VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Keerthana H Bhat (1BM23CS148)

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 Keerthana H Bhat**(1BM23CS148),** who is a 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.

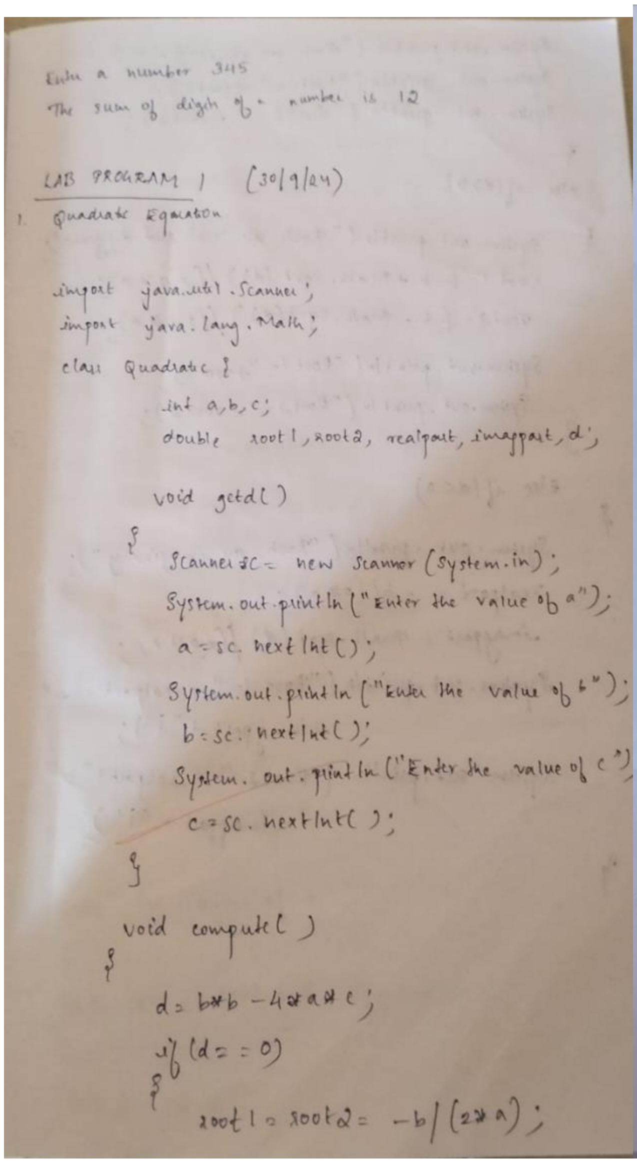
Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link: https://github.com/keerthanaa33/OOJ-LAB-PROGRAM

<u>Program 1</u>
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.

Algorithm:

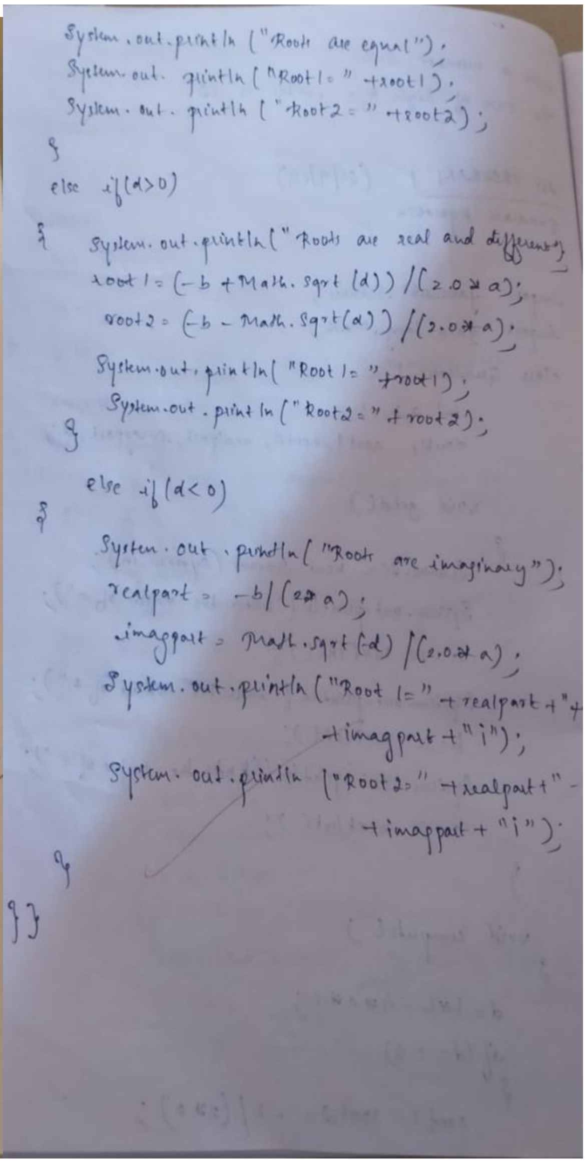


Enter a number 345
The sum of digits of number is 12

LAB PROGRAM 1 (30/9/24)

1. Quadratic Equation

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic {
        int a,b,c;
        double root1,root2, realpart, imagpart, d;

        void getd()
      {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter the value of a");
         a=sc.nextInt();
        System.out.println("Enter the value of b");
         b=sc.nextInt();
        System.out.println("Enter the value of c");
         c=sc.nextInt();
      }
    void compute( )
   {
      d= b*b -4*a*c;
      if (d= =0)
      {
         root1 = root2 = -b/(2*a);
```

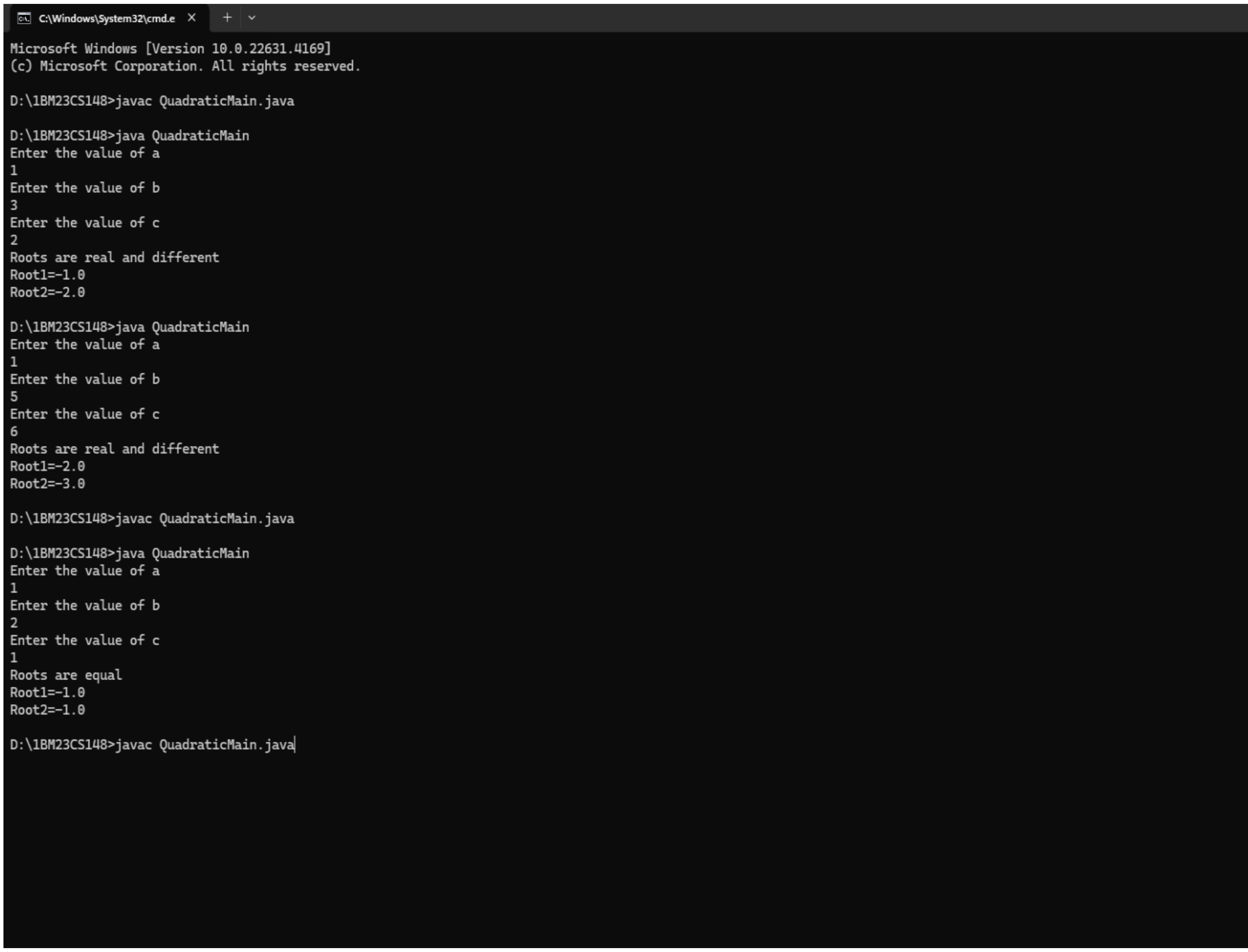


```
         System.out.println("Roots are equal");
         System.out.println("Root1=" +root1);
         System.out.println("Root2=" +root2);
      }
      else if(d>0)
      {
         System.out.println("Roots are real and different");
         root1=(-b +Math.sqrt(d))/(2.0*a);
          root2=(-b - Math.sqrt(d))/(2.0*a);
         System.out.println("Root1="+root1);
          System.out.println("Root2="+root2);
       }

       else if(d<0)
      {
         System.out.println("Roots are imaginary");
         realpart = -b/(2*a);
          imagpart = Math.sqrt(-d)/(2.0*a);
         System.out.println("Root1="+realpart+"+"
                                  +imagpart+"i");
          System.out.println("Root2="+realpart+"-"
                                     +imagpart+"i");
       }
}}
```

Code:

```
import java.util.Scanner;
import java.lang.Math;

class Quadratic{
	int a,b,c;
	double root1,root2,realpart,imagpart,d;


	void getd()
{
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter the value of a");
	a=sc.nextInt();
	System.out.println("Enter the value of b");
	b=sc.nextInt();
	System.out.println("Enter the value of c");
	c=sc.nextInt();
}

	void compute()
{
	d=b*b-4*a*c;
	if(d==0)
{
	root1=root2=-b/(2*a);
	System.out.println("Roots are equal");
	System.out.println("Root1="+root1);
	System.out.println("Root2="+root2);
}
	else if(d>0)
{	System.out.println("Roots are real and different");

	root1=(-b+Math.sqrt(d))/(2.0*a);
	root2=(-b-Math.sqrt(d))/(2.0*a);

	System.out.println("Root1="+root1);
	System.out.println("Root2="+root2);
}
	else if(d<0)
{
	System.out.println("Roots are imaginary");
	realpart=-b/(2*a);
	imagpart = Math.sqrt(-d) / (2.0 * a);
	System.out.println("Root1= " + realpart + " + " + imagpart + "i");
	System.out.println("Root2= " + realpart + " - " + imagpart + "i");
}
}}
class QuadraticMain{

        public static void main(String args[])
{
        Quadratic q=new Quadratic();
        q.getd();
        q.compute();
        System.out.println("name:Keerthana H Bhat");
        System.out.println("USN:1BM23CS148");

