VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

StudentName (1BM23CS000)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **StudentName (1BM23CS000),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/aishwarya-2005-cs/java-lab-programs

Program 1
Implement Quadratic Equation

Algorithm:

static void add(){ System.out.println("Good Morning");}
}

OUTPUT:
Hello GoodMorning

LAB-2

30-9-2024
9) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.lang.Math;
import java.util.Scanner;
class Quadratic
public static void main(String args[])
{
  double r1, r2, d;
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter coefficient of a");
  int a = sc.nextInt();
  System.out.println("Enter coefficient of b");
  int b = sc.nextInt();
  System.out.println("Enter coefficient of c");
  int c = sc.nextInt();
  if (a==0.0){
```

Date: / /

```
System.out.println("not a quadratic equation");
}
else {
d = b*b - 4*a*c;
if (d==0.0){
r1 = (-d) / (2*a);
System.out.println("Roots are real and equal");
}
else if (d>0.0)
{
r1 = ((-b) + (Math.sqrt(d))) / (double)(2*a);
r2 = ((-b) - (Math.sqrt(d))) / (double)(2*a);
System.out.println("Root are real and distinct");
}
else if (d<0.0)
{
r1 = (-b) / (2*a);
r2 = Math.sqrt(-d) / (2*a);
System.out.println("Roots are imaginary");
}
}
}
}
```

OUTPUT:

① enter coefficient of a
1
enter coefficient of b
2
enter coefficient of c
3
Roots are imaginary

② enter coefficient of a
3
enter coefficient of b
-4
enter coefficient of c
5
Roots are imaginary.

Date: / /

Enter coefficient of a

1

Enter coefficient of b

2

Enter coefficient of c

1

Roots are real and equal.

O/P run

Qua

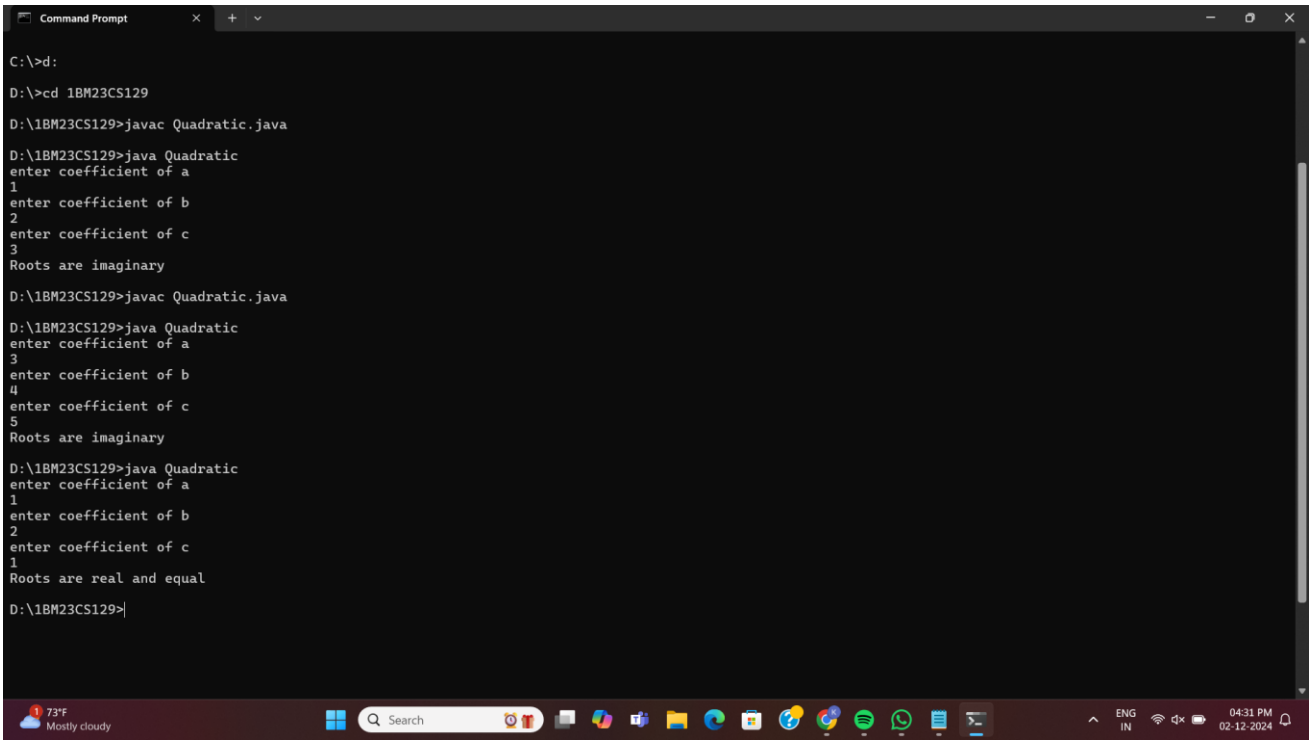
30-9-24

-10-2024 Lab Program - 2.

To create a class student w

Code:

```
import java.lang.Math;
import java.util.Scanner;
class Quadratic{
public static void main(String args[])
{
 double r1,r2, d;
 Scanner sc=new Scanner(System.in);
System.out.println("enter coefficient of a");
int a=sc.nextInt();
System.out.println("enter coefficient of b");
int b=sc.nextInt();
System.out.println("enter coefficient of c");
int c=sc.nextInt();
if(a==0.0){
System.out.println("not a quadratic equation");
}
else{
d=b*b-4*a*c;
 if(d==0.0){
r1=(-d)/(2*a);
System.out.println("Roots are real and equal");
}
else if(d>0.0)
{
r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real and distinct");
}
else if(d<0.0)
{
r1=(-b)/(2*a);
r2=Math.sqrt(-d)/(2*a);
System.out.println("Roots are imaginary");
}
}
}
}
```



(You should repeat as above, for all the remaining programs from 2 to 10)

Program 2
Implement Scgpa Equation

Algorithm:

Roots are real and equal.

Viva
30-9-24

17-10-2024 Lab Program - 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.

subject[i].grade = subject[i].subjectmarks/10 + 1

SGPA = score / credit

