

# **B.M.S COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



## **OBJECT ORIENTED JAVA PROGRAMMING**

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**SHASHANK C**  
**1BM22CS254**

Department of Computer Science and Engineering  
B.M.S College of Engineering  
Bull Temple Road, Basavanagudi, Bangalore 560 019

# OOJ LAB RECORD

NAME :- SHASHANK . C

USN :- IBM22CS254

Std.:

Sub. 005

Telephone No.

Div.

Roll No.

Blood Group.

E-mail ID.

Birth Day.

Sr.No.	Title	Page No.	Sign/Remarks
1.	Quadratic Equation	10	28/12/12
2.	SGPA calculator	10	28/19/12
3.	Book Details	10	28/26/12/20
4.	Abstract class	10	28/31/2024
5.	Bank Accounts	10	28/16/1/2024
6.	Package Program	10	28/28/2024
7.	Exception Program	10	28/30/1/2024
8.	Multi Threading Program	10	28/6/2/2024
9.	IPC and Deadlock	10	28/13/2/2024
10.	User Interface (Divider App).		

## Lab Program - 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 discriminant  $b^2-4ac$  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:");
            a = s.nextInt();
        }
        d = b*b - 4*a*c;
        if(d==0)
    }
}
  
```

$$\text{r1} = (-b) / (2 * a);$$

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

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

`else if (dso)`

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

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

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

`System.out.println("Root1 = " + r1 + "-j" + r2);`

}

`}`

`q`

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

`Quadratic q = new Quadratic();`

`q.getd();`

`q.compute();`

}

`q`

~~done by~~

`System.out.println ("Shashank C. IBM22CS254");`

`Output :`

- ① Enter the coefficients of a, b, c  
 2, 4, 5

~~roots are imaginary~~

~~$$\text{root1} = -1.0 + j1.224744871391589$$~~

~~$$\text{root2} = -1.0 - j1.224744871391589$$~~

- ② Enter the coefficients of a, b, c  
 1, 2, 1

~~roots are equals~~

$$\text{root1} = \text{root2} = -1 = 0$$

③ Enter the coefficients of  $a, b, c$   
1, 4, 1

roots are real and distinct

$$\text{root1} = -0.2679491924311288$$

$$\text{root2} = -3.732050807568877$$

Done by Shashank C IBM22CS254

18  
12/12/2023

19/12/23

PAGE NO:  
DATE:

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

$$\text{SGPA formula} = \frac{\sum [(\text{course credit})(\text{Grade Points})]}{\sum [(\text{course credits})]}$$

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

```
class Subject {
```

```
    int subjectMarks;
```

```
    int credits;
```

```
    int 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);
```

```
y
```

```
void getMarks() {
```

```
    int i;
```

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

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



```
int marks = s.nextInt();
System.out.println("Enter the credits:");
int credits = s.nextInt();
subject[i].subjectMarks = marks;
subject[i].credits = credits;
subject[i].grade = (subject[i].subjectMarks / 10) + 1;
if (subject[i].grade == 11)
    subject[i].grade = 10;
}
if (subject[i].grade < 4)
    subject[i].grade = 0;
}

void computeSGPA getStudentDetails()
{
    System.out.println("Enter the name:");
    name = s.nextLine();
    System.out.println("Enter the usn:");
    usn = s.nextLine();
}

void computeSGPA()
{
    int i;
    int totalCredits = 0;
    int totalGradePoints = 0;
    for (i = 0; i < 9; i++)
    {
        totalCredits += subject[i].credits;
        totalGradePoints += subject[i].grade *
            subject[i].credits;
    }
    sgpa = (float) totalGradePoints / totalCredits;
    System.out.println("SGPA = " + sgpa);
}
```

}

class Main {

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

}

Output :

Enter the name :

chashank

Enter the usn :

1bm22cs254

enter marks :

74

enter the credits :

2

enter marks :

90

enter the credits :

4

enter marks :

88

enter the credits :

3

enter the marks :

73

enter the credits :

3

enter marks :

81

enter the credits :

3

enter marks :

86

enter the credits :

1

enter marks :

94

enter the credits :

1

enter marks : 0

enter credits : 0

SGPA = 8.899999618530273

done by Shashank C IBM22CS254

8/12/2023

Create a class Book which contains four members: name, author, price, numPages. 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 books. 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 Lab3d