}
}
```

Output:

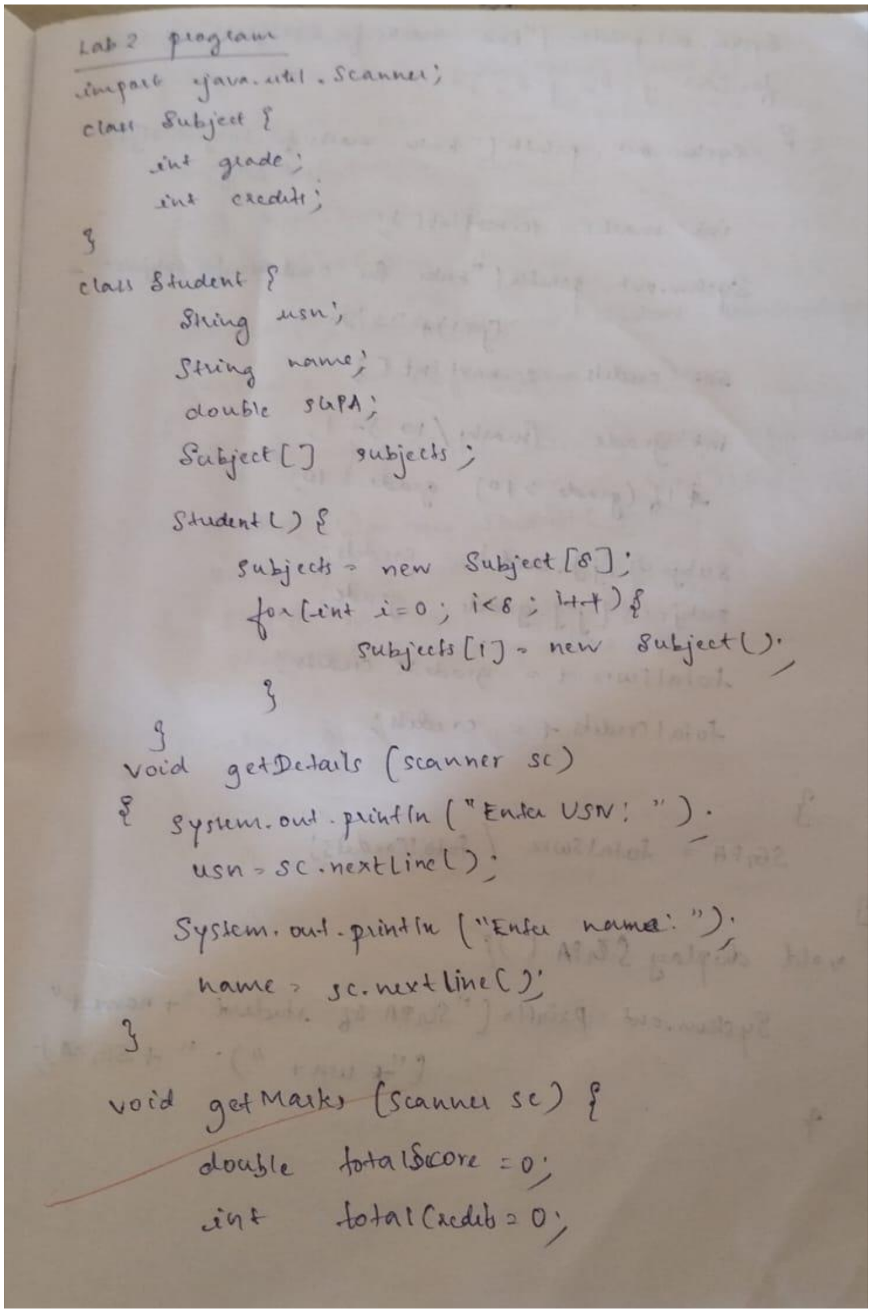


```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS148>javac QuadraticMain.java

D:\1BM23CS148>java QuadraticMain
Enter the value of a
1
Enter the value of b
3
Enter the value of c
2
Roots are real and different
Root1=-1.0
Root2=-2.0

D:\1BM23CS148>java QuadraticMain
Enter the value of a
1
Enter the value of b
5
Enter the value of c
6
Roots are real and different
Root1=-2.0
Root2=-3.0

D:\1BM23CS148>javac QuadraticMain.java

D:\1BM23CS148>java QuadraticMain
Enter the value of a
1
Enter the value of b
2
Enter the value of c
1
Roots are equal
Root1=-1.0
Root2=-1.0

D:\1BM23CS148>javac QuadraticMain.java
```

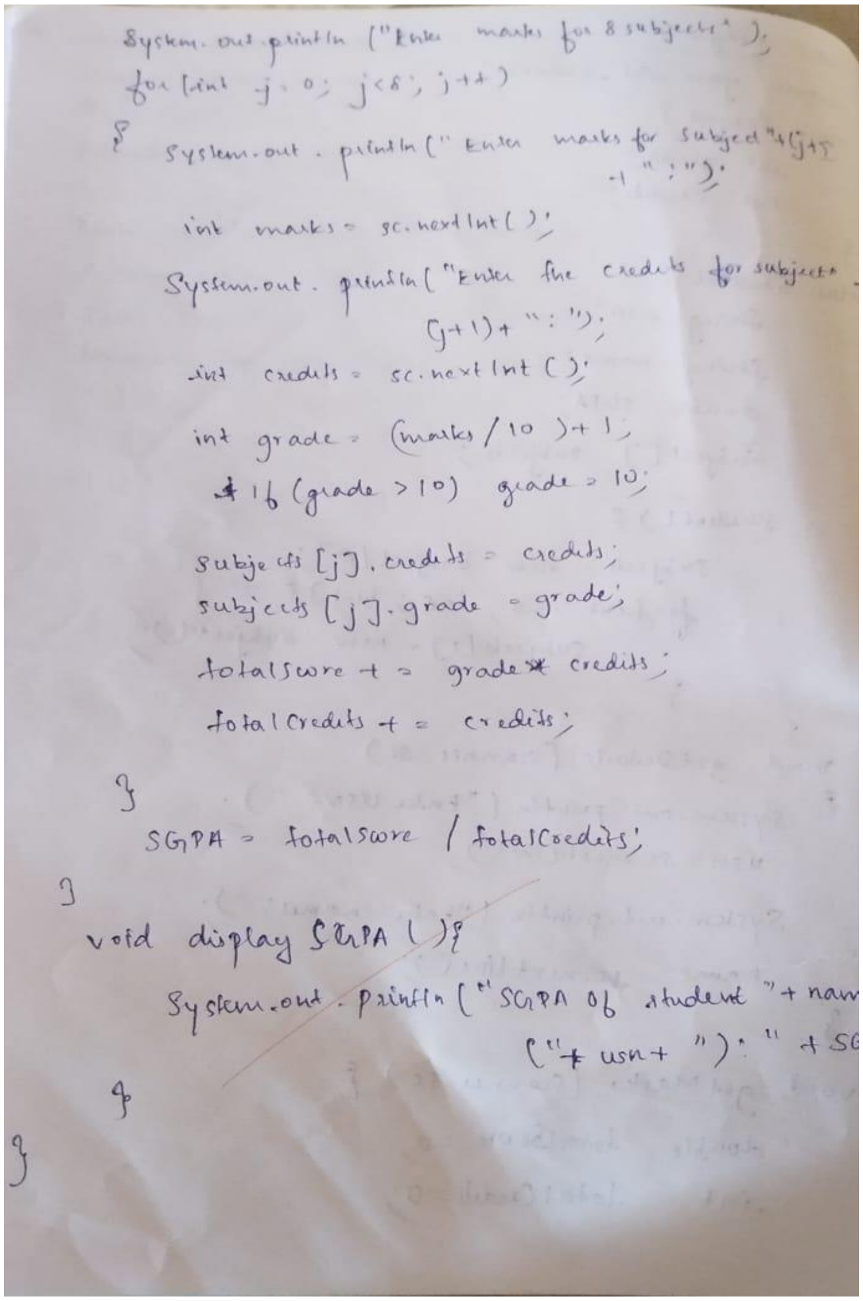
Program 2

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.

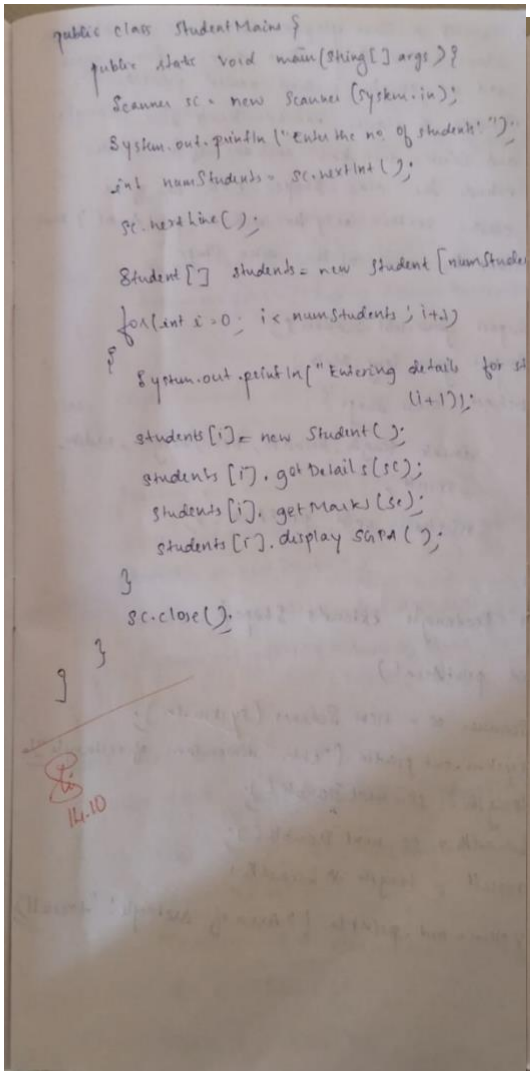
Algorithm:



```
Lab 2 program
import java.util.Scanner;
class Subject {
      int grade;
      int credits;
}
class Student {
          String usn;
          String name;
          double SGPA;
          Subject[] subjects;

