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:

RANI AISHWARYA HS

1BM22CS217

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024.

javac EDGE

javac helloworld.java
java hello
→ class

~~output:-~~

12|12|25

→ class hello
{
public Static void main (String args[])
{
System.out.println("hello world");
}
}

output: hello world.

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

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

output:
Rani 1BM22CS217

Enter the coefficients of a,b,c
1
-5
2
Roots are real and distinct
Root1 = 4.561552812 Root2 = 4.561552812

output2: Rani 1BM22CS217
Enter the coefficients of a,b,c
1
2
1
roots are real and equal
Root1 = Root2 = 1.0

output 3:
Ram 1BM22CS217
Enter the coefficients of a,b,c
0
0
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

19/12/23

```
import java.util.Scanner;
class Subject
{
    int SubjectMarks;
    int Credits;
    int grade;
}
class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s= new Scanner (System.in).
    Subject subject[];
    Student ()
    {
        int i;
        subject= new Subject(9);
        for (i=0; i<9; i++)
        {
            subject[i]=new Subject();
        }
    }
    void getstudentdetails()
    {
        System.out.println" enter the name: ");
        name=s.next();
        System.out.println"enter the usn:");
        usn=s.next()
    }
    void get marks()
```

```
{
    for(int i = 0 ; i<6; i++)
    {
        System.out.println("enter the subject marks"+
        (i+1)+":");
        subject[i].subjectmarks = S.nextInt();
        if(subject[i].subjectmarks>=90)
        {
            subject[i].grade=10;
        }
        else if(subject[i].subjectmarks>=80)
        {
            subject[i].grade=9;
        }
        else if(subject[i].subjectmarks>=70)
        {
            subject[i].grade=8;
        }
        else if(subject[i].subjectmarks>=60)
        {
            subject[i].grade=7;
        }
        else if(subject[i].subject marks>=50)
        {
            subject[i].grade=6;
        }
        else if(subject[i].subjectmarks>=40)
        {
            subject[i].grade=5;
        }
        else
        {
```

```
            subject[i].grade=0;
        }
        System.out.println("enter the credits of the
        subject "+(i+1)+": ");
        Subject[i].credits=S.nextInt();
    }
}
void computesgpa()
{
    int totalcredits=0;
    int totalcredithours=0;
    for(int i=0;i<8;i++)
    {
        totalcredits+=(subject[i].grade*subject[i].credits);
        totalcredithours+=(subject[i].credits);
    }
    sgpa=(double)totalcredits/totalcredithours;
    System.out.println("SGPA:"+sgpa);
}
}
class studentMain
{
    public static void main(String args[])
    {
        student S1=new student();
        S1.getstudentdetails();
        S1.getmarks();
        S1.computesgpa();
    }
}
```

output:
enter the name:
Rani Aishwarya
Enter the USN:
1BM22CS217
Enter the subject marks of subject 1
90
enter the credits of the subject1
4
enter the subject marks of subject 2
87
enter the credits of subject 2
4
enter the subject marks of subject 3
76
enter the credits of subject 3
3
enter the subject marks of subject 4
92
enter the credits of subject 4
1
enter the subject marks of subject 5
89
enter the credits of subject 5
3
enter the subject marks of subject 6
78
enter the credits of subject 6
1
enter the subject marks of subject 7
67
enter the credits of subject 7
3

enter the subject marks of subject 8:
87
enter the credits of subject 8:
1

SGPA : 8.75

11/2/23

URBAN EDGE 26.12.23

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

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

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

output:

Name : Rani Aishwarya
USN : 1BM22CS217
Enter the number of books
2
Enter the name of the book
richard
Enter the author of the book
Vijay
Enter the price of the book
539
Enter the number of pages of the book
180
Enter the name of the book
Amma
Enter the author of the book
manikanth
Enter the number of pages of the book
price
390
Enter the number of pages of the book
140
Book name: richard
Author name: vijay
Price : 539
Number of pages : 180

Book name : amma
Author name : manikanth
price : 590
Number of page : 140

→ Additional problems:

```
import java.util.Scanner;
class Test
{
    int a,b;
    Test(int i,int j){
        a=i; b=j;
    }
    boolean equals(Test o){
        if(o.a==a && o.b==b) return true;
        else return false;
    }
}
class PassOb
{
    public static void main(String args[]){
        Test ob1=new Test(100,22);
        Test ob2=new Test(100,22);
        Test ob3=new Test(-1,-1);
        System.out.println("ob1==ob2:"+ob1.equals(ob2));
        System.out.println("ob1==ob3"+ob1.equals(ob3));
    }
}
```

output:
ob1==ob2 : true
ob1==ob3 : false.