```
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 the number of books:");  
    n = s.nextInt();  
    Books b[] = new Books[n];  
    for(int i=0; i<n; i++)  
    {  
        System.out.println("enter the details of book : " + i);  
        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 of the book: ");  
        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(int i=0; i<n; i++)  
    {  
        System.out.println(b[i].toString());  
    }  
}
```

Output :-

enter the number of books :

2

enter the details of book : 0

enter the name of the book :

harry

enter the author name :

rk

enter the price of the book :

1200

enter the number of pages :

45

enter the details of book : 1

enter the name of the book :

goosebump

enter the author name :

sr

enter the price of the book :

1500

enter the number of pages :

35

Book Details :

Book name : harry

Author name : rk

Price : 1200

Number of pages : 45

Book name : goosebump

Author name : sr

Price : 1500

Number of pages : 35

Done by shashank C IBM22CS254

2/1/2024

Develop a Java program to create an abstract class name 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.*;  
class inputScanner {  
    Scanner s;  
    public inputScanner()  
    {  
        s = new Scanner(System.in);  
    }  
    public double getDoubleInput(String message)  
    {  
        System.out.println(message);  
        return s.nextDouble();  
    }  
    public void closeScanner()  
    {  
        s.close();  
    }  
}  
abstract class Shape extends inputScanner  
{  
    double dim1;  
    double dim2;  
    shape()  
    {  
        super();  
    }  
}
```

3

public abstract void printArea();

class Rectangle extends Shape

public Rectangle()

super();

public void printArea()

dim1 = getDoubleInput("enter length of rectangle");

dim2 = getDoubleInput("enter breadth of rectangle");

double area = dim1 \* dim2;

System.out.println("area of rectangle: " + area);

class Triangle extends Shape

public Triangle()

super();

public void printArea()

dim1 = getDoubleInput("enter base of triangle");

dim2 = getDoubleInput("enter height of triangle");

double area = 0.5 \* dim1 \* dim2;

System.out.println("area of triangle: " + area);

class Circle extends Shape

```
public Circle()
{
    super();
}

public void printArea()
{
    dim = getDoubleInput("Enter radius of circle");
    double area = Math.PI * dim * dim;
    System.out.println("Area of circle: " + area);
}

public class AreaDemo
{
    public static void main(String[] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
        r.closeScanner();
    }
}
```