          Student() {
              subjects = new Subject[8];
              for(int i=0; i<8; i++){
                       subjects[i] = new Subject();
              }
      }
   void getDetails (scanner sc)
   { System.out.println ("Enter USN: ").
       usn = sc.nextLine();
     System.out.println ("Enter name: ");
       name = sc.nextLine();
   }

void getMarks (Scanner sc) {
       double totalScore = 0;
       int    totalCredits = 0;
```

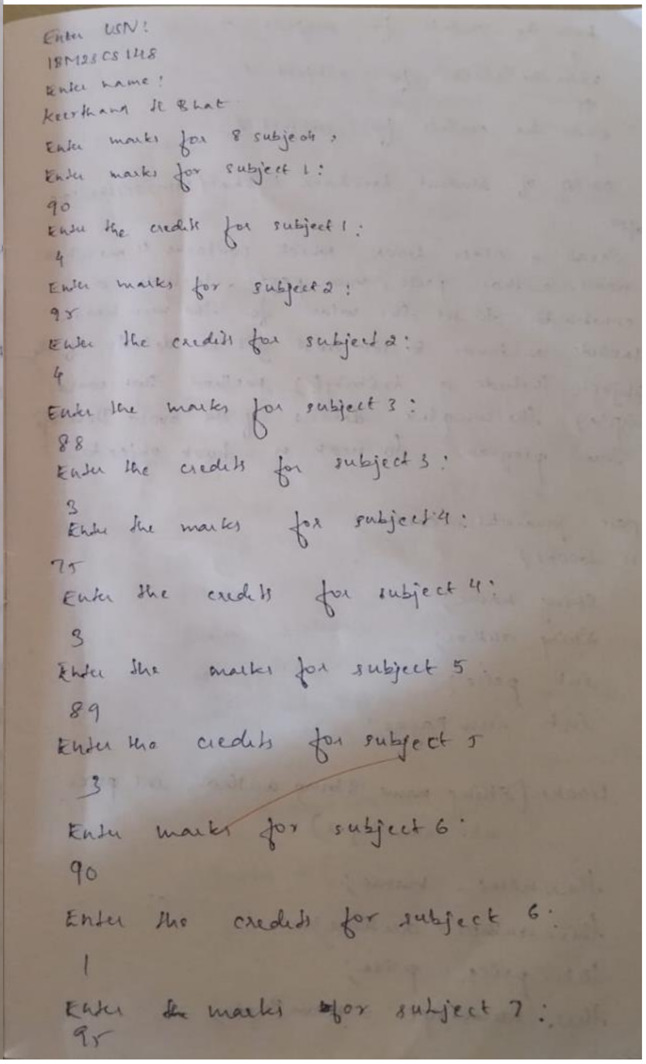


```
System.out.println ("Enter marks for 8 subjects" );
for (int j = 0; j<8; j++)
{
    System.out.println(" Enter marks for subject "+(j+1
                                   +" : ");

    int marks = sc.nextInt();

    System.out.println("Enter the credits for subject"
                        (j+1)+ ": ");

    int credits = sc.nextInt();

    int grade = (marks / 10 )+ 1;
     if (grade >10) grade = 10;

    subjects [j].credits = credits;
    subjects [j].grade = grade;
    totalscore += grade * credits;
    totalCredits += credits;

}
  SGPA = totalscore / totalCredits;
}
void display SGPA (){
    System.out.println ("SGPA of student "+ nam
                         ("+ usn+ "): " + SG
}
}
```

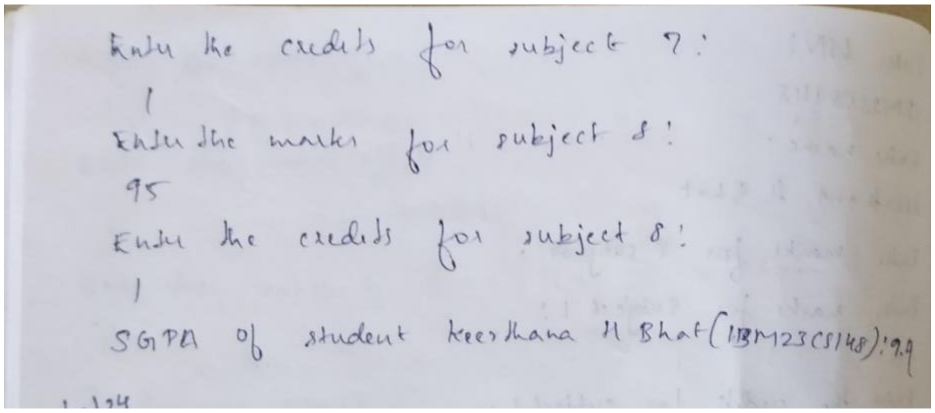


```
public class StudentMain {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the no. of students: ");
    int numStudents = sc.nextInt();
    sc.nextLine();
    Student[] students = new Student[numStude
    for(int i = 0; i < numStudents; i++)
    {
      System.out.println("Entering details for st
                                   (i+1));
      students[i] = new Student();
      students[i].getDetails(sc);
      students[i].getMarks(sc);
      students[i].displaySGPA();
    }
    sc.close();
  }
}
```

14.10



```
Enter USN:
1BM23CS148
Enter name:
Keerthana H Bhat
Enter marks for 8 subjects:
Enter marks for subject 1:
90
Enter the credits for subject 1:
4
Enter marks for subject 2:
95
Enter the credits for subject 2:
4
Enter the marks for subject 3:
88
Enter the credits for subject 3:
3
Enter the marks for subject 4:
75
Enter the credits for subject 4:
3
Enter the marks for subject 5
89
Enter the credits for subject 5
3
Enter marks for subject 6:
90
Enter the credits for subject 6:
1
Enter ~~the~~ marks ~~for~~ for subject 7:
95
```

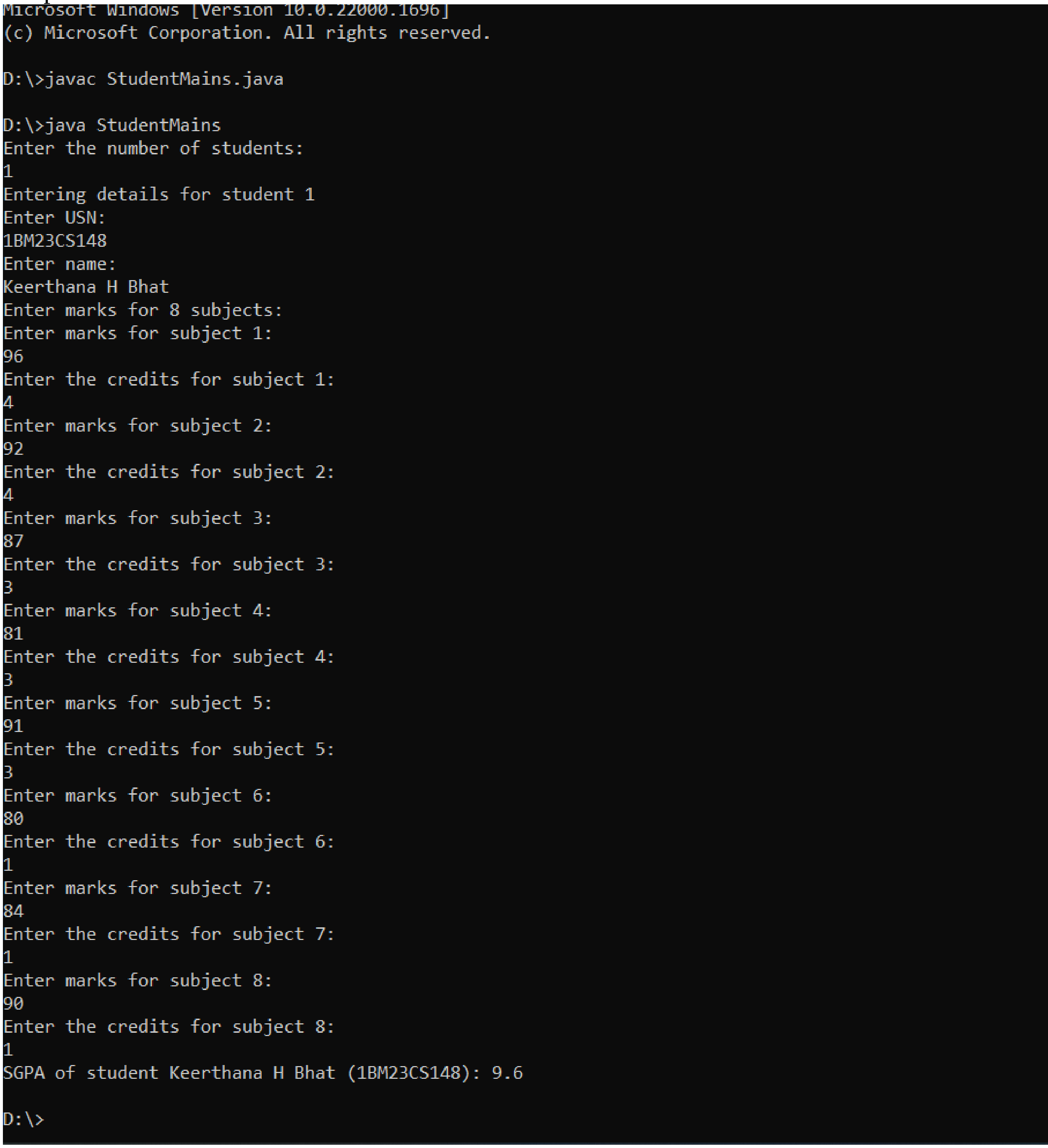


```
Enter the credits for subject 7:
1
Enter the marks for subject 8:
95
Enter the credits for subject 8:
1
SGPA of student Keerthana H Bhat(1BM23CS148): 9.9
```

Code:

```
import java.util.Scanner;

class Subject {
    int grade;
    int credits;
}

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


    Student() {
        subjects = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subjects[i] = new Subject();
        }
    }

    void getDetails(Scanner sc) {
        System.out.println("Enter USN:");
        usn = sc.nextLine();
        System.out.println("Enter name:");
        name = sc.nextLine();
    }

    void getMarks(Scanner sc) {
        double totalScore = 0;
        int totalCredits = 0;

        System.out.println("Enter marks for 8 subjects:");
        for (int j = 0; j < 8; j++) {
            System.out.println("Enter marks for subject " + (j + 1) + ":");
            int marks = sc.nextInt();
            System.out.println("Enter the credits for subject " + (j + 1) + ":");
            int credits = sc.nextInt();

            int grade = (marks / 10) + 1; // Calculate grade based on marks
            if (grade > 10) grade = 10; // Cap grade at 10

            // Store the information in the subjects array
            subjects[j].credits = credits;
            subjects[j].grade = grade;
            // Calculate score based on grade and credits
            totalScore += grade * credits;
            totalCredits += credits; // Accumulate total credits
        }

        // Compute SGPA
        SGPA = totalScore / totalCredits;
    }

    void displaySGPA() {
        System.out.println("SGPA of student " + name + " (" + usn + "): " + SGPA);
    }
}

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

        System.out.println("Enter the number of students:");
        int numStudents = sc.nextInt();
        sc.nextLine(); // Consume the newline