02/01/24

URBAN EDGE

```
import java.util.Scanner;
abstract class shape
{
    double dim1, dim2, radius;
    shape(double a, double b)
    {
        dim1=a;
        dim2=b;
    }
    shape(double a)
    {
        radius=a;
    }
    abstract void area();
}
class Rectangle extends shape
{
    Rectangle(double a, double b)
    {
        super(a,b);
    }
    void area()
    {
        System.out.println("the area of rectangle"+
        (dim1*dim2));
    }
}
class Triangle extends shape
{
    Triangle(double a, double b)
    {
```

```
        super(a,b);
    }
    void area()
    {
        System.out.println("area of triangle" +
            (dim1*dim2)/2);
    }
}
class Circle extends shape
{
    Circle(double a)
    {
        super(a);
    }
    void area()
    {
        System.out.println("area of circle" + 3.14*
            (radius)*(radius));
    }
}
class AbstractAreaMain
{
    public static void main(String args[])
    {
        System.out.println("enter your name: Panishwara
            +1 s");
        System.out.println("enter your USN: 1BM22CS218");
        Scanner s= new Scanner(System.in)
        System.out.println("enter the length and breadth
            of the rectangle");
        double l=s.nextInt();
        double b=s.nextInt();
```

```
System.out.println("enter the base and height");
double h1 = s.nextInt();
double h2 = s.nextInt();
System.out.println("enter the radius of circle");
double r = s.nextInt();
shape sh;
Rectangle o = new Rectangle(l, b);
Triangle t = new Triangle(h1, h2);
Circle c = new Circle(r);
sh = o;
sh.area();
sh = t;
sh.area();
sh = r;
sh.area();
    }
}
```

output:

enter the name: RaniAishwarya H S

enter your USN: 1BM22CS212

enter the length and breadth of the rectangle

5

6

enter the base and height

8

9

enter the radius of circle

6

the area of rectangle 30.0

area of triangle 36.0

area of circle 15.039994

02/01/24

09/01/24

```
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, servia charges
```

```
            imposed: " + serviceCharge);
            balance = serviceCharge;
            System.out.println("balance: " + 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 acc type (current/saving):");
        String type1 = s.next();
        System.out.println("enter the account number:");
        int accno = s.next();
        System.out.println("enter the initial balance:");
        double balance = s.nextDouble();
        int ch;
        double amount1, amount2;
        account acc = new account(name, accno, type, balance);
        savAct sa = new savAct(name, accno, balance);
        curAct ca = new curAct(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 amount:");
    amount1 = s.nextInt();
        sa.deposit(amount1);
        break;
    case 2: System.out.println("entr 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 amount");
        amount1 = s.nextInt();
        ca.deposit(amount1);
        break;
        case 2: System.out.println("enter amount");
```

```
        amount = s.nextInt();
        ca.withdraw(amount2);
        ca.checkmin();
        break;
    case3: ca.display();
        break
    case4: System.exit(0);
    default: System.out.println("invalid input");
        break;
    }
  }
}
```

output: Name: RaviAishwarya [illegible]: IBM22[illegible]

enter the name:
ravi
enter the type (current/savings):
current
enter account number:
1234
enter initial balance:
4555
Menu
1. deposit 2 withdraw 3. display
enter the choice:
1
enter the amount:
23

Menu
1. deposit 2. withdraw 3. display
enter the choice
2
enter the amount
34545
insufficient balance, cant withdraw
Menu
1. deposit 2. withdraw 3. display
enter the choice
3
name: Rani accno: 1234 type: current balance: 4578.0

09/01/24

Name: PoojaAishwarya USN: 1BM21CS217

1)
```
public class main
{
    public static void Main(String args[])
    {
        String s1 = "hello world";
        System.out.println(s1);
        String s2 = new string("Hello world");
        System.out.println(s2);
        char [] char array = {'h','e','l','l','o'};
        String s3 = new String(char array);
        System.out.println(s3);
    }
}
```

output:-

Hello World
hello world
hello

2) I am 19 years old

3) output:-

Demensions are 10.0 by 14.0 by 12.0
Box b : Dimensions are 10.0 by 14.0 by 12.0

4) bmsce

5) 65
66
67

6) Welcome to bmsce college
Bmsce equals Bmsce → true
Bmsce equals college → false

Bmsce equals BMSCE → false
Bmsce equals Ignore case BMSCE → true

7) Substring is matched
S1 = "Bmsce college"
S2 = "Welcome to Bmsce college of engineering"

8) true
false.