Output :-

enter length of rectangle : 4

enter breadth of rectangle : 5

area of rectangle : 20.0

enter base of triangle : 2

enter height of triangle : 3

area of triangle : 3.0

enter radius of circle : 3

area of circle : 28.274333882308138

16/11/2024

Develop a Java program to develop a class Bank with current account and saving account.

```
import java.util.*;  
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 ("No sufficient balance");  
    }  
    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

{  
 private double minBal = 1000;  
 private double serviceCharges = 150;  
 CurAcct(String name, int accno, double balance) {  
 super(name, accno, "current", balance);  
 }  
}

void checkMin() {  
 if (balance < minBal)  
 System.out.println("Balance is low service charges  
are added :" + serviceCharges);  
 balance -= serviceCharges;  
 System.out.println("balance :" + balance);  
}

public class accountMain {  
 public static void main (String [] args) {  
 Scanner sc = new Scanner (System.in);  
 System.out.println ("enter the name :");  
 String name = sc.next();  
 System.out.println ("enter the type of account :");  
 }  
}

int accno

string-type = sc.next();

System.out.println("enter account number:");

int accno = sc.nextInt();

System.out.println("enter the initial balance:");

double balance = sc.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("1.Menu 2.Deposit")

2.Withdraw 3.Interest 4.Display");

ch = sc.nextInt();

switch(ch)

{

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

amount1 = sc.nextInt();

sa.deposit(amount1);

break;

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

amount2 = sc.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("InMenu In 1.Deposit In 2.  
withdraw In 3.Display In");

System.out.print("enter the choice:");  
ch = sc.nextInt();

switch(ch)  
{

case 1: System.out.print("enter the amount:");  
amount1 = sc.nextInt();  
ca.deposit(amount1);  
break;

case 2: System.out.print("enter the amount:");  
amount2 = sc.nextInt();  
ca.withdraw(amount2);  
ca.checkMin(amount2);  
break;

case 3: ca.display();  
break;

case 4: System.exit(0);

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

}  
}

Output :-

enter the name :

S

enter the type of account :

current

enter account number :

1

enter the initial balance :

1000

Menu

1. Deposit

2. withdraw

3. Display

enter the choice :

1

enter the amount :

500

Menu

1. Deposit

2. withdraw

3. Display

enter the choice :

3

name : S accno : 1 type : current balance : 1500 . 0

Menu

1. Deposit

2. withdraw

3. Display

enter the choice : 2

enter the amount :

1500

Menu

1. Deposit

2. withdraw

3. Display

enter the choice :

3

name's accno: 1 type: current balance: 0.0  
Menu

1. Deposit

2. withdraw

3. Display

enter the choice:

2

enter the amount:

100

Balance is low service charges are added: 150.0

balance: -150.0

NO sufficient balance to withdraw.

enter the name:

3

enter the type of account:

saving

enter account number:

1

enter the initial balance:

1000

Menu

1. Deposit

2. withdraw

3. Display

enter the choice:

1

enter the amount:

500

Menu

1. Deposit

2. withdraw

3. Display

enter the choice: 2

enter the amount : 500

Menu

1. Deposit

2. withdraw

3. Display

enter the choice: 3

name : s accno: 1 type:saving balance: 1000.0

Done by shachank c IBM22CS25LJ.

16/1/2024

Develop a java program and create a CIE package consists of student and Internal classes and SEE package consists of External and main class and display the students marks in internals & externals.

1/ 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 Input student details()
    {
```

```
        Scanner s = new Scanner (System.in);
        System.out.println ("enter USN:");
        USN = s.nextLine();
        System.out.println ("enter Name:");
        name = s.nextLine();
        System.out.println ("enter semester:");
        sem = s.nextInt();
```

y

```
    public void display student details()
    {
```

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

✓ System.out.println ("sem:" + sem);

y

y

### //Internals.java

```
package CIE;  
import java.util.Scanner;  
public class Internals extends Student  
{  
    protected int marks[] = nextInt[5];  
    public void input(SEEmarks())  
    {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter internal marks for "+name);  
        for(i=0; i<5; i++)  
        {  
            System.out.print("subject "+(i+1)+" marks: ");  
            marks[i] = s.nextInt();  
        }  
    }  
}
```

### /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 input.SEEmarks()  
    {  
        Scanner s = new Scanner(System.in);  
        System.out.println("Enter SEEmarks for "+name);  
        for(i=0; i<5; i++)  
    }  
}
```

```
System.out.println("Subject "+(i+1)+" marks:");  
marks[i] = s.nextInt();
```

3

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

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

- 4

//Main.java.

```
import SEP.Externals;  
public class Main {  
    public static void main (String args[]) {  
        int num_of_students = 2;  
        Externals finalMarks[] = new Externals  
            (num_of_students);  
  
        for (int i = 0; i < num_of_students; i++) {  
            finalMarks[i] = new Externals();  
            finalMarks[i].inputStudentDetails();  
            System.out.println("Enter AIE marks:");  
            finalMarks[i].inputAIEMarks();  
            System.out.println("Enter SEP marks:");  
            finalMarks[i].inputSEPMarks();  
        }  
    }  
}
```