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Entering details for student " + (i + 1));
            students[i] = new Student();
            students[i].getDetails(sc);
            students[i].getMarks(sc);
            students[i].displaySGPA();
        }

        sc.close(); // Close the scanner
    }
}
```

Output:

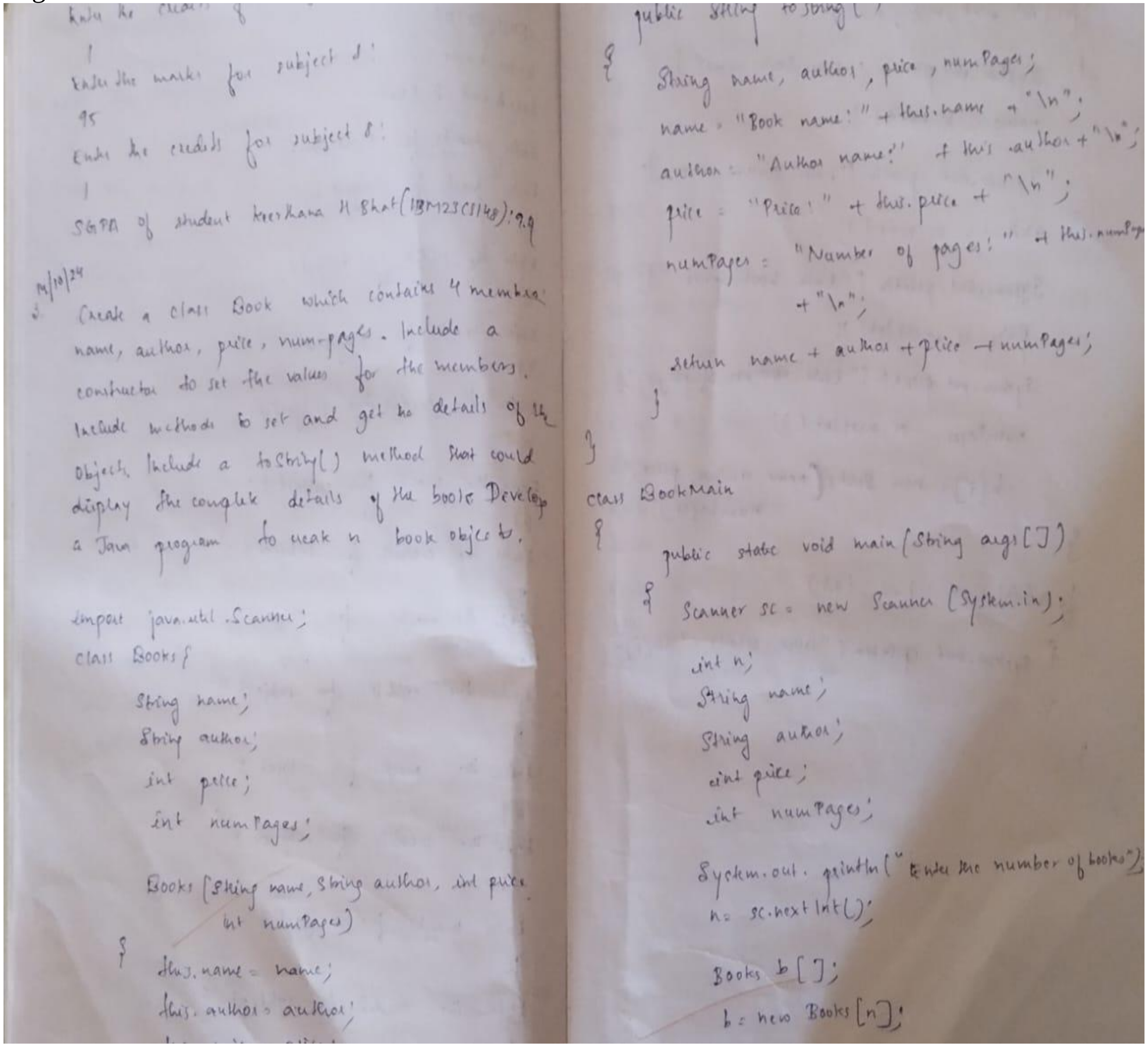


```
Microsoft Windows [Version 10.0.22000.1696]
(c) Microsoft Corporation. All rights reserved.

D:\>javac StudentMains.java

D:\>java StudentMains
Enter the number of students:
1
Entering details for student 1
Enter USN:
1BM23CS148
Enter name:
Keerthana H Bhat
Enter marks for 8 subjects:
Enter marks for subject 1:
96
Enter the credits for subject 1:
4
Enter marks for subject 2:
92
Enter the credits for subject 2:
4
Enter marks for subject 3:
87
Enter the credits for subject 3:
3
Enter marks for subject 4:
81
Enter the credits for subject 4:
3
Enter marks for subject 5:
91
Enter the credits for subject 5:
3
Enter marks for subject 6:
80
Enter the credits for subject 6:
1
Enter marks for subject 7:
84
Enter the credits for subject 7:
1
Enter marks for subject 8:
90
Enter the credits for subject 8:
1
SGPA of student Keerthana H Bhat (1BM23CS148): 9.6

D:\>
```

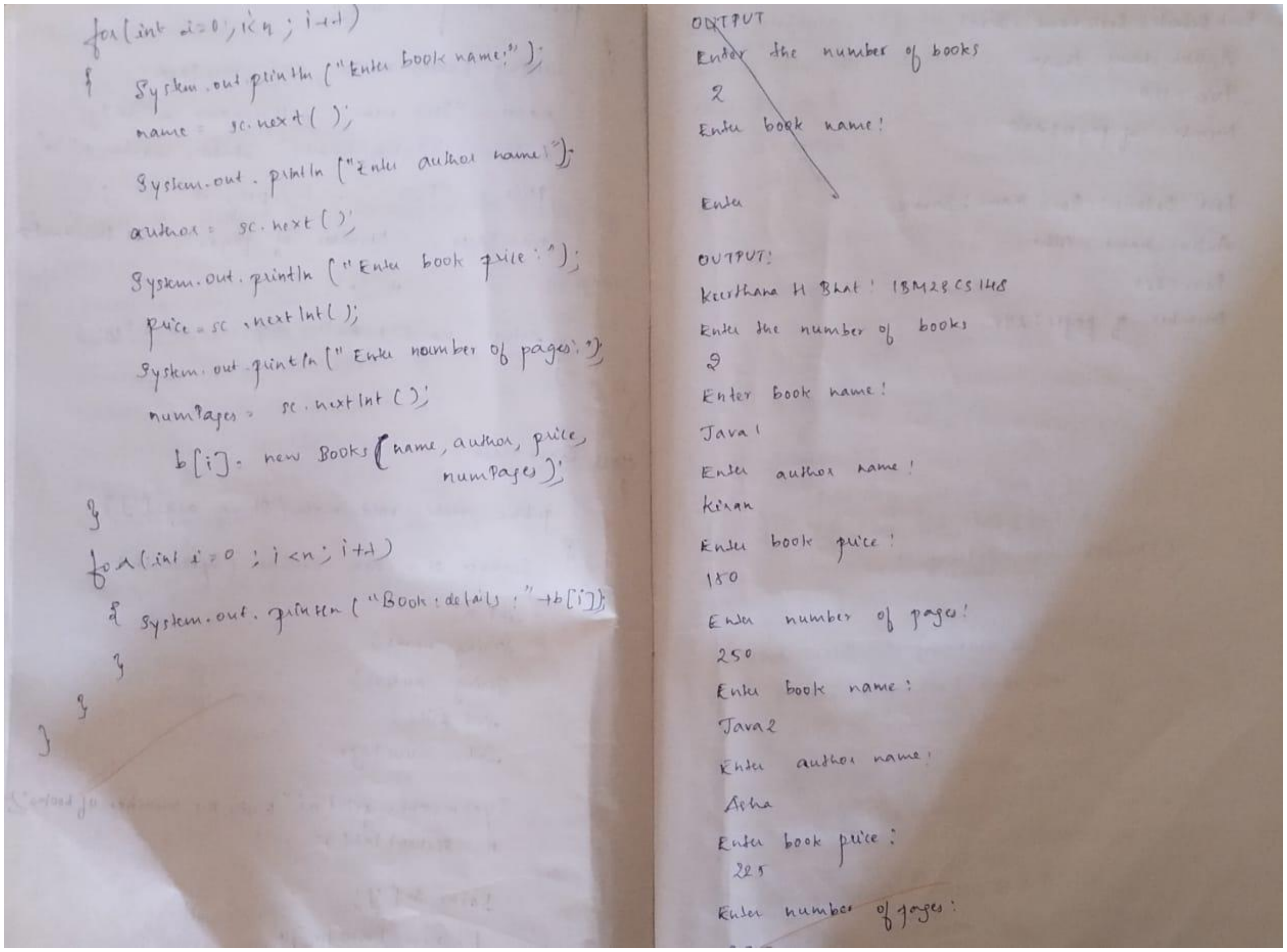
<u>Program 3</u>

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.

Algorithm:

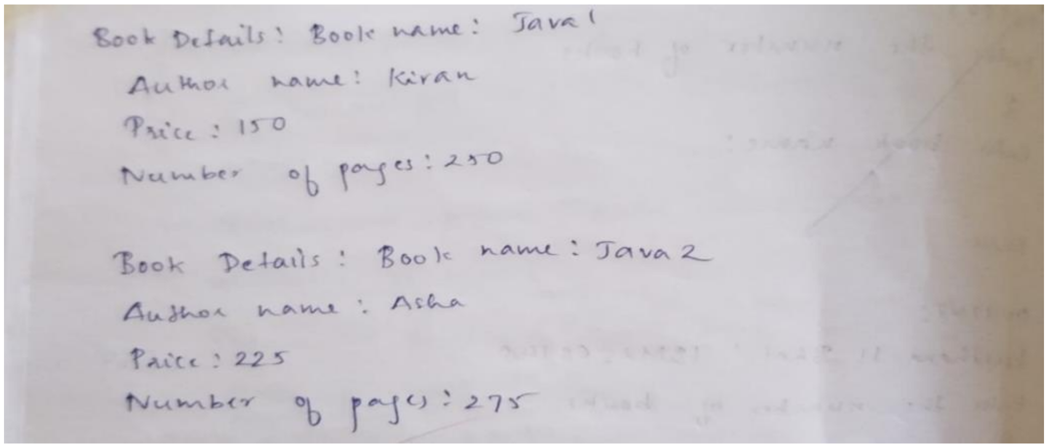


```
Enter the credits of
1
Enter the marks for subject 8:
95
Enter the credits for subject 8:
1
SGPA of student Keerthana H Bhat(1BM23CS148):9.9

14/10/24
3   Create a class Book which contains 4 members
    name, author, price, num-pages. Include a
    constructor to set the values for the members.
    Include methods to set and get the details of the
    objects. Include a toString() method that could
    display the complete details of the book. Develop
    a Java program to create n book objects.

import java.util.Scanner;
class Books {
        String name;
        String author;
        int price;
        int numPages;

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

   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 BookMain
{
     public static void main(String args[])
  {  Scanner sc = new Scanner(System.in);

        int n;
        String name;
        String author;
        int price;
        int numPages;

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

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



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

```
~~OUTPUT~~
~~Enter the number of books~~
~~2~~
~~Enter book name:~~

~~Enter~~

OUTPUT:
Keerthana H Bhat : 1BM23CS148
Enter the number of books
2
Enter book name:
Java1
Enter author name:
Kiran
Enter book price:
150
Enter number of pages:
250
Enter book name:
Java2
Enter author name:
Asha
Enter book price:
225
Enter number of pages:
```