```
import java.util.Scanner;
class subject{
int subjectmarks;
int credit;
int grade;
void calgrade()
{ grade=(subject marks/10)+1;
  if (grade == 11){
```

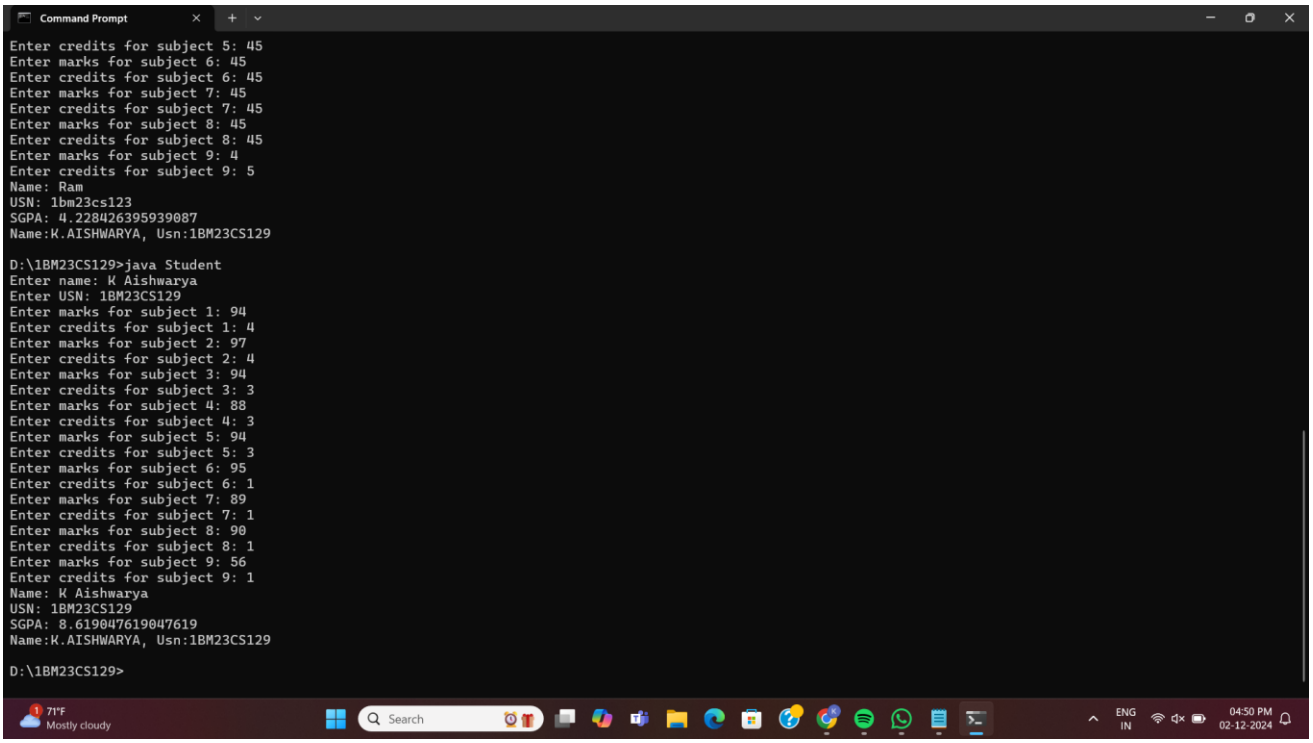
```
Call Super Class
        { grade = grade - 1; }
        else if (grade < 1)
        { grade = 0;
        }
    }
}
class student {
string name;
string usn;
double sgpa;
double temp;
int totalCredit=0;
subject subject[];
Scanner s;
student(){
int i;
subject = new subject [8];
"A")    for(i=0; i<5; i++)
            subject[i]= new subject();
        s=new Scanner(System.in);
    }
    void getstudentdetails()
    { System.out.println("Enter name and usn of student");
      name = s.next();
      usn = s.next();
)   }
    void getmarks()
    { System.out.println("Enter marks and
      credits of subject" + (i+1));
      for(int i=0; i<8; i++)
      { System.out.println("Enter marks and
        credits of subject" + (i+1));
```

```
class subject[i].subjectmarks = s.nextInt();
subject[i].credit = s.nextInt();
subject[i].calgrade();
}}
void calsgpa()
{ for(int i=0;i<8;i++)
  { totalcredit = totalcredit + subject
  temp=0.0;
  for(int i=0;i<8;i++)
  temp = temp + (subject[i].grade *
  sgpa =(double)(temp/totalcredi
  System.out.println("SGPA of
  class Student
  {public static void main(Strin
  { student1 stud = new studen
    stud.getstudentdetails();
    stud.getmarks();
    stud.calsgpa();
  }}

OUTPUT:
Enter name and usn of
K Ashwarya
129
Enter marks and credits
Enter marks and credits of
Enter marks and credits of
Enter marks and credits
Enter marks and credits
Enter marks and credits
Enter marks and credits
Enter marks and credit
Enter marks and credit
```

Code:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
}

class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subjects;
    Scanner sc;


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

    void getStudentDetails() {
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter USN: ");
        usn = sc.nextLine();
    }

    void getMarks() {
        for (int i = 0; i < 9; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            subjects[i].subjectMarks = sc.nextInt();
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            subjects[i].credits = sc.nextInt();
        }
    }

    void computeSGPA() {
        double totalGradePoints = 0;
        int totalCredits = 0;

        for (int i = 0; i < 9; i++) {
            int grade = subjects[i].subjectMarks / 10; // Assuming grade is based on marks
            totalGradePoints += grade * subjects[i].credits;
```

```
            totalCredits += subjects[i].credits;
        }

        if (totalCredits > 0) {
            SGPA = totalGradePoints / totalCredits;
        } else {
            SGPA = 0;
        }
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }

    public static void main(String[] args) {
        Student student = new Student();
        student.getStudentDetails();
        student.getMarks();
        student.computeSGPA();
        student.display();
    System.out.println("Name:K.AISHWARYA, Usn:1BM23CS129");
    }
}
```

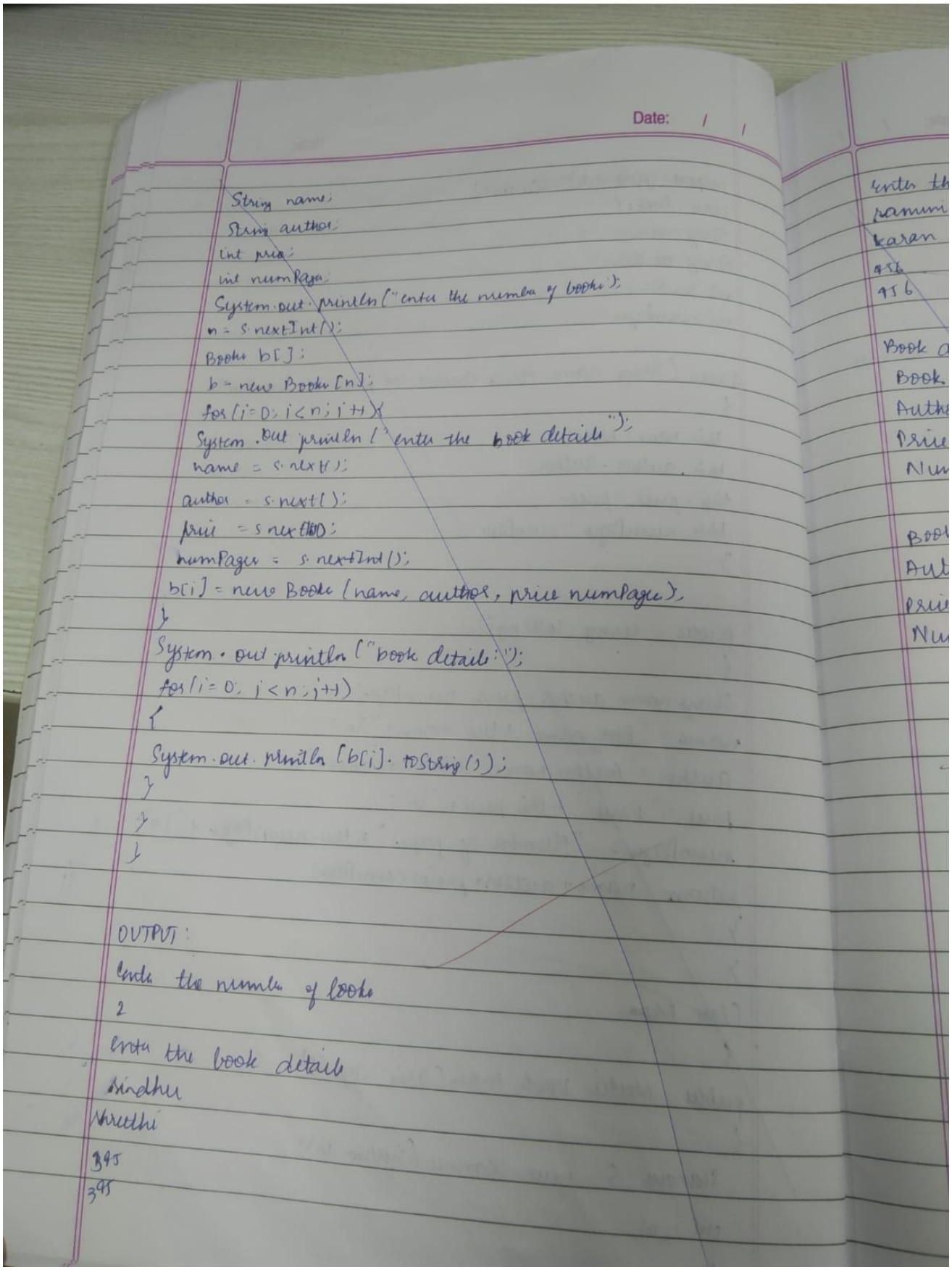


Program 3
Implement Books Equation

Algorithm:

Date: / /

```
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 i, n;
```



Date: / /

```
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 book details");
name = s.next();
author = s.next();
price = s.nextDouble();
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].toString());
}
}
}
```

OUTPUT:

```
Enter the number of books
2
enter the book details
Sindhu
Murthi
345
345
```

te: / /

Date: / /

Write the book details
Rammi
Karan
496
496

Book details:
Book name: Sindhu
Author name: Shruthi
Price: 345
Number of pages: 345

Book name: Rammi
Author name: Karan
Price: 496
Number of pages: 496.