```
System.out.println("Display data:");  
for (int i = 0; i < num_of_students; i++) {  
    finalMarks[i].calculateFinalMarks();  
}
```

finalmarks(); display FinalMarks();

3  
3

Entered USN:

IBNM22CS254

Enter name:

Shashank C

Entered SEM:

3

Entered CIE marks:

Entered Internal marks:

Subject 1: 45

Subject 2: 46

Subject 3: 42

Subject 4: 43

Subject 5: 44

Enter SEE marks:

Enter SEE marks for Shashank C.

Subject 1: 85

Subject 2: 89

Subject 3: 87

Subject 4: 88

Subject 5: 87

Display data:

USN: IBNM22CS254

Name: Shashank C.

Sem: 3

Subject 1: 87

Subject 2: 90

Subject 3: 85

Subject 4: 87

Subject 5: 87

Q4  
24/24

Syllabus

Write a program that demonstrates handling exceptions in inheritance tree. Create a base class called "Father" and derived class "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  $\geq$  father's age.

```
import java.util.*;  
class WrongAge extends Exception{  
    public WrongAge(String s){  
        super(s);  
    }  
}  
  
class InputScanner{  
    protected Scanner s;  
    public InputScanner(){  
        s = new Scanner(System.in);  
    }  
}  
  
class Father extends InputScanner{  
    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");  
        }  
    }  
    public void display(){  
        System.out.println("Age is " + fatherAge);  
    }  
}
```

```
System.out.println("father's age:" + fatherAge);
```

{

```
class Son extends Father {
```

```
    int sonAge;
```

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

```
        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() {
```

```
        System.out.println("son's Age " + sonAge);
```

{

```
public class ExceptionDemo {
```

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

```
        try {
```

~~Son son = new Son();~~~~son.display();~~

```
        catch (WrongAge e) {
```

~~System.out.println ("NegativeError" + e)~~~~System.out.println (e.getMessage());~~

{

{

Output:-

Enter father's age:

41

Enter son's Age:

18

Son's Age 18

Enter fathers age:

41

Enter son's Age:

15

Negative Error Wrong Age: Son's age cannot be greater than father's age

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

Enter father's age:

41

Enter son's Age:

-21

Negative Error Wrong Age: Age cannot be negative

Age cannot be negative.

Enter father's age:

45

Enter son's Age:

48

Negative Error Wrong Age: Sons age and fathers age cannot be same

~~Sons age and father's age cannot be same~~

30/1/2024

6/6/24

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 first extends Thread

{  
public void run()

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

System.out.println("BMS College of Engineering");  
Thread.sleep(10000);

}

catch (InterruptedException ie)

{

System.out.println("The sleeping thread woken");

}

}

class second extends Thread

{  
public void run()

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

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

try {

System.out.println("CSE");

Thread.sleep(2000);

}

catch (InterruptedException ie)

{

System.out.println("The sleeping thread woken");

}

}

}

class lab 8

{

public static void main(String[] args)

{

Thread t1 = new Thread(new first());

Thread t2 = new Thread(new second());

t1.start();

t2.start();

}

Output :

RMSCE college of Engineering

CSE

CSE

CSE

CSE

CSE

~~RMSCE college of Engineering~~

CSE

CSE

CSE

CSE

~~RMSCE college of Engineering~~

CSE

CSE

CSE

12/2024  
CSE

Demonstrate Inter process communication and deadlock.

⑩ @ class & q (IPC)

int n;  
boolean valueset = false;

synchronized int get() {

    while (!valueset)

        try {

            System.out.println("consume waiting\n");  
            wait();

        } catch (InterruptedException e) {

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

        System.out.println("InterruptedException got " + n);  
        valueset = false;

        System.out.println("In Intimate Producer\n");

        notify();

    return n;

}

    synchronized void put(int n) {

        while (valueset)

            try {

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

            } catch (InterruptedException e) {

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

            this.n = n;

            valueset = true;

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

            System.out.println("In Intimate consumer\n");

            notify();

}

{

class Producer implements Runnable

{ q;

Producer(Q q)

{

this.q=q;

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

{

public void run()

{ int i=0;

while(i<6)

{

q.put(i++);

{

{

class Consumer implements Runnable

{ q;

Consumer(Q q)

{

this.q=q;

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

{

public void run()

{ int i=0;

while(i<6)

{

int x=q.get();

System.out.println("consumed: "+x);

i++;

{

4  
class PCFixed d

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

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

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

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

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

5

Output:

Press Control-c to stop

Put : 0

Intimate consumer

Producer waiting

Get : 0

Intimate Producer

consumed : 0

put : 1

Intimate Producer

consumed : 0

put : 1

Intimate consumer

~~Producer waiting~~

Get : 1

Intimate Producer

consumed : 1

put : 2

Intimate consumer

~~Producer waiting~~

Get : 2

Intimate Producer

consumed : 2

Put : 3

Intimate Producer  
consumed : 2

Put : 3

Intimate Consumer

Producer waiting

Brof : 3

Intimate Producer

consumed : 3

Put : 4

Intimate consumer

Producer waiting

Brof : 4

Intimate Producer

consumed : 4

Put : 5

Intimate consumer

Producer waiting

Brof : 5

Intimate Producer

consumed : 5

Put : 6

Intimate consumer

Producer waiting

Brof : 6

⑩ ⑥

## 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 mainThread");
```

```
}
```

```
public void run()
```

```
{
```

```
    b.bar(a);
```

```
    System.out.println("Back in other thread");
```

```
}
```