9) true
false

10) S1 = Hello
S2 = Hello
Hello equals Hello → true
Hello == Hello → false

11) apple, ball, cat, dog, ent, fox, gun, hen, ice, jug, kite, lift, man, net, orange, parrot, queen, ring, star, tree, umbrella, van, watch, x-mas, yatch, zeal

12) Sorted number: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

13) It was a test. It was too

14) hello World

15) Commerce

16) Hello friend

17)
enter the no. of students: 3
enter the name: Rani
enter the register no: 567
enter the cgpa: 8.2
enter the name: Rushika
enter the register no.: 767
enter the cgpa: 9.8
~~enter the:~~
Sorted by cgpa
name: Rushika Regno.: 767 cgpa: 9.8
name: Rani Reg no: 567 cgpa: 8.2

18)
Set length: Hello
char At index 1: e
After set char At: Hillo
get chars: Hillo
After append: Hillo How are you?
After insert: Hello How awesome are you?
After delete: Hello How are you?
After delete char At: ello How are you?
After replace: Holo: how are you?

23/01/2024

Students.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 s = new Scanner(System.in);
        System.out.println("enter your name: Rani Ashwarya H S");
        System.out.println("enter your usn: 1bm22cs218");

        System.out.println("enter usn");
        int usn = s.nextInt();
        System.out.println("enter your name");
        char name = s.next();
        System.out.println("enter the sem");
        int sem = s.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("usn of this student:" +usn);
        System.out.println("name:" +name);
        System.out.println("sem:" +sem);
    }
}
```

→ internals.java

```
import java.util.Scanner;
package CIE;
public class internals extends Student
{
    protected int marks[] = new int[5];
    public void input(){ marks()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter internal marks for
                                five subjects");
        for (int j=0; j<5; j++)
            Marks[j] = s.nextInt();
    }
}
```

→ externals.java

```
package see;
import cie.internals;
import java.util.Scanner;
public class external extends internals
{
    protected int mark[] = new int[5];
    protected int finalMarks[] = new int[5];
    public void getexternal()
    {
        Scanner s = new Scanner(System.in);
        for (int i=0; i<5; i++)
        {
            System.out.println("enter the external
                marks of course" + (i+1));
```

```
            marks[i]=s.next.Int();
        }
    }
    public void cal()
    {
        for(int i=0;i<5;i++)
        {
            finalMarks[i]=marks[i]/2+Super.marks[i];
        }
    }
    public void displayFinal()
    {
        displayDetails();
        for(int i=0;i<5;i++)
        {
            System.out.println("course"+(i+1)+": "+
                                finalMarks[i]);
        }
    }
}
```

Main.java

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

```
        finalMarks[i] = new External();
        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();
    }
  }
}
```

output:

~~Rani Aishwarya~~

enter your name: RaniAishwarya H S
enter your USN: 1bm22cs217

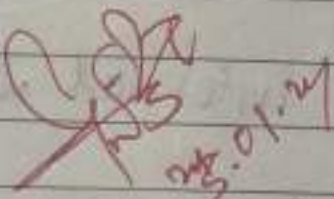
enter usn : 1BM22CS217
enter your name: Rani Aishwarya
enter the sem: 3
enter cie marks
enter the internal marks of course1
45
enter the internal marks of course2
46
enter the internal marks of course3
47
enter the internal marks of course4
48

enter the internal marks of course1
49
enter see marks
enter the external marks of course1
99
enter the external marks of course2
80
enter the external marks of course3
98
enter the external marks of course4
87
enter the external marks of course5
98

enter USN: 1BM22CS233
enter your name: Sakshi
enter the sem:3
enter cie marks
enter the internal marks of course1
45
enter the internal marks of course2
43
enter the internal marks of course3
46
enter the internal marks of course4
50
enter the internal marks of course5
20
enter see marks
enter the external marks of course1
98
enter the external marks of course2
99

enter the internal marks of course3
87
enter the external marks of course3
89
enter the external marks of course4
98

d Displaying details:
usn: 1BM22CS217 name: Ranjitha sem: 3
course1: 94
course2: 86
course3: 96
course4: 91
course5: 96
usn: 1BM22CS223 name: Lakshi sem: 3
course1: 94
course2: 92
course3: 84
course4: 94
course5: 89



30/01/24 URBAN EDGE

