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

Output :=

Main Thread entered A-foo

Racing Thread entered B.bar

Racing Thread trying to call B.last()

Inside A.last

Bcuk in main thread

bcuk in other thread

13-02-M

```java
Lab-'

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

class UserInterface {

    UserInterface() {

        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);

        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and dividend");

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

        JButton button = new JButton("Calculate");

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

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

```java
ActionListener CalculateListener = new Action nes
    Public void action Performed (Action Event and) {
        try {
            int a = Integer. parseInt (ajtf. get Text ());
            int b = Integer. parse Int (bjtf. get Text ());
            if (b == 0) {
                throws new ArithmeticException ();
            }
            int ans = a/b;

            alab. set Text = (" \n A =" +a);
            blab. set Text = (" \n B =" + b);
            anslab. set Text (" \n Ans =" + ans);
            err. set Text (" ");
        } catch (NumberFormatException e) {

            display Error Message ("Enter Only Integers !");
        } catch (Arithmetic Exception e) {

            display Error Message (" B Should be non - Zero !");
        }
    }

    private void display Error Message (String message) {

        alab. set Text ("");
        blab. set Text ("");
        anslab. set Text (" ");
        err. set Text (message);
    }
};
```

button - add Action Listener (
jfrom. Set Visible (true);
}

public static void main (St
    Swing Utilities - invoke l
        public void run
            new UserI

});
}
}
}

→ Output :-

Enter the d

7

Calculat

```java
button.addActionListener( calculateListener );
jfram.setVisible(true);
}

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

→ <u>Output:-</u>

Enter the divider and dividend:

    7   4

Calculate   A=7 B=4 Ans = 1

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

```java
import java.util.Scanner;

class Quadratic
{
    int a, b, c;
    double r1, r2, d;
    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }
    void compute()
    {
    while(a==0)
        {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        d = b*b-4*a*c;
        if(d==0)
        {
            r1 = (-b)/(2*a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);
        }
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
```

```java
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1 = " + r1 + " + i"+r2);
            System.out.println("Root1 = " + r1 + " - i"+r2);
        }
    }
}

class quadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("1BM22CS214 REVANTH K");
    }
}
```

```java
                                    LAB 2
Develop a Java program to create a class Student with members usn,
name, an array credits and an
array marks. Include methods to accept and display details and a
method to calculate SGPA of a student.

import java.util.Scanner;

class Subject
{
    int 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);
    }
```

```java
    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)
```

```java
            {
                subject[i].grade = "F";
            }
        }
    }
    void computeSGPA()
    {
        int i;
        double sgpa;
        double totalcredits = 0;
        double totalgradepoints = 0;

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

```
        }
}
```

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

```java
import java.util.Scanner;

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

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

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

public class Mainbook
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int n;
        int i;
```

```java
        String name;
        String author;
        int price;
        int numPages;

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

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

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

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

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

```
                        LAB 4
Develop a Java program to create an abstract class named Shape that
contains two integers and an empty method named printArea( ).
Provide three classes named Rectangle, Triangle and Circle such that
each one of the classes extends the class Shape. Each one of the
classes contain only the method printArea( ) that prints the area of
the given shape.

import java.util.Scanner;

class inputScanner
{
```

```java
    protected Scanner scanner;

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

abstract class shape extends inputScanner
{
    double a, b;
    public shape()
    {
        super();
        System.out.println("the area of a : ");
        a = scanner.nextDouble();
        System.out.println("the area of b : ");
        b = scanner.nextDouble();
    }

}

class rectangle extends shape
{
    public rectangle()
    {
            super();
        }

    void area()
    {
    double area = a*b;
    System.out.println("the area of rectangle is : " + area);
    }
}

class triangle extends shape
{
    public triangle()
    {
            super();
        }

    void area()
    {
    double area = 0.5*a*b;
    System.out.println("the area of triangle is : " + area);
    }
```

```java
}

class circle extends shape
{
    public circle()
    {
            super();
        }

    void area()
    {
    double area = 3.14*a*a;
    System.out.println("the area of rectangle is : " + area);
    }
}

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

        rectangle r = new rectangle();
        triangle t = new triangle();
        circle c = new circle();

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

LAB 5
Develop a Java program to create a class Bank that maintains two
kinds of account for its
customers, one called savings account and the other current account.
The savings account
provides compound interest and withdrawal facilities but no cheque
book facility. The
current account provides cheque book facility but no interest.
Current account holders
should also maintain a minimum balance and if the balance falls
below this level, a service
charge is imposed.

Create a class Account that stores customer name, account number and
type of account.

From this derive the classes Cur-acct and Sav-acct to make them more specific to their
requirements. Include the necessary methods in order to achieve the following tasks:

a)
Accept deposit from customer and update the balance.

b)
Display the balance.

c)
Compute and deposit interest

d)
Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```java
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
        {
```

```java
            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)
        {
```

```java
            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();
```

```java
                            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;
                }
            }
        }
    }
```

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

the two packages in a file that declares the final marks of n
students in all five
courses.

```java
// Internals.java
package CIE;

import java.util.Scanner;

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


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

import java.util.Scanner;

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

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

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

```java
}


// Externals.java
package SEE;

import CIE.Internals;

import java.util.Scanner;

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

    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }

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

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

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

public class Main {
    public static void main(String args[]) {
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];
```

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

LAB 7

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

```java
import java.util.Scanner;

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


class InputScanner
{
 protected Scanner s;
 public InputScanner()
 {
  s = new Scanner(System.in);
 }
}
```

```java
class Father extends InputScanner
{
 protected int fatherAge;
 public Father() throws WrongAge
 {
  System.out.println("Enter Father's Age:");
  fatherAge=s.nextInt();

  if(fatherAge<0)
  {
   throw new WrongAge("Age cannot be negetive:");
  }
 }


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

}


class Son extends Father
{
 private int sonAge;

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


  if(sonAge>fatherAge)
  {
   throw new WrongAge("Son's age cannot be greater than father's age");
  }
  else if (sonAge<0)
  {
   throw new WrongAge("Age cannot be negative");
  }
 }

 public void display()
 {
  super.display();
  System.out.println("Son's Age: " + sonAge);
```

```java
 }

}


public class FatherSonAge
{
 public static void main(String args[])
 {
  try
  {
   Son son=new Son();
   son.display();
  }

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

}
```

LAB 8
Write a program which creates two threads, one thread displaying "BMS College
of Engineering" once every ten seconds and another displaying "CSE" once
every two seconds.

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

class CSEThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("CSE");
```

```java
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

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

LAB 9

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

```java
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
```

```java
        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);
```

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

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

LAB 10
Demonstrate Inter process Communication and deadlock

```java
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");
```

```java
    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.");
    }
}
```

```
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();
    }
```

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