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

```
{
```

```
    new Deadlock();
```

```
}
```

Y

Output :-

MainThread entered A.foo

RacingThread entered B.bar

MainThread trying to call B.last()

Inside A.last

Back in main thread

RacingThread trying to call A.last()

Inside A.last

~~Bact in other thread~~

SG  
13/2/2024

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 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo {  
    private JFrame jfrm;  
    private JTextField ajtf, bjtf;  
    private JLabel alab, blab, anslab, err;  
  
    public SwingDemo() {  
        jfrm = new JFrame("Divide App");  
        jfrm.setSize(275, 200);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JLabel jlab = new JLabel("Enter the  
divisor and dividend");  
        ajtf = new JTextField(8);  
        bjtf = new JTextField(8);  
        JButton button = new JButton("calculate");  
        err = new JLabel();  
        alab = new JLabel();  
        blab = new JLabel();  
        anslab = new JLabel();
```

```
jfbm.add(ex8);  
jfrm.add(jlab);  
jfrm.add(ajtf);  
jfrm.add(bjtf);  
jfrm.add(button);  
jfrm.add(alab);  
jfrm.add(blab);  
jfrm.add(anslab);
```

```
button.addActionListener(new ActionListener()) {
```

```
    public void actionPerformed(ActionEvent evt) {  
        calculateDivision();  
    }  
};
```

```
jfrm.setVisible(true);
```

```
private void calculateDivision() {
```

```
    try {
```

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

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

```
        if (b == 0) {
```

```
            throw new ArithmeticException("Division by zero!");
```

```
        }
```

```
        int ans = a/b;
```

~~```
a1ab.setText("A = " + a);
```~~~~```
blab.setText("B = " + b);
```~~~~```
anslab.setText("Ans = " + ans);
```~~~~```
err.setText(" "));
```~~

```
} catch (NumberFormatException e) {
```

~~```
    alab.setText(" "));
```~~~~```
    blab.setText(" "));
```~~~~```
    anslab.setText(" "));
```~~

ex. `System.out.println("B should be non-zero!");`

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

Output :-

Enter the divisor and dividend

    

Calculate. A = 4    B = 1    Ans = 4,

Functions:-

- 1) `JFrame` :- It is a top level container in Java, string that represent a window with a title bar, border and optional menu.
- 2) `setSize` :- It is used to set size of the frame.
- 3) `setLayout` :- This line sets the layout manager for the frame to `FlowLayout` which arranges components from left to right in a flow like manner.
- 4) `Add` :- This line adds the user label to the frame.

5) `setVisible` :- This line makes the frame visible.