08/11/24

It is out cut by mistake Ashwajeevika
20/11/24

Code:

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

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

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

      for (int i = 0; i < n; i++) {
         System.out.println("Enter the book details (name, author, price, number of pages):");
         String name = s.next();
         String author = s.next();
         int price = s.nextInt();
         int numPages = s.nextInt();
         books[i] = new Books(name, author, price, numPages);
      }

      System.out.println("Book details:");
      for (Books book : books) {
         System.out.println(book.toString());
         System.out.println("Name:K.Aishwarya,Usn:1BM23CS129");
```

```
        }

        s.close();
    }
}
```

```
D:\1BM23CS129>javac Main.java

D:\1BM23CS129>java Main
Enter the number of books:
2
Enter the book details (name, author, price, number of pages):
Sindhu
Shruthi
345
345
Enter the book details (name, author, price, number of pages):
Rammi
Karan
456
456
Book details:
Book name: Sindhu
Author name: Shruthi
Price: 345
Number of pages: 345

Name:K.Aishwarya,Usn:1BM23CS129
Book name: Rammi
Author name: Karan
Price: 456
Number of pages: 456

Name:K.Aishwarya,Usn:1BM23CS129

D:\1BM23CS129>
```

Program 4
Implement Area Equation

Algorithm:

Lab Program no 6

Date: / /

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;
public class StudentMarks {
protected String usn;
protected String name;
protected int sem;

public StudentMarks (String usn, String name, int sem) {
this.usn = usn;
this.name = name;
this.sem = sem;
}
public void displayStudentDetails () {
System.out.println ("USN:" + usn);
System.out.println ("Name:" + name);
System.out.println ("Semester:" + sem);
}
}

class Internals extends StudentMarks {
protected int[] internalMarks = new int[5];
public Internals (String usn, String name, int sem, int[]
```

21-10-2024 Lab Program No 4

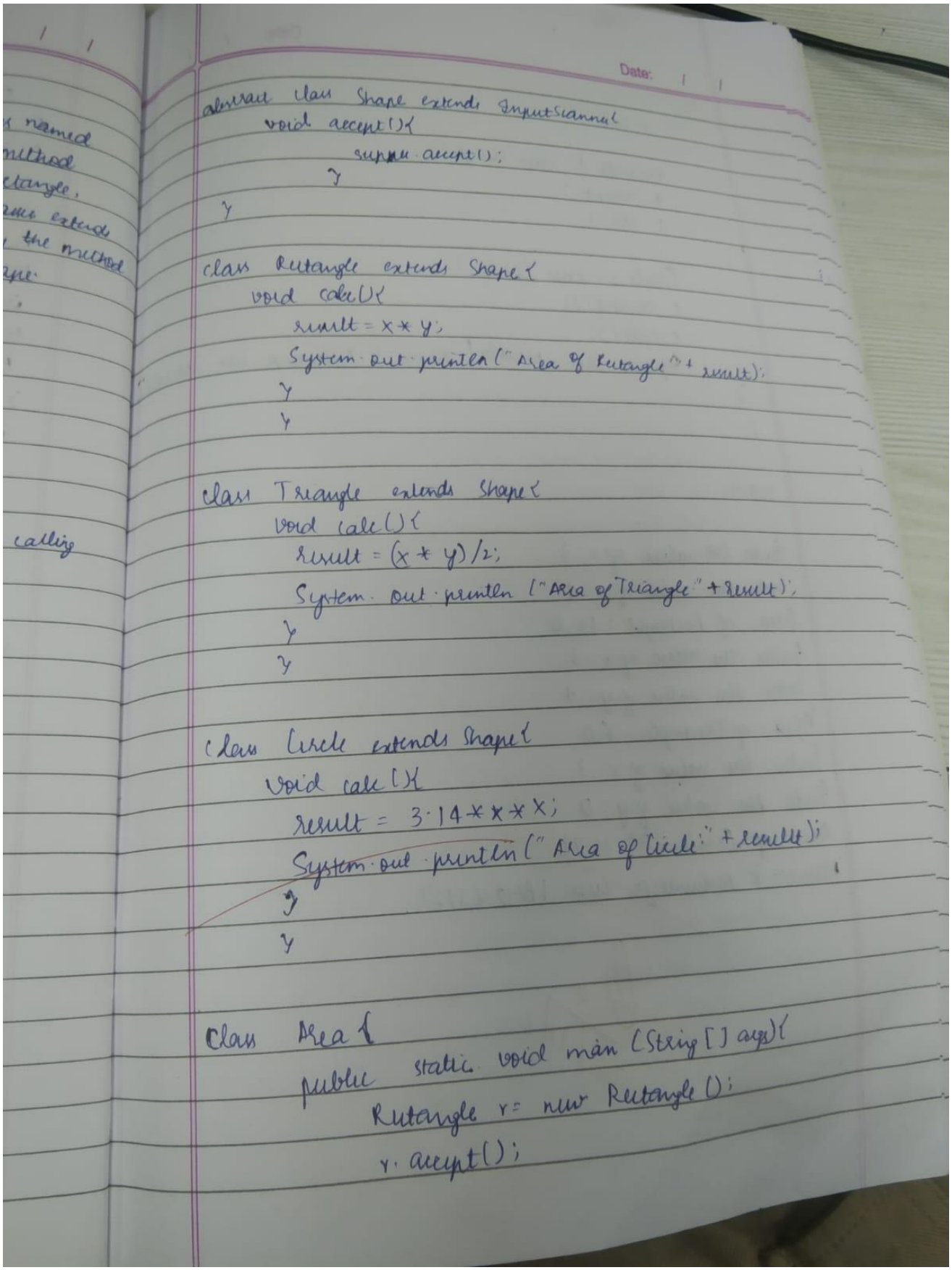
- Develop a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the class contain only the method printArea()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

Steps

1. Add class InputScanner
2. Add class Shape extend InputScanner
3. Add class Rectangle extends Shape
4. Add class Triangle extends Shape
5. Add class Circle extends Shape
6. Add main class invoking main method with objects and calling input method and printarea method.

Program:

```
import java.util.Scanner;
abstract class InputScanner{
  double x, y, result;
  void accept(){
  Scanner sc = new Scanner(System.in);
  System.out.print("Enter the value of x:");
  x = sc.nextDouble();
  System.out.print("Enter the value of y:");
  y = sc.nextDouble();
  }
abstract void calc();
}
```



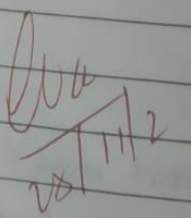
```
abstract class Shape extends InputScanner{
    void accept(){
        super.accept();
    }
}

class Rectangle extends Shape{
    void calc(){
        result = x * y;
        System.out.println("Area of Rectangle" + result);
    }
}

class Triangle extends Shape{
    void calc(){
        result = (x * y)/2;
        System.out.println("Area of Triangle" + result);
    }
}

class Circle extends Shape{
    void calc(){
        result = 3.14 * x * x;
        System.out.println("Area of Circle:" + result);
    }
}

Class Area{
    public static void main(String[] args){
        Rectangle r = new Rectangle();
        r.accept();
```

Date: / /

```
r.calc();

Triangle t = new Triangle();
t.accept();
t.calc();

Circle c = new Circle();
c.accept();
c.calc();
System.out.println("Name: K.Ashwanya, usn: 1BM23CS129");
}
}
```