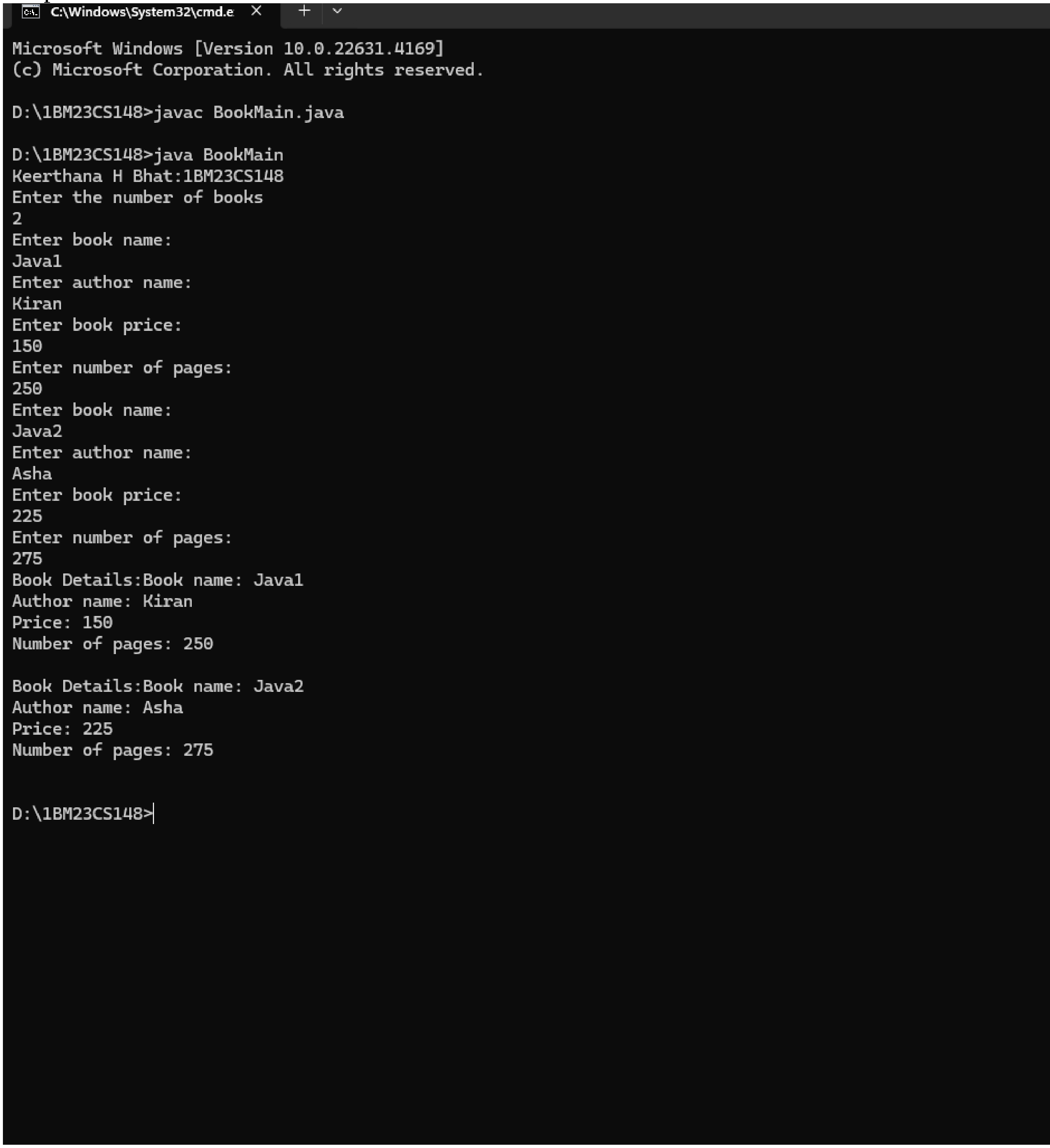


```
Book Details: Book name: Java1
 Author name: Kiran
 Price : 150
 Number of pages: 250

Book Details: Book name: Java2
 Author name: Asha
 Price: 225
 Number of pages: 275
```

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 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 BookMain

{

        public static void main(String args[])
        {

        Scanner sc = new Scanner(System.in);

        int n;
        String name;
        String author;
        int price;
        int numPages;

        System.out.println("Keerthana H Bhat:1BM23CS148");

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

        Books b[];

        b = new Books[n];

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

                b[i] = new Books(name,author,price,numPages);
        }
        for(int i=0;i<n;i++)
        {
                System.out.println("Book Details:"+b[i]);
        }
        }
}
```

Output:

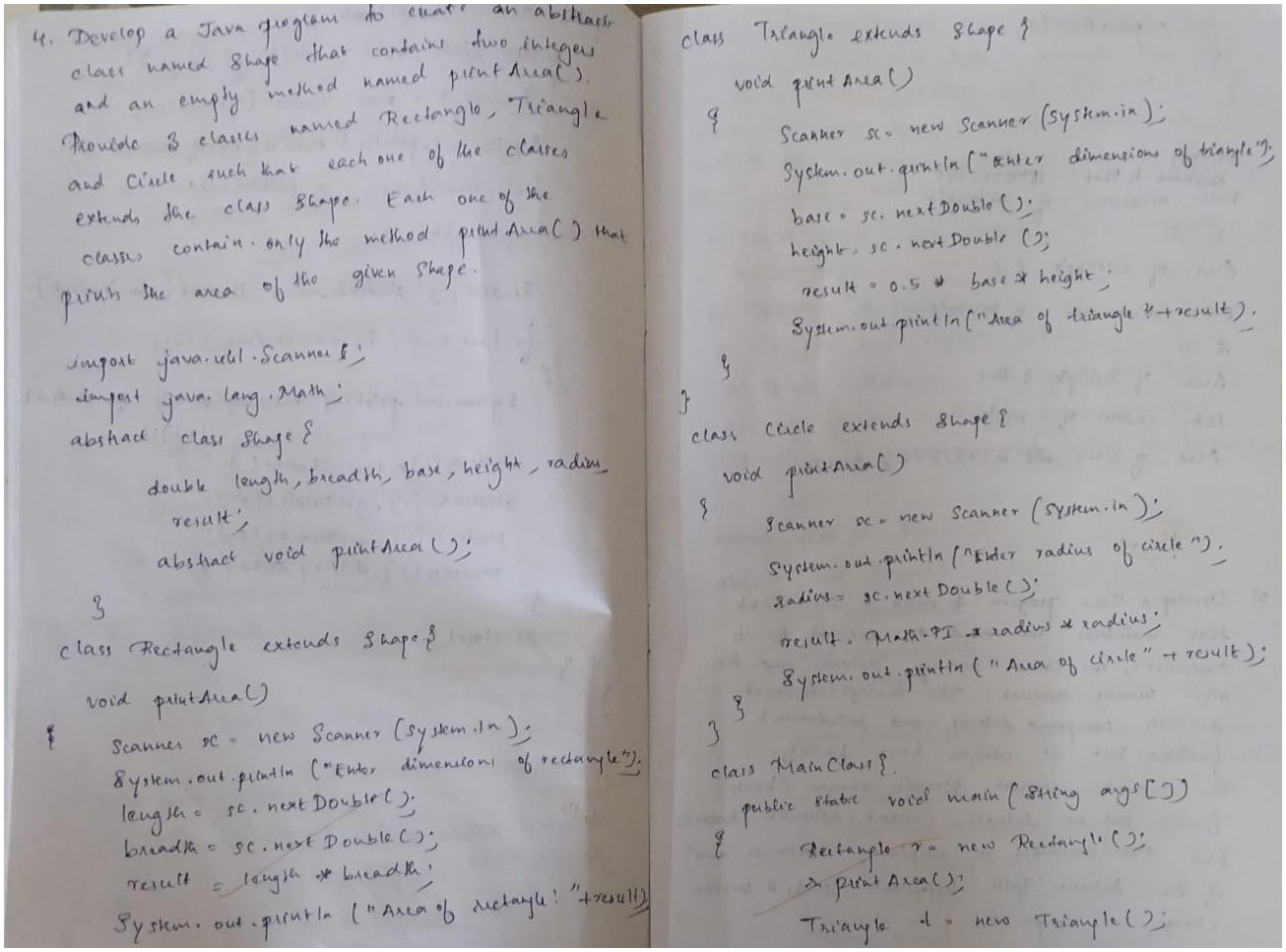


```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS148>javac BookMain.java

D:\1BM23CS148>java BookMain
Keerthana H Bhat:1BM23CS148
Enter the number of books
2
Enter book name:
Java1
Enter author name:
Kiran
Enter book price:
150
Enter number of pages:
250
Enter book name:
Java2
Enter author name:
Asha
Enter book price:
225
Enter number of pages:
275
Book Details:Book name: Java1
Author name: Kiran
Price: 150
Number of pages: 250

Book Details:Book name: Java2
Author name: Asha
Price: 225
Number of pages: 275


D:\1BM23CS148>
```

Program 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.

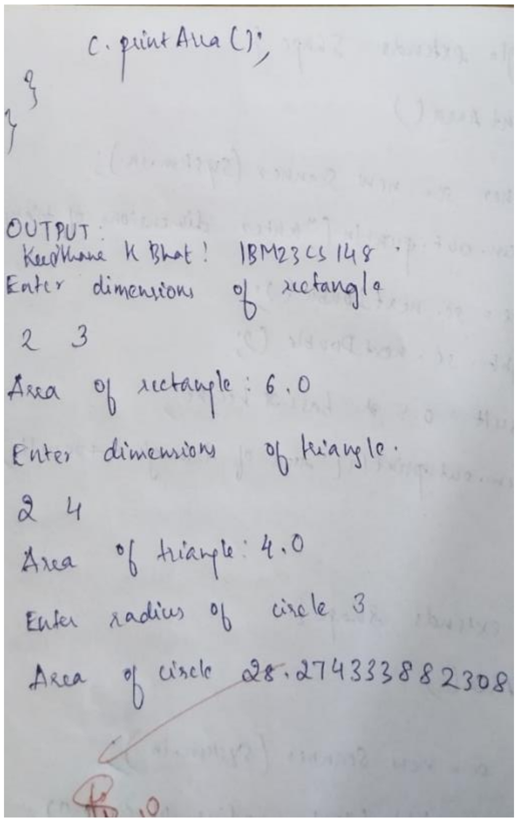
Algorithm:



4. Develop a Java program to creat an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide 3 classes named Rectangle, Triangle and Circle, such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given Shape.

```
import java.util.Scanner;
import java.lang.Math;
abstract class Shape {
        double length, breadth, base, height, radius,
          result;
        abstract void printArea();

  }
class Rectangle extends Shape {
  void printArea()
{    Scanner sc = new Scanner (System.in);
     System.out.println ("Enter dimensions of rectangle");
     length = sc.nextDouble();
     breadth = sc.nextDouble();
     result = length * breadth;
     System.out.println ("Area of rectangle: "+result);
```

```
class Triangle extends Shape {
    void printArea()
  {      Scanner sc = new Scanner (System.in);
         System.out.println ("Enter dimensions of triangle");
         base = sc.nextDouble();
         height = sc.nextDouble();
         result = 0.5 * base * height;
         System.out.println ("Area of triangle "+result);