5  
21/2/2024

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 discriminant  $b^2 - 4ac$  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("root1="+r1+"root2="+r2);

    }

    else if(d<0)

    {

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

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

        r2=Math.sqrt(-d)/(2*a);

        System.out.println("root1="+r1+"+i"+r2);

        System.out.println("root1="+r1+"-i"+r2);

    }

}

class QuadraticMain

{

    public static void main(String args[])

    {

        Quadratic q=new Quadratic();

        q.getd();

        q.compute();

        System.out.println("Shashank C USN-1BM22CS254");

    }

}

```

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;
    int grades;
}

class Student{
    Subject subject[];
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Student(){
        subject = new Subject[9];
        for(int i = 0;i<9;i++){
            subject[i] = new Subject();
        }
        s= new Scanner(System.in);
    }
    void getStudentDetails(){
        System.out.println("Enter your name: ");
        this.name = s.nextLine();
        System.out.println("Enter your usn: ");
        this.usn = s.next();
    }
    void getMarks(){
        for(int i=0;i<8;i++){
            System.out.println("Enter the marks of the "+(i+1)+" subject");
        }
    }
}
```

```
subject[i].subjectMarks = s.nextInt();

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

subject[i].credits = s.nextInt();

subject[i].grades = (subject[i].subjectMarks/10)+1;

if(subject[i].grades >10)

subject[i].grades = 10;

if(subject[i].grades <4)

subject[i].grades = 0;

}

}

void computeSGPA(){

int sum=0;

int totalCredits=0;

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

sum+=(subject[i].grades*subject[i].credits);

totalCredits+=subject[i].credits;

}

this.SGPA=(double)sum/totalCredits;

}

}

class Main2{

public static void main(String args[]){

Student s1=new Student();

s1.getStudentDetails();

s1.getMarks();

s1.computeSGPA();

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

System.out.println("Usn: "+s1.usn);

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

System.out.println("Shashank C USN-1BM22CS254");

}
```

```
}
```

```
}
```

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

```
class Main3
{
    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");
        n=s.nextInt();
        Book b[];
        b=new Book[n];
        for(i=0;i<n;i++)
        {
            System.out.println("Enter the details of the book:"+i);
            System.out.println("Enter the name of the book:");
            name=s.nextLine();
            System.out.println("Enter the author of the book:");
            author=s.nextLine();
            System.out.println("Enter the price of the book:");
            price=s.nextInt();
            System.out.println("Enter the no. of pages of the book:");
            numPages=s.nextInt();
            b[i]=new Book(name,author,price,numPages);
        }
        System.out.println("Book details:");
    }
}
```

```

for(i=0;i<n;i++)
{
    System.out.println(b[i].toString());
}
System.out.println("Shashank C USN-1BM22CS254");
}
}

```

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.*;
class inputScanner{
    protected Scanner s;
    public inputScanner()
    {
        s=new Scanner(System.in);
    }
    public double getDoubleinput(String message)
    {
        System.out.println(message);
        return s.nextDouble();
    }
    public void closeScanner()
    {
        s.close();
    }
}
abstract class Shape extends inputScanner
{

```

```
protected double dim1;  
protected double dim2;  
Shape()  
{  
    super();  
}  
public abstract void printArea();  
}  
  
class Rectangle extends Shape  
{  
    public Rectangle()  
    {  
        super();  
    }  
    public void printArea()  
    {  
        dim1=getDoubleinput("enter length of the rectangle");  
        dim2=getDoubleinput("enter breadth of the rectangle");  
        double area=dim1*dim2;  
        System.out.println("area of the rectangle:"+area);  
    }  
}  
  
class Triangle extends Shape  
{  
    public Triangle()  
    {  
        super();  
    }  
    public void printArea()  
    {
```

```
    dim1=getDoubleinput("enter base of the triangle");

    dim2=getDoubleinput("enter height of the triangle");

    double area=0.5*dim1*dim2;

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

}

}

class Circle extends Shape

{

public Circle()

{

super();

}

public void printArea()

{

    dim1=getDoubleinput("enter radius of the circle");

    double area=Math.PI*dim1*dim1;

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

}

}

public class Area

{

public static void main(String[] args)