OUTPUT:

```
Enter the value of x: 3
Enter the value of y: 4
Area of Rectangle: 12.0
Enter the value of x: 3
Enter the value of y: 4
Area of Triangle: 6.0
Enter the value of x: 3
Enter the value of y: 0
Area of Circle: 28.2599
Name: K.Ashwanya usn: 1BM23CS129
```



Code:

```
import java.util.Scanner;

class InputScanner {

    public int getInput() {
        Scanner scanner = new Scanner(System.in);
        return scanner.nextInt();
    }
}

abstract class Shape extends InputScanner {

    int dimension1, dimension2;


    public abstract void printArea();


    public void inputDimensions() {
        System.out.println("Enter the first dimension (e.g., length, radius, base):");
        dimension1 = getInput();
        System.out.println("Enter the second dimension (e.g., width, height, 0 for Circle):");
        dimension2 = getInput();
    }
}

class Rectangle extends Shape {

    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}


class Triangle extends Shape {

    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {

    public void printArea() {
```

```
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        Shape rectangle = new Rectangle();
        Shape triangle = new Triangle();
        Shape circle = new Circle();


        System.out.println("Enter dimensions for Rectangle:");
        rectangle.inputDimensions();
        rectangle.printArea();


        System.out.println("Enter dimensions for Triangle:");
        triangle.inputDimensions();
        triangle.printArea();


        System.out.println("Enter dimensions for Circle:");
        circle.inputDimensions();
        circle.printArea();
    }
}
```

```
D:\1BM23CS129>java Figure
Enter dimensions for Rectangle:
Enter the first dimension (e.g., length, radius, base):
20
Enter the second dimension (e.g., width, height, 0 for Circle):
40
Area of Rectangle: 800
Enter dimensions for Triangle:
Enter the first dimension (e.g., length, radius, base):
20
Enter the second dimension (e.g., width, height, 0 for Circle):
40
Area of Triangle: 400.0
Enter dimensions for Circle:
Enter the first dimension (e.g., length, radius, base):
20
Enter the second dimension (e.g., width, height, 0 for Circle):
40
Area of Circle: 1256.6370614359173

D:\1BM23CS129>
```

Program 5
Implement Bank Equation

Algorithm:

LAB Program no 5

- Develop a Java Program to create a class Bank that maintains two kind of account for its customer, 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 store 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 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
  e) Check for the minimum balance, impose penalty if necessary and update the balance.

```
Class Account {
String CustomerName;
int AccountNumber;
double balance;
String accountType;

public Account(String customerName, int accountNumber, String
  accountType){
This.CustomerName = CustomerName;
this.accountNumber = accountNumber;
this.accountType = accountType;
this.balance = 0.0
}
```

Date: / /

```
public void deposit (double amount){
    if (amount>0){
        balance+= amount;
        System.out.println("Deposited "+amount);
    }else{
        System.out.println("Deposit amount should be positive");
    }}
public void withdraw(double amount){
    if(amount > 0 && balance >= amount){
        balance -= amount;
        System.out.println("withdraw "+amount);
    }else{
        System.out.println("Insufficient balance or invalid amount");
    }}
public void displayBalance(){
    System.out.println("Balance "+balance);}}
class SavAct extends Account{
    double interestRate;}
public void computeAndDepositInterest(){
    double interest = balance * Math.pow(1+interestRate/100,1) -
    balance;
    balance += interest;
    System.out.println("Interest Deposited:"+interest);}}
class CurAct extends Account{
    double minimumBalance;
    double serviceCharge;
public CurAct(String customerName, int accountNumber, double
minimumBalance, double serviceCharge){
super(customerName, accountNumber, "current");
    this.minimumBalance = minimumBalance;
```

Date: / /

```
        this.serviceCharge = serviceCharge;
    }
    public void ChkAndImposeServiceCharge(){
        if (balance < minimumBalance){
            double charge = serviceCharge;
            balance -= charge;
            System.out.println("Service Charge Imposed: "+charge);}}}
class Bank{
public static void main (String[] args){
    SavAcct savingAccount = new SavAcct ("John", 1001, 5.0);
    savingsAccount.deposit(1000);
    savingsAccount.computeAndDepositInterest();
    savingsAccount.displayBalance();
    savingsAccount.withdraw(200);
    savingsAccount.displayBalance();

    CurAcct currentAccount = new CurAcct("Ram", 2001, 500, 50.0);
    currentAccount.deposit(300);
    currentAccount.ChkAndImposeServiceCharge();
    currentAccount.displayBalance();
    currentAccount.withdraw(100);
    currentAccount.displayBalance();
    currentAccount.ChkAndImposeServiceCharge();
    currentAccount.displayBalance();
}
}
```

OUTPUT:

Deposited: 1000.0
Interest Deposited: 50.0
Balance: 1050.0
Withdraw: 200.0
Balance: 850.0
Deposited: 300.0
Service Charge Imposed: 0.0
Balance: 450.0
Withdraw: 100.0
Balance: 150.0
Service Charge Imposed: 50.0
Balance: 100.0

Code:

```
class Account {
    protected String customerName;
    protected int accountNumber;
    protected double balance;
    protected String accountType;


    public Account(String customerName, int accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }


    public void deposit(double amount) {
        if(amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount should be positive!");
        }
    }


    public void withdraw(double amount) {
        if(amount > 0 && balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance or invalid amount!");
        }
    }


    public void displayBalance() {
        System.out.println("Balance: " + balance);
    }
}

class SavAcct extends Account {
    private double interestRate;


    public SavAcct(String customerName, int accountNumber, double interestRate) {
        super(customerName, accountNumber, "Savings");
```

```
        this.interestRate = interestRate;
    }


    public void computeAndDepositInterest() {
        double interest = balance * Math.pow((1 + interestRate / 100), 1) - balance;
        balance += interest;
        System.out.println("Interest Deposited: " + interest);
    }
}

class CurAcct extends Account {
    private double minimumBalance;
    private double serviceCharge;


    public CurAcct(String customerName, int accountNumber, double minimumBalance, double
serviceCharge) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }


    public void checkAndImposeServiceCharge() {
        if(balance < minimumBalance) {
            double charge = serviceCharge;
            balance -= charge;
            System.out.println("Service Charge Imposed: " + charge);
        }
    }
}

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

        SavAcct savingsAccount = new SavAcct("John ", 1001, 5.0);
        savingsAccount.deposit(1000);
        savingsAccount.computeAndDepositInterest();
        savingsAccount.displayBalance();
        savingsAccount.withdraw(200);
        savingsAccount.displayBalance();


        CurAcct currentAccount = new CurAcct("Ram", 2001, 500, 50.0);
        currentAccount.deposit(300);
        currentAccount.checkAndImposeServiceCharge();
```

```
        currentAccount.displayBalance();
        currentAccount.withdraw(100);
        currentAccount.displayBalance();
        currentAccount.checkAndImposeServiceCharge();
        currentAccount.displayBalance();
    }
}
```

```
D:\1BM23CS129>javac Bank.java

D:\1BM23CS129>java Bank
Deposited: 1000.0
Interest Deposited: 50.0
Balance: 1050.0
Withdrawn: 200.0
Balance: 850.0
Deposited: 300.0
Service Charge Imposed: 50.0
Balance: 250.0
Withdrawn: 100.0
Balance: 150.0
Service Charge Imposed: 50.0
Balance: 100.0

D:\1BM23CS129>
```

Program 6:
Implement Package Equation

Algorithm:

Lab Program no 6

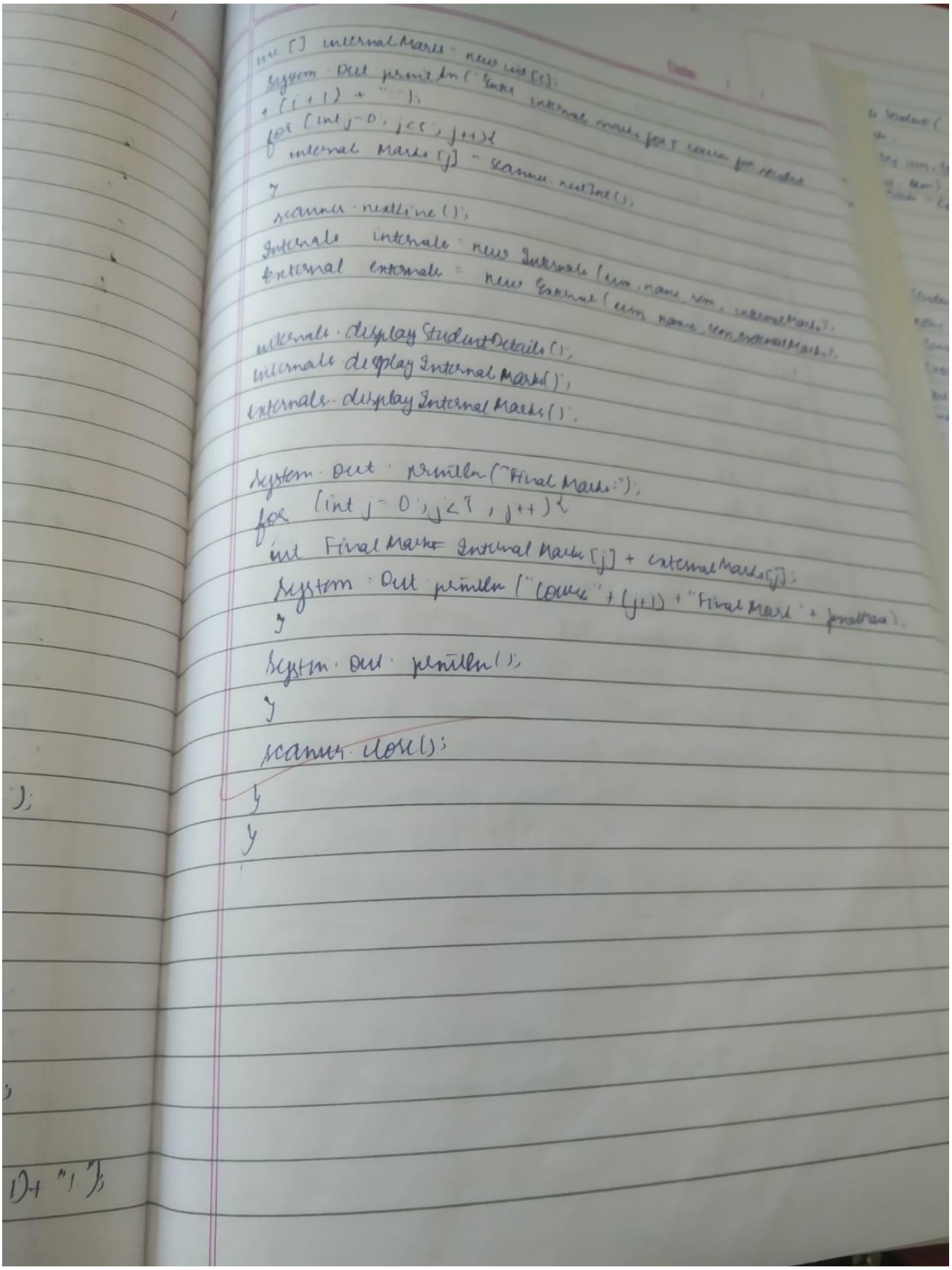
Date: / /

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.

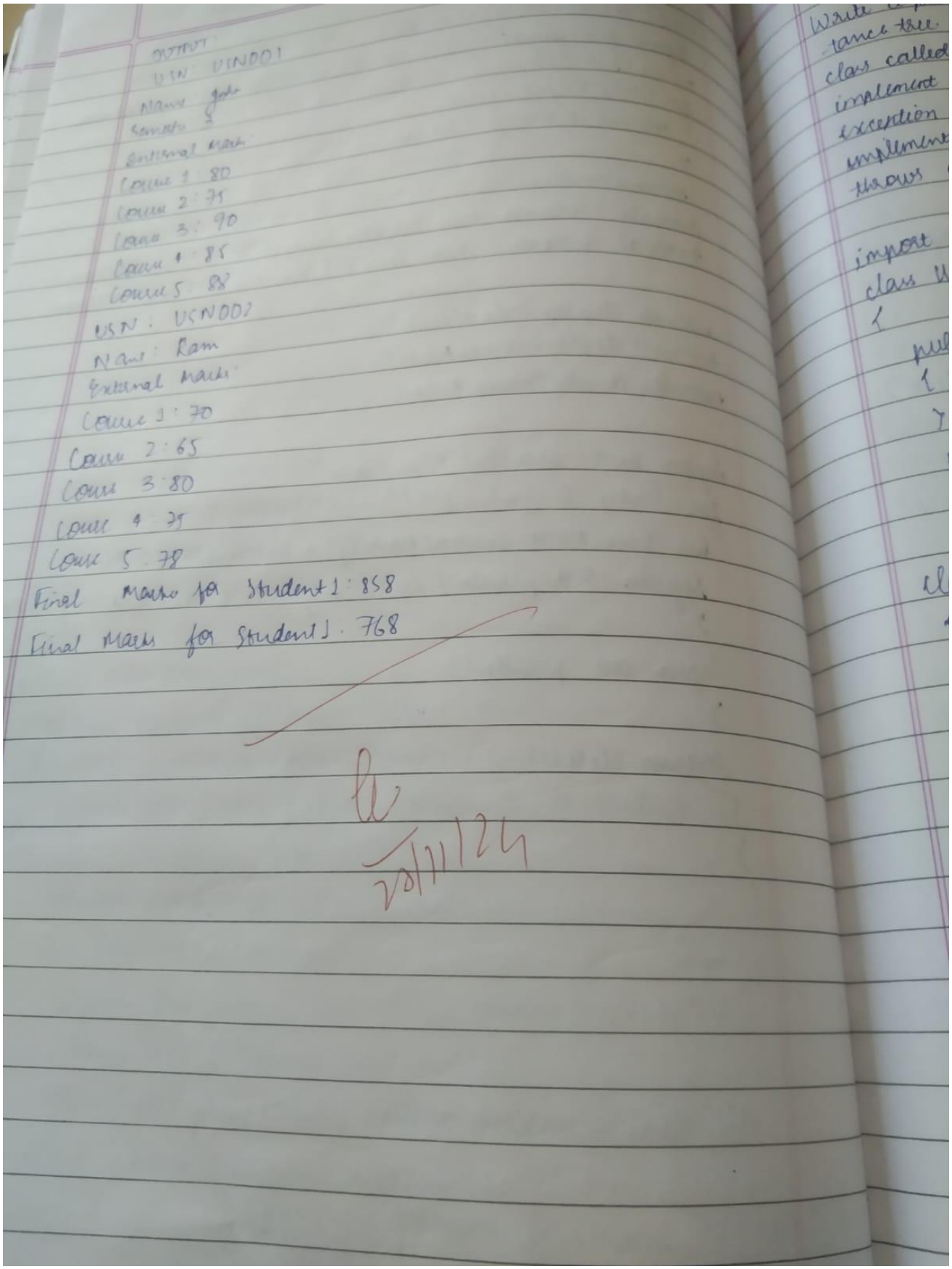
```
package CIE;
public class StudentMarks {
protected String usn;
protected String name;
protected int sem;

public StudentMarks (String usn, String name, int sem) {
this.usn = usn;
this.name = name;
this.sem = sem;
}
public void displayStudentDetails () {
System.out.println ("USN:" + usn);
System.out.println ("Name:" + name);
System.out.println ("Semester:" + sem);
}
}

class Internals extends StudentMarks {
protected int[] internalMarks = new int[5];
public Internals (String usn, String name, int sem, int[]
```

```
internalMarks){
super(usn, name, sem);
this.internalMarks = internalMarks;
}

public void displayInternalMarks(){
System.out.println("Internal Marks:");
for(int mark : internalMarks){
System.out.print(mark + " ");
}
System.out.println();
}
}
```

```
package SEE;
import CIE.StudentMarks;
import SEE.External;
import java.util.Scanner;

public class FinalMarks{
public static void main(String[] args){
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of students:");
int n = scanner.nextInt();
scanner.nextLine();
for(int i=0; i<n; i++){
System.out.print("Enter USN for student "+(i+1)+":");
String usn = scanner.nextLine();
System.out.print("Enter name of Student "+(i+1)+":");
String name = scanner.nextLine();
System.out.print("Enter Semester for student "+(i+1)+":");
int sem = scanner.nextInt();
```



```
int [] internalMarks = new int[5];
System.out.println("Enter internal marks for 5 courses for student
+ (i+1) + ":");
for (int j=0; j<5; j++){
  internalMarks[j] = scanner.nextInt();
}
scanner.nextLine();
Internals internals = new Internals(usn, name, sem, internalMarks);
External externals = new External(usn, name, sem, externalMarks);

internals.displayStudentDetails();
internals.displayInternalMarks();
externals.displayExternalMarks();

System.out.println("Final Marks:");
for (int j=0; j<5; j++){
int FinalMarks = InternalMarks[j] + externalMarks[j];
System.out.println("Course" + (j+1) + "Final Mark" + finalMarks);
}
System.out.println();
}
scanner.close();
}
}
```