  }
}
class Circle extends Shape {
  void printArea()
  {     Scanner sc = new Scanner (System.in);
        System.out.println ("Enter radius of circle");
        radius = sc.nextDouble();
        result = Math.PI * radius * radius;
        System.out.println ("Area of circle" + result);
    }
  }
class MainClass {
  public static void main (String args[])
  {      Rectangle r = new Rectangle();
         r.printArea();
         Triangle t = new Triangle();
```



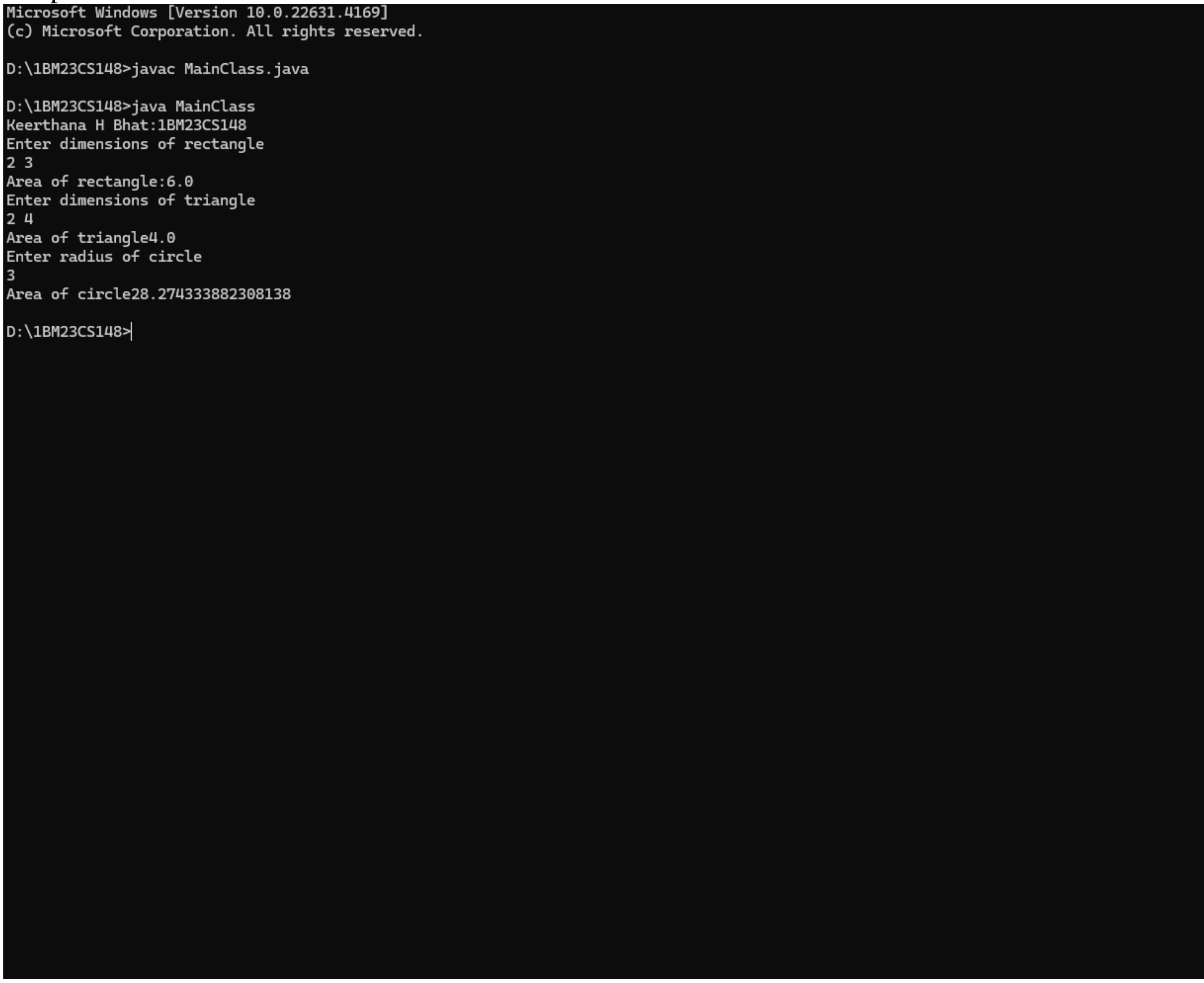
```
        c.printArea();
    }
}

OUTPUT:
Keerthana K Bhat! 1BM23CS148.
Enter dimensions of rectangle
 2  3
Area of rectangle: 6.0
Enter dimensions of triangle.
 2  4
Area of triangle: 4.0
Enter radius of circle 3
Area of circle 28.274333882308
```

Code:

```
import java.util.Scanner;
import java.lang.Math;
abstract class Shape{

            double length,breadth,base,height,radius,result;

            abstract void printArea();
}


class Rectangle extends Shape{

void printArea()
{

            Scanner sc=new Scanner(System.in);
            System.out.println("Enter dimensions of rectangle");
            length=sc.nextDouble();
            breadth=sc.nextDouble();
            result=length*breadth;
            System.out.println("Area of rectangle:"+result);
}
```

```
}
class Triangle extends Shape{

void printArea()
{
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter dimensions of triangle");

        base=sc.nextDouble();
        height=sc.nextDouble();
        result=0.5*base*height;
        System.out.println("Area of triangle"+result);
}
}
class Circle extends Shape{

void printArea()
{
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter radius of circle");
        radius=sc.nextDouble();


        result=Math.PI*radius*radius;
        System.out.println("Area of circle"+result);
}
}
class MainClass{

public static void main(String args[])
{

        System.out.println("Keerthana H Bhat:1BM23CS148");

        Rectangle r =new Rectangle();

        r.printArea();

        Triangle t=new Triangle();

        t.printArea();

        Circle c = new Circle();

        c.printArea();
}
```

Output:



```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS148>javac MainClass.java

D:\1BM23CS148>java MainClass
Keerthana H Bhat:1BM23CS148
Enter dimensions of rectangle
2 3
Area of rectangle:6.0
Enter dimensions of triangle
2 4
Area of triangle4.0
Enter radius of circle
3
Area of circle28.274333882308138

D:\1BM23CS148>
```

Program 5
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called a 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 a 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 deposits from customers 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 a penalty if necessary and update the balance.

Algorithm:

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

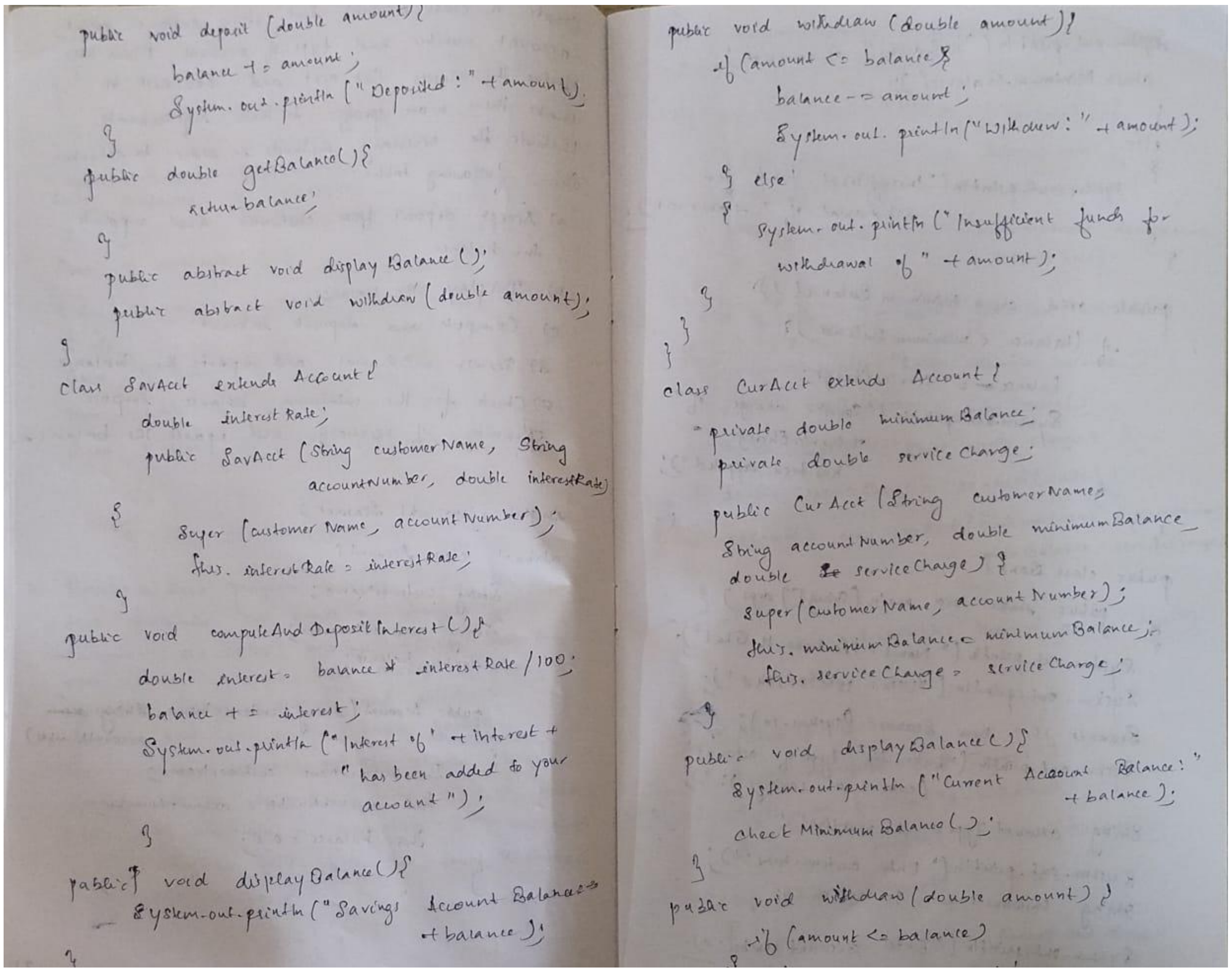
c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

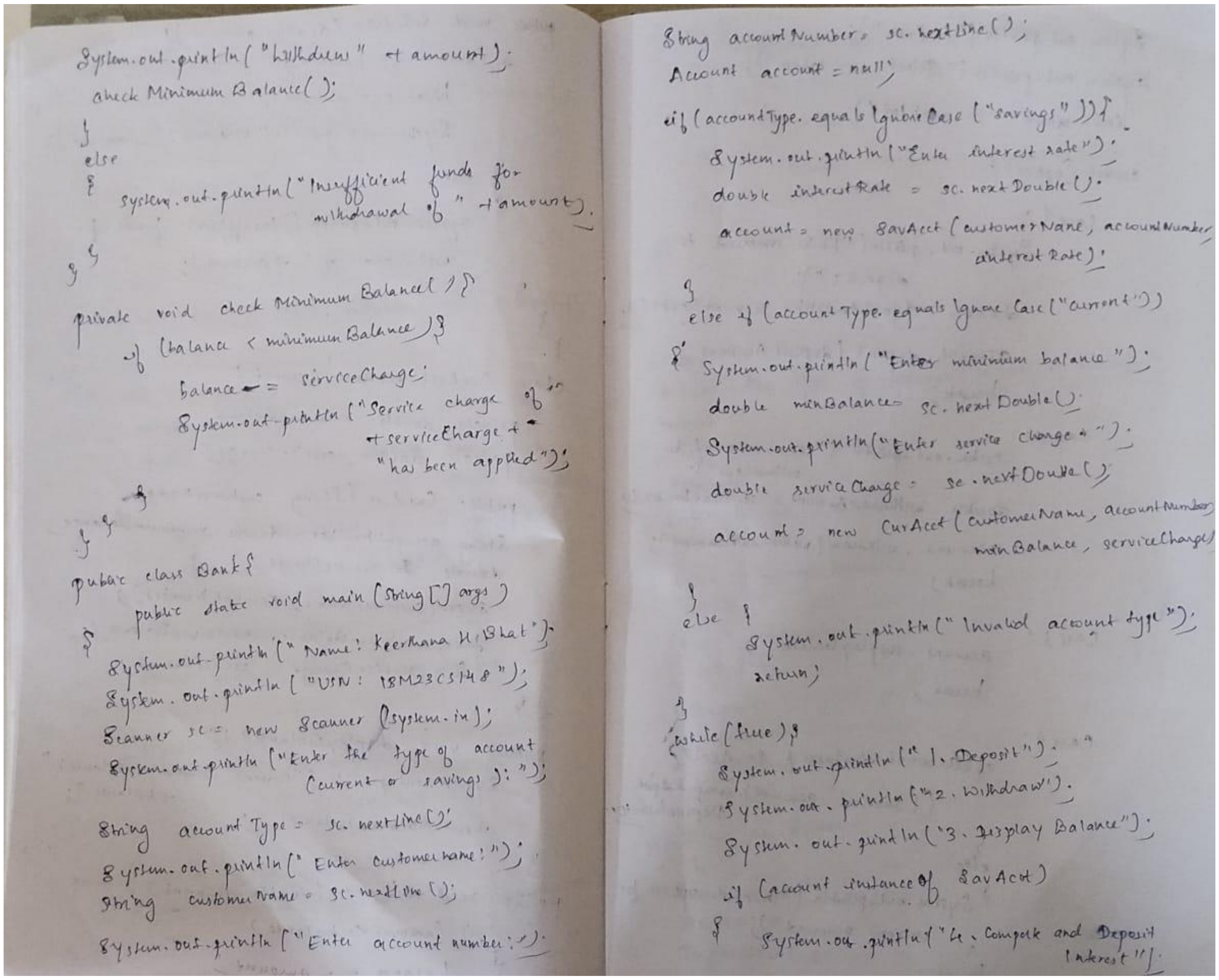
```
import java.util.Scanner;
abstract class Account {
        String customerName;
        String accountNumber;
        double balance;