```
import java.util.Scanner;
class WrongAge extends Exception{
    public WrongAge(String message){
        super(message);
    }
}
class Father{
    private int age;
    public Father(int age) throws WrongAge{
        if(age<0){
            throw new WrongAge("age can't be negetive");
        }
        this.age = age;
    }
    public int getAge(){
        return age;
    }
}
class Son extends Father{
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws
        WrongAge{
        super(fatherAge);
        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
                negetive");
        }
        this.sonAge = sonAge;
    }
```

```
    public int getSonAge(){
        return sonAge;
    }
}
public class Main{
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        try{
            System.out.println("enter the father's age:\n");
            int fatherAge=scanner.nextInt();
            System.out.println("enter the son's age:\n");
            int sonAge= scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's Age:"+ son.getAge());
            System.out.println("Son's Age:"+ son.getSonAge());
        }catch(WrongAge e){
            System.out.println("Exception:"+e.getMessage());
        }finally{
            scanner.close();
        }
    }
}
```

output:- Name: Rani Aishwarya USN: 1BM22CS217

1) enter father's age:
   -5
   enter the son's age:
   10
   exception: age can't be negative

2) enter father's age:
   5

enter son's age:
7
exception: Son's age can't be greater than father's age.

enter father's age
19
enter son's age:
-5
exception: age can't be negative.

8
30/01/24

06/02/24

→ Multi Threading

```
class BMS extends Thread
{
    public void run()
    {
        for(int i=1; i<=5; i++)
        {
            System.out.println("BMS college of
            engineering "+i);
            try
            {
                Thread.sleep(10000);
            }
            catch(InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}
class CS extends Thread
{
    public void run()
    {
        for(int i=1; i<=5; i++)
        {
            System.out.println("CSE "+i);
            try
            {
                Thread.sleep(2000);
            }
            catch(InterruptedException e)
            {
```

```
            e.printStackTrace();
        }
    }
}
class threadMain
{
    public static void main(String args[])
    {
        BMS b1 = new BMS();
        CS c1 = new CS();
        b1.start();
        c1.start();
    }
}
```

output: Name: Punitashwaraya USN: 1BM22CS217

BMS college of engineering
CSE1
CSE2
CSE3
CSE4
CSE5
BMS college of engineering 2
BMS college of engineering 3
BMS college of engineering 4
BMS college of engineering 5

Inter process communication

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

```
class Produces implement Runnable{
 Q v;
Produces(Q v){
this.v=v;
new Thread(this,"Produces").start();
}
public void run(){
int i=0;
while(i<25){
v.put(i++);
}
}
}
class Consume implements Runnable{
Q v;
Consumer(Q v){
this.v=v;
new Thread(this,"Consumes").start();
}
public void run(){
    int i=0;
while(i<25){
int r=v.get();
System.out.println("consumed: "+r);
i++;
}
}
}
class PCFixed{
public static void main(String args[]){
Q v=new Q();
new producer(v);
```

new Consumer(N);
System.out.println("Press Control-C to stop");
}
}

output:-

Name: Ravi Aishwarya USN: 1BM22CS217

Put : 1
Got : 1
Put : 2
Got : 2
put : 3
Got: 3
put : 4
Got: 4
put: 5
Got : 5

06/02/24

13/02/24

URBAN EDGE

```
// 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);
        System.out.println("Back in main thread");
    }
    public void run(){
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]){
        new Deadlock();
    }
}
```

output:

Name: Ravishwaryan USN: 1BM22CS217

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

13/2/24

Lab programme. :09.

// User Interface to perform integer divisions

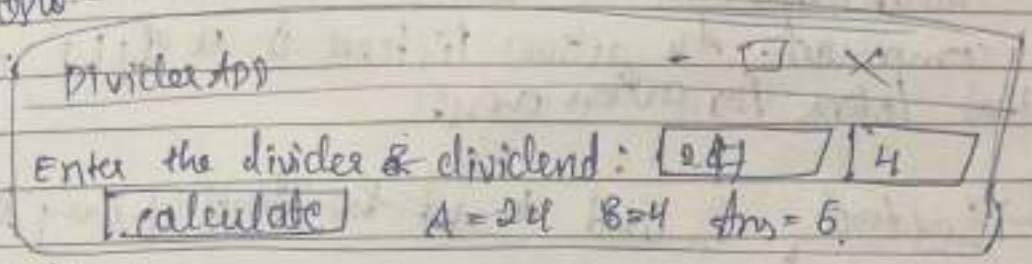
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class UserInterface{
    UserInterface(){
        JFrame jfrm= new JFrame("DividerApp");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab= new JLabel("Enter the divider & 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(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("");
        } 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);

jfrm.setVisible(true);
}
```