OUTPUT
USN: 1VIND01
Name: John
Semester: 5
Enter internal marks:
Course 1: 80
Course 2: 71
Course 3: 90
Course 4: 85
Course 5: 88
USN: 1VIND02
Name: Ram
Enter internal marks:
Course 1: 70
Course 2: 65
Course 3: 80
Course 4: 71
Course 5: 78
Final marks for Student 1: 858
Final marks for Student 1: 768
20/11/24

Code: import CIE.*;
import SEE.Externals;
import java.util.Scanner;

```
public class Final {
    public static void main(String args[]) {
        int n;
        Scanner s1 = new Scanner(System.in);

        System.out.println("Enter number of students:");
        n = s1.nextInt();

        Externals[] eobj = new Externals[n];

        for (int i = 0; i < n; i++) {
            eobj[i] = new Externals();

            System.out.println("Enter name, USN, semester of student:");
            String name = s1.next();
            int usn = s1.nextInt();
            int sem = s1.nextInt();

            eobj[i].getdetails(name, usn, sem);
            eobj[i].displaystudent();
            eobj[i].getimarks();
            eobj[i].getsmarks();
            eobj[i].calfinal();
            eobj[i].displayfmarks();
        }
    }
}
```

```
D:\1BM23CS129>java Final
Enter number of students:
4
Enter name, USN, semester of student:
Aish 12 3
Student name: Aish
USN: 12
Semester: 3
Enter CIE marks for 5 subjects:
50
49
45
45
45
Enter SEE marks for 5 subjects:
100
98
90
90
90
Final marks are:
100
98
90
90
90
Enter name, USN, semester of student:
```

Program 10A
PCFixed

Algorithm:

Program : PCA [Method]

Date: / /

```
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("InterruptedException caught");
}
this.n=n;
valueSet = true;
System.out.println("Put: "+n);
System.out.println("\n Intimate Consumer\n");
notify();
}
```

Date: / /

System.out.println("press control-C to stop.");
}
}

Press Control-C to stop.
Put=0
Intimate Consumer
Producer waiting
Got=0
Put=1
Intimate Consumer
Producer waiting
Got=1
Put=2
Intimate Consumer
Producer Waiting
Got=2
Put=3
Intimate Consumer
Producer Waiting
Got=3
Put=4
Intimate Consumer
Producer Waiting
Got=4
Put=5
Intimate Consumer
Producer Waiting
Got=5
Put=6
Intimate Consumer

LAB program no: 7 [Exception]

Date: / /

Write a program that demonstrates handling of exception 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 takes 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()
  { System.out.println("Age Error");
  }
  public WrongAge(String message)
  { System.out.println(message);
  }
class Father
{ protected int fage;
  public Father() throws WrongAge
  { Scanner scanner = new Scanner(System.in);
  System.out.print("Enter Father's age: ");
  fage = scanner.nextInt();
  if (fage <0)
    { throw new WrongAge("Father's age cannot be negative");
    }
  }
  public void display()
  { System.out.println("Father's Age: "+fage);
  }
}
```

```
Class Son extends Father {
private int sonage;
public Son() throws WrongAge
{
Scanner scanner = new Scanner(System.in);
System.out.print("Enter Son's age:");
sonage = scanner.nextInt();
if (sonage >= fage)
{ throw new WrongAge("Son's age cannot be greater than OR
  equal to Father's age");
}
elsif (sonage < 0)
{ throw new WrongAge("Son's age cannot be negative");
}
}
public void display()
{ super.display();
  System.out.println("Son's Age: " + sonage);
}
}
public class Demo
{ public static void main(String args[])
{ System.out.println("K Aishwarya 1BM23CS129");
try
{ Son s = new Son();
  s.display();
}
catch (WrongAge e)
{ System.out.println("Exception caught"); }
}
}
```

OUTPUT

K Aishwa
Enter
Enter
Father
Son's

K. Ais
Ente
Ente
Son's
Ex

Ente
Fat
Ex

Code:

```
import java.util.Scanner;
class WrongAge extends Exception
{
   public WrongAge()
 {
      System.out.println("Age Error");
 }

   public WrongAge(String message)
 {
      System.out.println(message);
 }
}
class Father
 {
   protected int fage;
   public Father() throws WrongAge
   {
   Scanner scanner = new Scanner(System.in);
   System.out.print("Enter Father's age: ");
   fage = scanner.nextInt();
   if (fage < 0)
      {
         throw new WrongAge("Father's age cannot be negative");
      }
   }
   public void display()
   {
      System.out.println("Father's Age: " + fage);
   }
}
class Son extends Father{
   private int sonage;
   public Son() throws WrongAge
 {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter Son's age: ");
      sonage = scanner.nextInt();
      if (sonage >= fage)
      {
         throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
      }
     else if (sonage < 0)
      {
```

```
            throw new WrongAge("Son's age cannot be negative");
        }
    }
    public void display()
    {
        super.display();
        System.out.println("Son's Age: " + sonage);
    }
}
public class Demoa
 {
    public static void main(String args[])
     {
        System.out.println("K Aishwarya 1BM23CS129");
        try
        {
            Son s=new Son();

            s.display();
        }
        catch (WrongAge e)
        {
            System.out.println("Exception Caught");
        }
    }
}
```

```
D:\1BM23CS129>java Demoa
K Aishwarya 1BM23CS129
Enter Father's age: 90
Enter Son's age: 70
Father's Age: 90
Son's Age: 70

D:\1BM23CS129>java Demoa
K Aishwarya 1BM23CS129
Enter Father's age: 40
Enter Son's age: 50
Son's age cannot be greater than or equal to Father's age
Exception Caught

D:\1BM23CS129>java Demoa
K Aishwarya 1BM23CS129
Enter Father's age: -1
Father's age cannot be negative
Exception Caught

D:\1BM23CS129>java Demoa
K Aishwarya 1BM23CS129
Enter Father's age: 80
Enter Son's age: -2
Son's age cannot be negative
Exception Caught

D:\1BM23CS129>
```

Program 8
Multithreading

Algorithm:

Lab program no 8 (Multithreading)

Date: / /

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 BMS extends Thread
{  public void run()
   {  try
      {  for(int i=0;i<5;i++){
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         }
      }
      Catch(InterruptedException e){
         System.out.println("Thread Interrupted:"+e.getMessage());
      }
   }
}

class CSE extends Thread
{  public void run()
   {  try {  for(int i=0;i<5;i++){
            System.out.println("CSE");
            Thread.sleep(2000);
         } }
      Catch (InterruptedException e){
         System.out.println("Thread interrupted:"+e);
      }
   }
}
```

Lab program no 8 (Multithreading)