        public Account (String customerName, String ~~accoun~~ accountNumber)
        {
            this.customerName = customerName;
            this.accountNumber = accountNumber;
            this.balance = 0.0;
        }
```

2



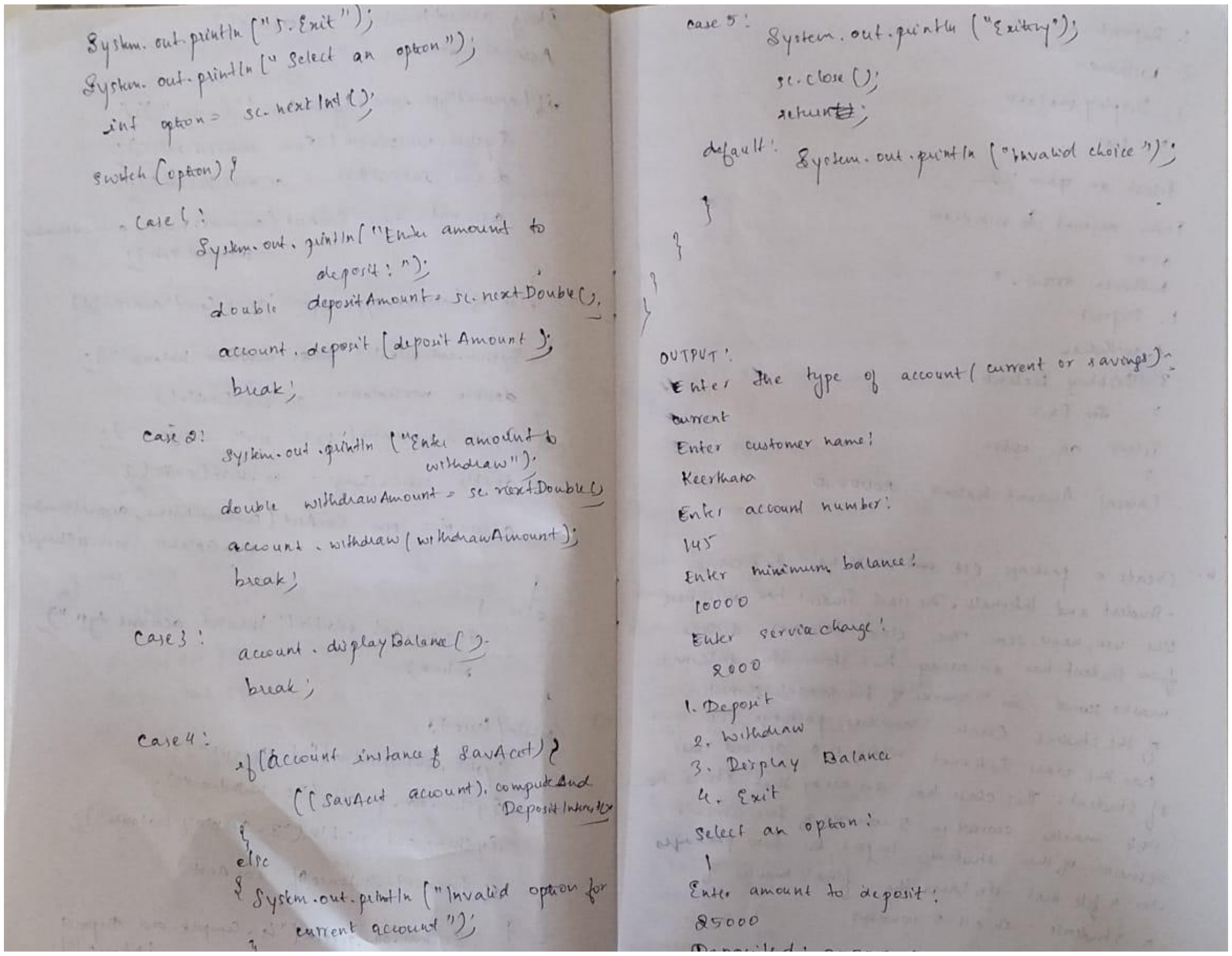
```
public void deposit (double amount){
        balance += amount;
        System.out.println ("Deposited : " +amount);
  }
public double getBalance(){
        return balance;
  }
  public abstract void display Balance ();
  public abstract void withdraw (double amount);
}
class SavAcct extends Account {
      double interest Rate;
      public SavAcct (String customerName, String
                          accountNumber, double interestRate)
  {
         Super (customer Name, account Number);
         this. interestRate = interestRate;
  }
public void computeAndDepositInterest (){
      double interest = balance * interestRate /100;
      balance += interest;
      System.out.println ("Interest of' + interest +
                     " has been added to your
                            account");
    }
public void displayBalance(){
   System.out.println ("Savings Account Balance:"
                          +balance);
}
```

```
public void withdraw (double amount){
    if (amount <= balance){
         balance -= amount;
         System.out. println("Withdrew: " + amount);
    } else
    {
      System.out.println ("Insufficient funds for
         withdrawal of " +amount);
    }
  }
}
class CurAcct extends Account {
   private double minimumBalance;
   private double service Charge;
    public CurAcct (String customerName,
    String accountNumber, double minimumBalance
    double ~~St~~ serviceCharge) {
     super(CustomerName, account Number);
     this. minimumBalance = minimumBalance;
      this. serviceCharge = serviceCharge;
  }
public void displayBalance(){
   System.out.println ("Current Account Balance:"
                                  + balance);
   check MinimumBalance();
  }
public void withdraw(double amount) {
      if (amount <= balance)
```



```
System.out.println("Withdrew" + amount);
 check Minimum Balance();
}
else
{
   system.out.println("Insufficient funds for
                        withdrawal of " + amount);
}
}
private void check Minimum Balance(){
    if (balance < minimum Balance){
         balance -= serviceCharge;
         System.out.println("Service charge of "
                        + serviceCharge +
                        "has been applied");
    }
}
}
public class Bank{
     public static void main(String[] args)
{
  System.out.println(" Name: Keerthana H Bhat");
  System.out.println("USN: 1BM23CS148");
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter the type of account
                 (current or savings): ");
  String account Type = sc.nextLine();
  System.out.println(" Enter customer name:");
  String customerName = sc.nextLine();
  System.out.println("Enter account number:");
```

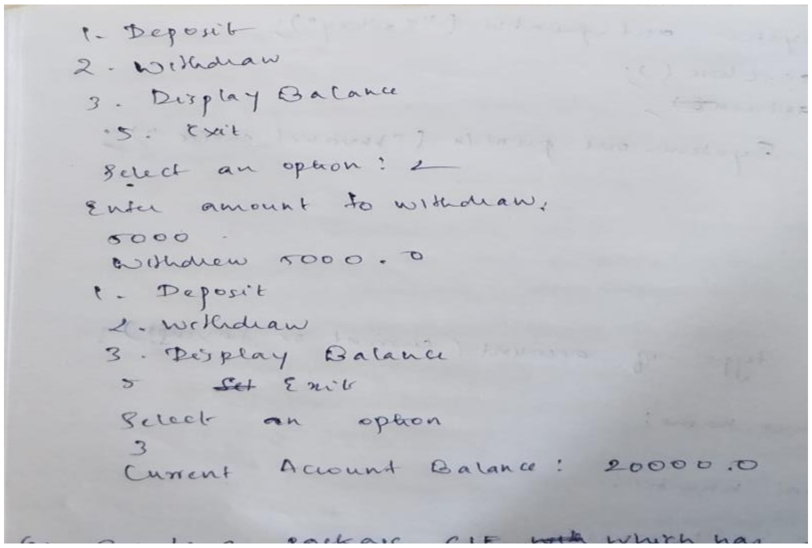
```
String accountNumber = sc.nextLine();
Account account = null;
if(accountType.equalsIgnoreCase("savings")){
    System.out.println("Enter interest rate");
    double interestRate = sc.nextDouble();
    account = new SavAcct(customerName, accountNumber,
                       interestRate);
 }
 else if (accountType.equalsIgnoreCase("current"))
{  System.out.println("Enter minimum balance");
   double minBalance = sc.nextDouble();
   System.out.println("Enter service charge:");
   double serviceCharge = sc.nextDouble();
   account = new CurAcct(customerName, accountNumber,
                       minBalance, serviceCharge)
 }
 else {
      System.out.println("Invalid account type");
      return;
 }
while(true){
   System.out.println("1. Deposit");
   System.out.println("2. Withdraw");
   System.out.println("3. Display Balance");
   if (account instanceof SavAcct)
   {    System.out.println("4. Compute and Deposit
                          Interest");
```