{

    Rectangle r=new Rectangle();

    Triangle t=new Triangle();

    Circle c=new Circle();

    r.printArea();

    t.printArea();

    c.printArea();

    r.closeScanner();

}
```

```
        System.out.println("Shashank C USN-1BM22CS254");  
    }  
}
```

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.*;  
  
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 to withdraw");  
    }  
}  
  
void display()  
{  
    System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance"+balance);  
}  
}  
  
class savings extends account  
{  
    private static double rate=10;  
    private static double time=1;  
    savings(String name,int accno,String type,double balance)  
    {  
        super(name,accno,"savings",balance);  
    }  
    void interest()  
    {  
        balance+=balance*java.lang.Math.pow(1-(rate/100),time);  
    }  
}
```

```
System.out.println("balance="+balance);

}

}

class current extends account

{

private static double minbal=1000;

private static double service_taxes=50;

current(String name,int accno,String type,double balance)

{

super(name,accno,"current",balance);

}

void checkmin()

{

if(balance<minbal)

{

System.out.println("balance is low service taxes are added"+service_taxes);

balance-=service_taxes;

System.out.println("balance="+balance);

}

}

}

public class accountmain

{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

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

String name=s.nextLine();

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

String type=s.nextLine();
```

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

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

int ch;

double amount1,amount2;

account a0=new account(name,accno,type,balance);
savings s0=new savings(name,accno,type,balance);
current c0=new current(name,accno,type,balance);

while(true)

{
    if(a0.type.equals("savings"))

    {
        System.out.println("enter 1.deposit 2.withdraw 3.interest 4.display");
        System.out.println("enter the choice");
        ch=s.nextInt();
        switch(ch)

        {

            case 1:System.out.println("enter the amount to be deposited");
            amount1=s.nextDouble();
            s0.deposit(amount1);
            break;

            case 2:System.out.println("enter the amount to be withdrawn");
            amount2=s.nextDouble();
            s0.withdraw(amount2);
            break;

            case 3:s0.interest();
            break;

            case 4:s0.display();
            break;
        }
    }
}
```

```
        case 5:System.exit(0);

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

            break;

        }

    }

else

{

    System.out.println("enter 1.deposit 2.withdraw 3.display");

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

    ch=s.nextInt();

    switch(ch)

    {

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

            amount1=s.nextDouble();

            c0.deposit(amount1);

            break;

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

            amount2=s.nextDouble();

            c0.checkmin();

            c0.withdraw(amount2);

            break;

        case 3:c0.display();

            break;

        case 4:System.exit(0);

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

            break;

    }

}

System.out.println("Shashank C USN-1BM22CS254");
```

```
 }  
 }
```

6.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.

```
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 usn");  
  
usn=s.nextLine();  
  
System.out.println("enter name");  
  
name=s.nextLine();  
  
System.out.println("enter sem");  
  
sem=s.nextInt();  
  
}  
  
public void displayStudentDetails()  
{  
  
System.out.println("USN"+usn);  
  
System.out.println("Name"+name);  
  
System.out.println("Sem"+sem);  
  
}  
}  
  
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(i=0;i<5;i++)
        {
            System.out.println("SUbject"(i+1)+"marks:");
            marks[i]=scanner.nextInt();
        }
    }
}

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

```

```
finalMarks[i] = marks[i]/2 + super.marks[i];  
}  
  
public void displayFinalMarks() {  
    displayStudentDetails();  
    for(int i=0;i<5;i++)  
        System.out.println("Subject " + (i+1) + ":" + finalMarks[i]);  
}  
}  
  
import SEE.Externals;  
  
class main6 {  
    public static void main(String args[])  
    {  
        int numOfStudents = 2;  
        Externals finalMarks[] = new  
        Externals[numOfStudents];  
        for(int i=0;i<numOfStudents;i++)  
        {  
            finalMarks[i] = new Externals();  
            finalMarks[i].inputStudentDetails();  
            System.out.println("Enter CIE marks");  
            finalMarks[i].inputCIEmarks();  
            System.out.println("Enter SEE marks");  
            finalMarks[i].inputSEEmarks();  
        }  
        System.out.println("Displaying data:\n");  
        for(int i=0;i<numOfStudents;i++)  
        {  
            finalMarks[i].calculateFinalMarks();  
            finalMarks[i].displayFinalMarks();  
        }  
    }
```