Date: / /

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 BMS extends Thread
{  public void run()
   {  try
      {  for(int i=0;i<5;i++){
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         }
      }
      Catch(InterruptedException e){
         System.out.println("Thread Interrupted:"+e.getMessage());
      }
   }
}

class CSE extends Thread
{  public void run()
   {  try {  for(int i=0;i<5;i++){
            System.out.println("CSE");
            Thread.sleep(2000);
         } }
      Catch (InterruptedException e){
         System.out.println("Thread interrupted:"+e);
      }
   }
}
```

Code:

```
class BMS extends Thread
{
   public void run()
{
       try{
          for (int i = 0; i < 5; i++) {
             System.out.println("BMS College of Engineering");
             Thread.sleep(10000);
          }
       } catch (InterruptedException e) {
          System.out.println("Thread interrupted: " + e.getMessage());
       }
   }
}

class CSE extends Thread
 {
   public void run()
 {
       try{
          for (int i = 0; i < 5; i++){
             System.out.println("CSE");
             Thread.sleep(2000);
          }
       }
      catch (InterruptedException e){
          System.out.println("Thread interrupted: " + e);
       }
   }
}

public class ThreadExample{
   public static void main(String[] args){
      System.out.println("Aishwarya K");
      BMS bms=new BMS();
      CSE cse=new CSE();
       bms.start();
       cse.start();
   }
}
```

```
D:\1BM23CS129>java ThreadExample
Aishwarya K
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering

D:\1BM23CS129>
```

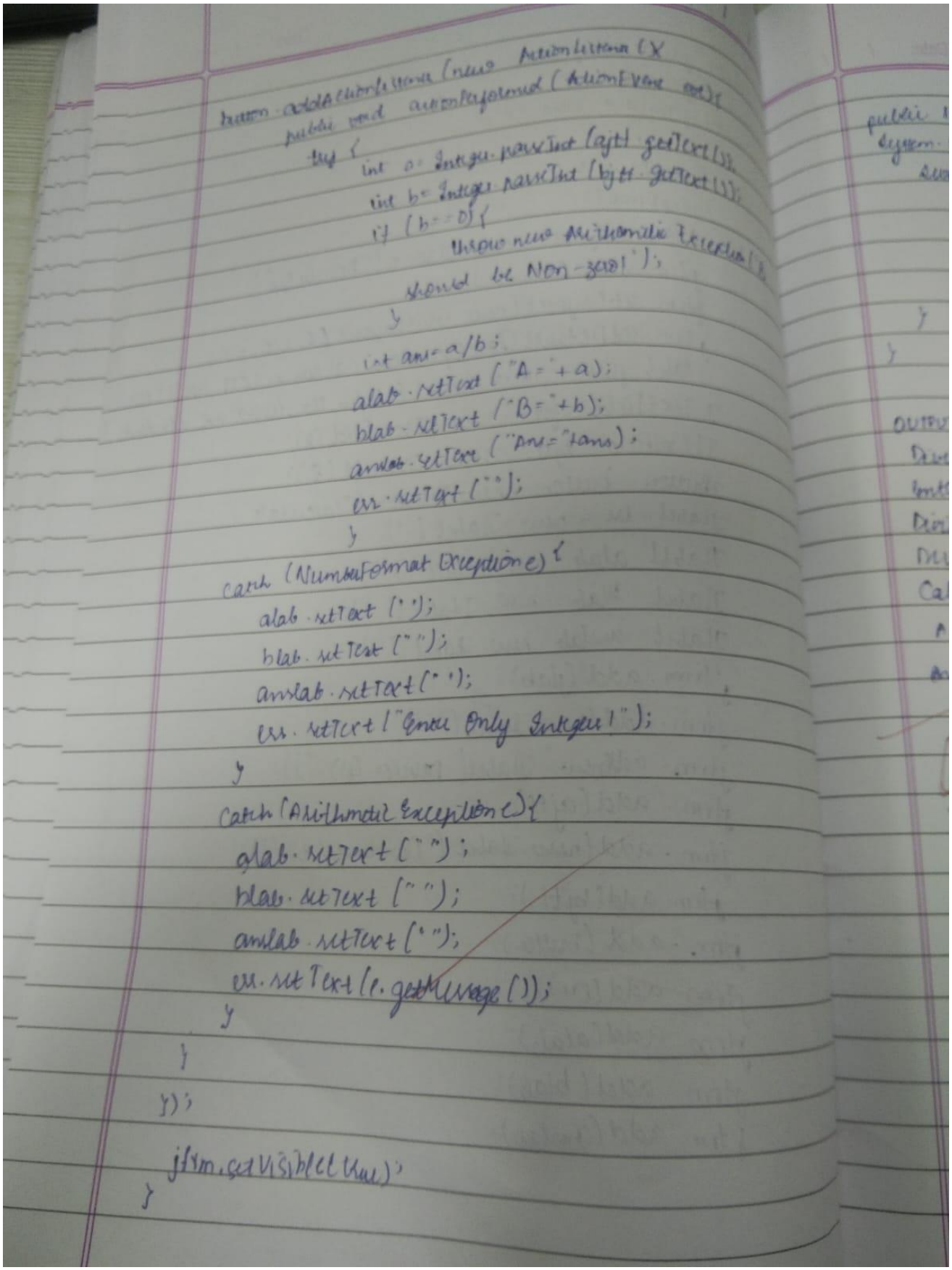
Program 9:
Interface

Algorithm:

Program 09 [Interface]

Date: / /

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

Class SwingDemo {
    SwingDemo() {
      JFrame jfram = new JFrame ("Divider App");
      jfrm.setSize(300, 200);
      jfrm.setLayout(new GridLayout(6, 2, 5, 5));
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      JLabel jlab = new JLabel("Enter the divisor and dividend");
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);
      JButton button = new JButton("Calculate");
      JLabel err = new JLabel("");
      JLabll alab = new JLabel("");
      JLabel blab = new JLabel("");
      JLabel anslab = new Label("");
      jfrm.add(jlab);
      jfrm.add(new Label(""));
      jfrm.addnew JLabel("Divisor (A):"));
      jfrm.add(ajtf);
      jfrm.add(new JLabel("Dividend (B):"));
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(err);
      jfrm.add(alab);
      jfrm.add(blab);
      jfrm.add(anslab);
```



```
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e){
try {
int a= Integer.parseInt(ajtf.getText());
int b= Integer.parseInt(bjtf.getText());
if (b==0){
throw new ArithmeticException("
Should be Non-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(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText(e.getMessage());
}
}
});
jfrm.setVisible(true);
}
```

Date: / /

```
public static void main(String args[]){
System.out.println("K Ashwarya");
SwingUtilities.invokeLater(new Runnable(){
    public void run(){
        new SwingDemo();
    }
});
}
}
```

OUTPUT:

Divider App:

Enter the divisor and dividend.

Divisor (A): 3

Dividend (B): 3

Calculate

A = 3, B = 3

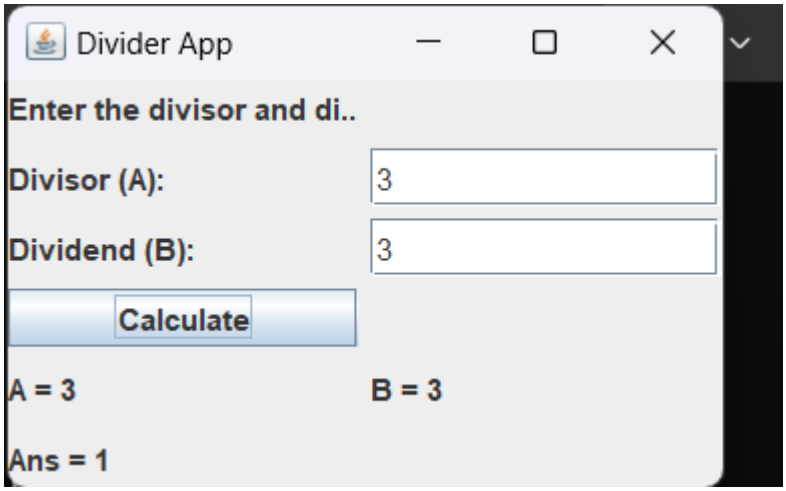
Ans = 1

Code:

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

class SwingDemo {
   SwingDemo() {
      // Create jframe container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(300, 200);
      jfrm.setLayout(new GridLayout(6, 2, 5, 5)); // Using GridLayout for better arrangement

      // To terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // Text label
      JLabel jlab = new JLabel("Enter the divisor and dividend:");

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

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

      // Labels for displaying results and error
      JLabel err = new JLabel("");
      JLabel alab = new JLabel("");  // A label for the divisor
      JLabel blab = new JLabel("");  // B label for the dividend
      JLabel anslab = new JLabel(""); // Answer label

      // Add components in a structured manner using GridLayout
      jfrm.add(jlab);
      jfrm.add(new JLabel(""));  // Empty label for layout spacing

      jfrm.add(new JLabel("Divisor (A):"));
      jfrm.add(ajtf);

      jfrm.add(new JLabel("Dividend (B):"));
      jfrm.add(bjtf);

      jfrm.add(button);
      jfrm.add(err);  // Error message label

      jfrm.add(alab);  // A value label
      jfrm.add(blab);  // B value label
```

```
        jfrm.add(anslab);  // Answer label

        // Add action listener to the Calculate button
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    int a = Integer.parseInt(ajtf.getText());  // Read divisor
                    int b = Integer.parseInt(bjtf.getText());  // Read dividend

                    if (b == 0) { // Handle division by zero
                        throw new ArithmeticException("B should be NON-zero!");
                    }

                    int ans = a / b;  // Perform division
                    alab.setText("A = " + a);  // Display A value
                    blab.setText("B = " + b);  // Display B value
                    anslab.setText("Ans = " + ans);  // Display result
                    err.setText("");  // Clear error if valid input

                } catch (NumberFormatException e) {
                    // Handle invalid integer input
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmeticException e) {
                    // Handle division by zero
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText(e.getMessage());
                }
            }
        });

        // Display frame
        jfrm.setVisible(true);
    }

    public static void main(String args[]) {
System.out.println("K Aishwarya");
        // Create frame on the event dispatching thread
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
```

```
    }
}
```


Divider App
Enter the divisor and di..
Divisor (A):
3
Dividend (B):
3
Calculate
A = 3
B = 3
Ans = 1

Program 10A:
Interface

Algorithm:

Program : PCA [Method]

Date: / /

```
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("InterruptedException caught");
}
this.n=n;
valueSet = true;
System.out.println("Put: "+n);
System.out.println("\n Intimate Consumer\n");
notify();
}
```

Date: / /

```
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[]) {
System.out.println ("Keshavaya");
Q q = new Q();
new Producer (q);
new Consumer (q);
```

Date: / /

System.out.println("press control-C to stop.");
}
}

Press Control-C to stop.
Put=0
Intimate Consumer
Producer waiting
Got=0
Put=1
Intimate Consumer
Producer waiting
Got=1
Put=2
Intimate Consumer
Producer Waiting
Got=2
Put=3
Intimate Consumer
Producer Waiting
Got=3
Put=4
Intimate Consumer
Producer Waiting
Got=4
Put=5
Intimate Consumer
Producer Waiting
Got=5
Put=6
Intimate Consumer

Date: / /

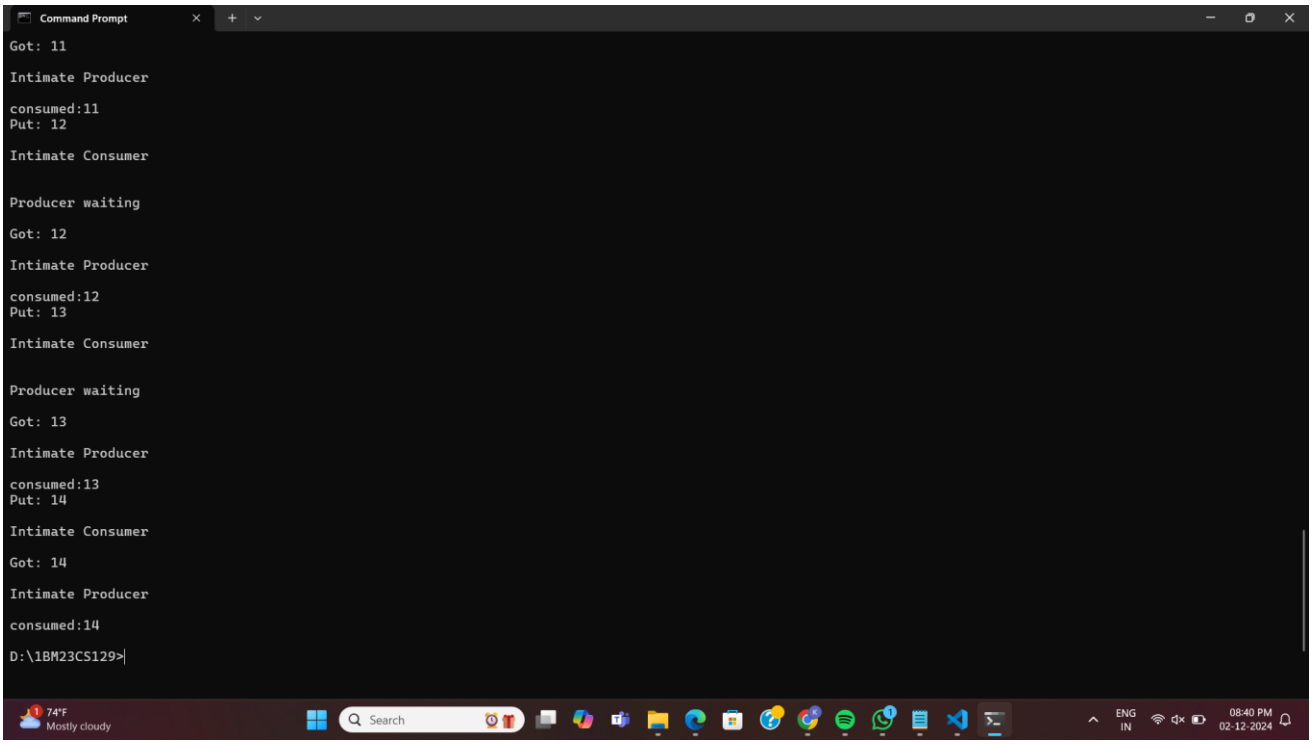
Producer Waiting
Got = 6
Put = 7
Initimate consumer
Producer Waiting
Got = 7
Put = 8
Initimate Consumer
Producer Waiting
Got = 8
Put = 9
Initimate Consumer
Producer Waiting
Got = 9
Put = 10
Initimate Consumer
Producer Waiting
Got = 10
Initimate Producer
consumed = 10
Put = 11
Initmate producer
consumed = 11
Put = 12
Initimate Producer
Consumed = 12
Put = 13
Initimate Producer
consumed = 13
Put = 14
Initimate Producer
Consumed = 14

Code:

```
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[]) {
System.out.println("K Aishwarya");

Q q = new Q();

new Producer(q);

new Consumer(q);

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

}

}
```


Command Prompt
Got: 11
Intimate Producer
consumed:11
Put: 12
Intimate Consumer
Producer waiting
Got: 12
Intimate Producer
consumed:12
Put: 13
Intimate Consumer
Producer waiting
Got: 13
Intimate Producer
consumed:13
Put: 14
Intimate Consumer
Got: 14
Intimate Producer
consumed:14
D:\1BM23CS129>
74°F
Mostly cloudy
Search
ENG
IN
08:40 PM
02-12-2024

Program 10B:
Deadlock

Algorithm:

Date: / /

Program 20B [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");
}
}
```

Date: / /

```
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[]){
    System.out.println("Aishwarya");
  new Deadlock();
  }
  }
```

Date: / /

Aishwarya

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

28-11-2024

Code:

```
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[]) {
System.out.println("K Aishwarya");

new Deadlock();
```

```
}

}
```

```
D:\1BM23CS129>javac Deadlock.java

D:\1BM23CS129>java Deadlock
K Aishwarya
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread

D:\1BM23CS129>
```