Name: Pari Aishwarya USN: 1BM22CS217

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

output:



Functions:

→ JFrame is used to create a graphical user interface for a simple division app

→ setsize() method is used to set the size of the JFrame

→ setlayout() method is used to set the layout manager for the JFrame. The layout manager is responsible for arranging the components that are added to the JFrame.

→ setDefaultCloseOperation() method is used to set the default close operation for the JFrame

→ JLabel class is used to create a label component which is a graphical user interface component that displays text or an image

→ JTextField class is used to create a text field component which is a GUI that allows users to enter and edit a single line of text.

→ addFrame is used to add components to the JFrame. The add() method is called on Frame object from

→ addActionListener() is used to add an action listener to a component. An action listener is a GUI component that listens for action events.

→ setText is used to set the text of any label component.

LAB - 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

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

LAB - 2

Sgpa calculator.

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);
    }

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

```
    void getMarks()
    {
        int i;
        for(i=0;i<8;i++)
        {
            System.out.println("enter the marks and credits for course " +
(i+1) + ":");
            System.out.println("marks : ");
            int marks = s.nextInt();
            System.out.println("credits : ");
            int credit = s.nextInt();
            subject[i].subjectMarks = marks;
            subject[i].credits = credit;

            if(marks >= 90 && marks<=100)
            {
                subject[i].grade = "O";
            }
            else if(marks>=80 && marks<90)
            {
                subject[i].grade = "A+";
            }
            else if(marks>=70 && marks<80)
            {
                subject[i].grade = "A";
            }
            else if(marks>=60 && marks<70)
            {
                subject[i].grade = "B+";
            }
            else if(marks>=50 && marks<60)
            {
                subject[i].grade = "B";
            }
            else if(marks>=40 && marks<50)
            {
                subject[i].grade = "C";
            }
            else if(marks>=0 && marks<40)
            {
                subject[i].grade = "F";
            }
```

```
        }
    }
    void computeSGPA()
    {
        int i;
        double sgpa;
        double totalcredits = 0;
        double totalgradepoints = 0;

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

```
    }
}
```

LAB - 3

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

```
import java.util.Scanner;

class Books
{
    String 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;
        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
{

    protected Scanner s;

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

    public int getInput(String message)
    {
        System.out.println(message);
        return scanner.nextInt();
    }



}

abstract class Shape extends inputScanner
{
    protected int a,b;

    public Shape()
    {
        super();
    }

    abstract public void printArea();
}

class Rectangle extends Shape
{
```

```
    protected int a,b;
    public Rectangle()
    {
        super();
    }

    public void printArea()
    {

        a=getInput("Enter the length:");
        b=getInput("Enter the 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();
    }
}
```

LAB - 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class 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 - 6

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

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

```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);
    }
}
```

```
// 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]);
    }
}
```

LAB - 7

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

```
import java.util.Scanner;

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

class Father
{
    protected int fatAge;
    public Father() throws WrongAge
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter father's age : ");
        fatAge = s.nextInt();
        if(fatAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    public void displayfat()
    {
            System.out.println("father's age is : "+fatAge);
    }
}

class Son extends Father
{
    private int sonAge;
    public Son() throws WrongAge
    {
        super();
```

```
        Scanner s = new Scanner(System.in);
        System.out.println("enter the son's age : ");
        sonAge = s.nextInt();
        if(sonAge >= fatAge)
        {
            throw new WrongAge("son's age is more than or equal to father's
age");
        }
        else if(sonAge<0)
        {
            throw new WrongAge("age cannot be negative");
        }
    }
    public void display()
    {
        System.out.println("son's age is : "+sonAge);
    }
}

public class Mainfatson
{
    public static void main(String args[])
    {
        try
        {
            Son son = new Son();
            son.displayfat();
            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.

```
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");
            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.

```
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 divident:");

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

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

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

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

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

        // 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

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<2) {
    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();
    }

    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();
    }
}
```