```
System.out.println("Shashank C USN-1BM22CS254");  
}  
}
```

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 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");
}

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

}

class Son extends Father
{
protected int SonAge;
public Son() throws WrongAge
{
    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");
    }
    if(SonAge==FatherAge)
    {
        throw new WrongAge("Son's age and Father's age can't be same");
    }
    if(SonAge<0)
    {
        throw new WrongAge("Age cannot be negative");
    }
}
```

```

}

public void display()
{
    super.display();
    System.out.println("Son's age:"+SonAge);
}

}

public class Main7
{
    public static void main(String args[])
    {
        try
        {
            Son son=new Son();
            son.display();
        }
        catch(WrongAge e)
        {
            System.out.println("Error:"+e.getMessage());
        }
        System.out.println("Shashank C USN-1BM22CS254");
    }
}

```

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 thread1 implements Runnable
{
    public void run()
    {

```

```
for(int i=0;i<5;i++){  
    try  
    {  
        System.out.println("BMS College Of Engineering");  
        Thread.sleep(10000);  
    }  
    catch(InterruptedException ie)  
    {  
        System.out.println("The sleeping thread is woken up");  
    }  
}  
  
class thread2 implements Runnable  
{  
    public void run()  
    {  
        for(int j=0;j<5;j++){  
            for(int i=0;i<5;i++){  
                try  
                {  
                    System.out.println("CSE");  
                    Thread.sleep(2000);  
                }  
                catch(InterruptedException ie)  
                {  
                    System.out.println("The sleeping thread is woken up");  
                }  
            }  
        }  
    }  
}
```

```

    }
}

class Main8
{
    public static void main(String[] args)
    {
        Thread t1=new Thread(new thread1());
        Thread t2=new Thread(new thread2());
        t1.start();
        t2.start();
        System.out.println("Shashank C USN-1BM22CS254");
    }
}

```

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

```

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

public class SwingDemo {
    private JFrame jfrm;
    private JTextField ajtf, bjtf;
    private JLabel alab, blab, anslab, err;
    public SwingDemo() {
        jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 200);
        jfrm.setLayout(new FlowLayout());

```

```
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JLabel jlab = new JLabel("Enter the divider and dividend:");
ajtf = new JTextField(8);
bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

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

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        calculateDivision();
    }
});

jfrm.setVisible(true);
}

private void calculateDivision() {
    try {
        int a = Integer.parseInt(ajtf.getText());
        int b = Integer.parseInt(bjtf.getText());
        if (b == 0) {
            throw new ArithmeticException("Division by zero!");
        }
    }
}
```

```

        }

        int ans = a / b;

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

    } catch (NumberFormatException e) {

        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter only integers!");

    } catch (ArithmaticException e) {

        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be non-zero!");

    }
}

public static void main(String args[]) {

    SwingUtilities.invokeLater(new Runnable() {

        public void run() {

            new SwingDemo();

        }

    });

    System.out.println("Shashank C USN-1BM22CS254");

}
}

```

10.a) Demonstrate Inter process Communication.

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

```
}

}

class Producer implements Runnable {

    Q q;

    Producer(Q q) {

        this.q = q;

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

    }

    public void run() {

        int i = 0;

        while(i<15) {

            q.put(i++);

        }

    }

}

class Consumer implements Runnable {

    Q q;

    Consumer(Q q) {

        this.q = q;

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

    }

    public void run() {

        int i=0;

        while(i<15) {

            int r=q.get();

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

            i++;

        }

    }

}
```

```

class PCFixed {

    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("Shashank C USN-1BM22CS254");
    }
}

```

b).Demonstrate 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();
    System.out.println("Shashank C USN-1BM22CS254");
}
```