```
System.out.println("5.Exit");
System.out.println(" Select an option");
 int option = sc.nextInt();
switch (option) {
   case 1:
        System.out.println("Enter amount to
                        deposit: ");
        double depositAmount = sc.nextDouble();
        account.deposit(depositAmount);
        break;
   case 2:
        System.out.println("Enter amount to
                              withdraw");
        double withdrawAmount = sc.nextDouble()
        account.withdraw(withdrawAmount);
        break;
   case 3:
        account.displayBalance();
        break;
   case 4:
        if(account instanceof SavAcct){
             ((SavAcct account).computeAnd
                              DepositInterest()
        }
        else
        { System.out.println("Invalid option for
                current account");
        }
   case 5: System.out.println("Exiting");
           sc.close();
           return;
   default: System.out.println("Invalid choice");
   }
 }
}
}
```

OUTPUT:

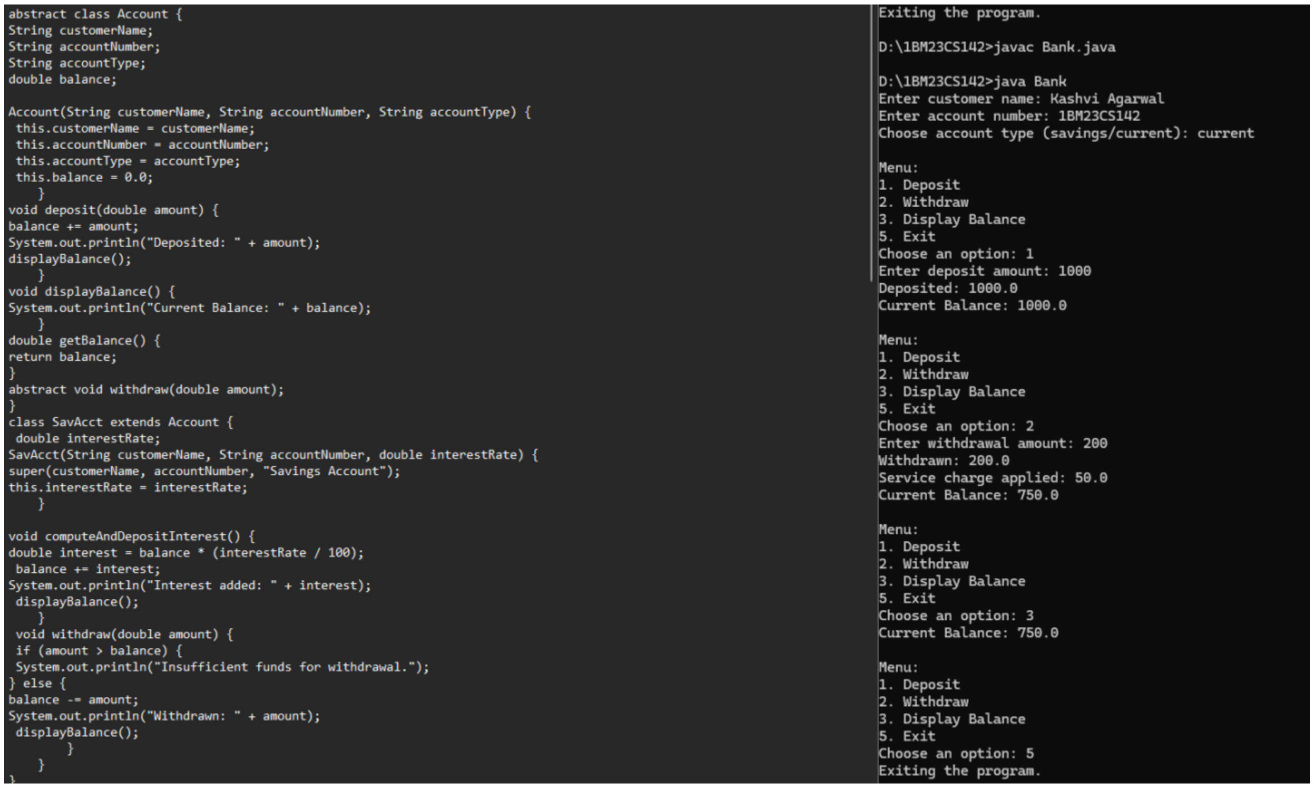
```
Enter the type of account( current or savings):
current
Enter customer name:
Keerthana
Enter account number:
145
Enter minimum balance:
10000
Enter service charge:
2000
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Select an option:
1
Enter amount to deposit:
25000
```



```
1. Deposit
2. Withdraw
3. Display Balance
5. Exit
Select an option: 2
Enter amount to withdraw:
5000
Withdrew 5000.0
1. Deposit
2. Withdraw
3. Display Balance
5   ~~Set~~ Exit
Select an option
3
Current Account Balance: 20000.0
```

Code:
```
import java.util.Scanner;

abstract class Account {
    String customerName;
    String accountNumber;
    double balance;

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

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }

    public double getBalance() {
        return balance;
    }

    public abstract void displayBalance();
    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, String accountNumber, double interestRate) {
        super(customerName, accountNumber);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest of " + interest + " has been added to your account");
    }

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

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Insufficient funds for withdrawal of " + amount);
        }
    }
}

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

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

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

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew " + amount);
            checkMinimumBalance();
        } else {
            System.out.println("Insufficient funds for withdrawal of " + amount);
        }
    }

    private void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " has been applied");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
            System.out.println("Name:Keerthana H Bhat");
            System.out.println("USN:1BM23CS148");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the type of account (current or savings):");
        String accountType = sc.nextLine(); // Fixed variable name
        System.out.println("Enter customer name:");
        String customerName = sc.nextLine(); // Fixed variable name
        System.out.println("Enter account number:");
        String accountNumber = sc.nextLine();
        Account account = null;

        if (accountType.equalsIgnoreCase("savings")) {
            System.out.println("Enter interest rate:");
            double interestRate = sc.nextDouble();
            account = new SavAcct(customerName, accountNumber, interestRate);
        } else if (accountType.equalsIgnoreCase("current")) {
            System.out.println("Enter minimum balance:");
            double minBalance = sc.nextDouble();
            System.out.println("Enter service charge:");
            double serviceCharge = sc.nextDouble();
            account = new CurAcct(customerName, accountNumber, minBalance,
serviceCharge);
        } else {
            System.out.println("Invalid account type");
            return;
        }

        while (true) {
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            if (account instanceof SavAcct) {
                System.out.println("4. Compute and Deposit Interest");
            }
            System.out.println("5. Exit");
            System.out.println("Select an option:");
            int option = sc.nextInt();

            switch (option) {
                case 1:
                    System.out.println("Enter amount to deposit:");
                    double depositAmount = sc.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.println("Enter amount to withdraw:");
                    double withdrawAmount = sc.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    if (account instanceof SavAcct) {
                        ((SavAcct) account).computeAndDepositInterest();
                    } else {
                        System.out.println("Invalid option for current account");
                    }
                    break;
                case 5:
                    System.out.println("Exiting");
                    sc.close();
                    return;
                default:
                    System.out.println("Invalid option");
            }
        }
      }
}
```

Output:



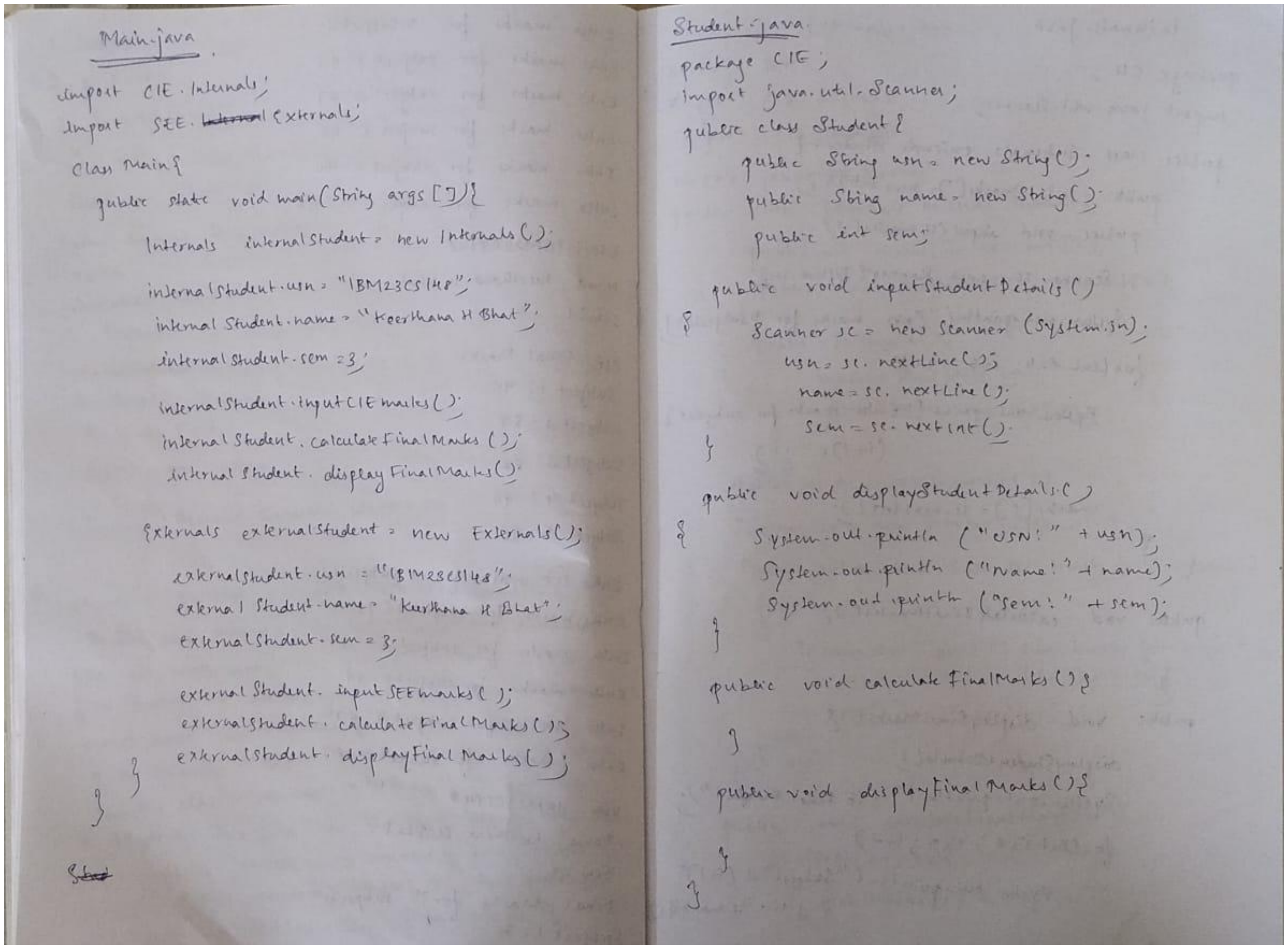
```
abstract class Account {
String customerName;
String accountNumber;
String accountType;
double balance;

Account(String customerName, String accountNumber, String accountType) {
 this.customerName = customerName;
 this.accountNumber = accountNumber;
 this.accountType = accountType;
 this.balance = 0.0;
    }
void deposit(double amount) {
balance += amount;
System.out.println("Deposited: " + amount);
displayBalance();
    }
void displayBalance() {
System.out.println("Current Balance: " + balance);
    }
double getBalance() {
return balance;
}
abstract void withdraw(double amount);
}
class SavAcct extends Account {
 double interestRate;
SavAcct(String customerName, String accountNumber, double interestRate) {
super(customerName, accountNumber, "Savings Account");
this.interestRate = interestRate;
    }

void computeAndDepositInterest() {
double interest = balance * (interestRate / 100);
 balance += interest;
System.out.println("Interest added: " + interest);
 displayBalance();
    }
 void withdraw(double amount) {
 if (amount > balance) {
 System.out.println("Insufficient funds for withdrawal.");
} else {
balance -= amount;
System.out.println("Withdrawn: " + amount);
 displayBalance();
        }
    }
}
```

```
Exiting the program.

D:\1BM23CS142>javac Bank.java

D:\1BM23CS142>java Bank
Enter customer name: Kashvi Agarwal
Enter account number: 1BM23CS142
Choose account type (savings/current): current

Menu:
1. Deposit
2. Withdraw
3. Display Balance
5. Exit
Choose an option: 1
Enter deposit amount: 1000
Deposited: 1000.0
Current Balance: 1000.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
5. Exit
Choose an option: 2
Enter withdrawal amount: 200
Withdrawn: 200.0
Service charge applied: 50.0
Current Balance: 750.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
5. Exit
Choose an option: 3
Current Balance: 750.0

Menu:
1. Deposit
2. Withdraw
3. Display Balance
5. Exit
Choose an option: 5
Exiting the program.
```

<u>Program 6</u>

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.

Algorithm:



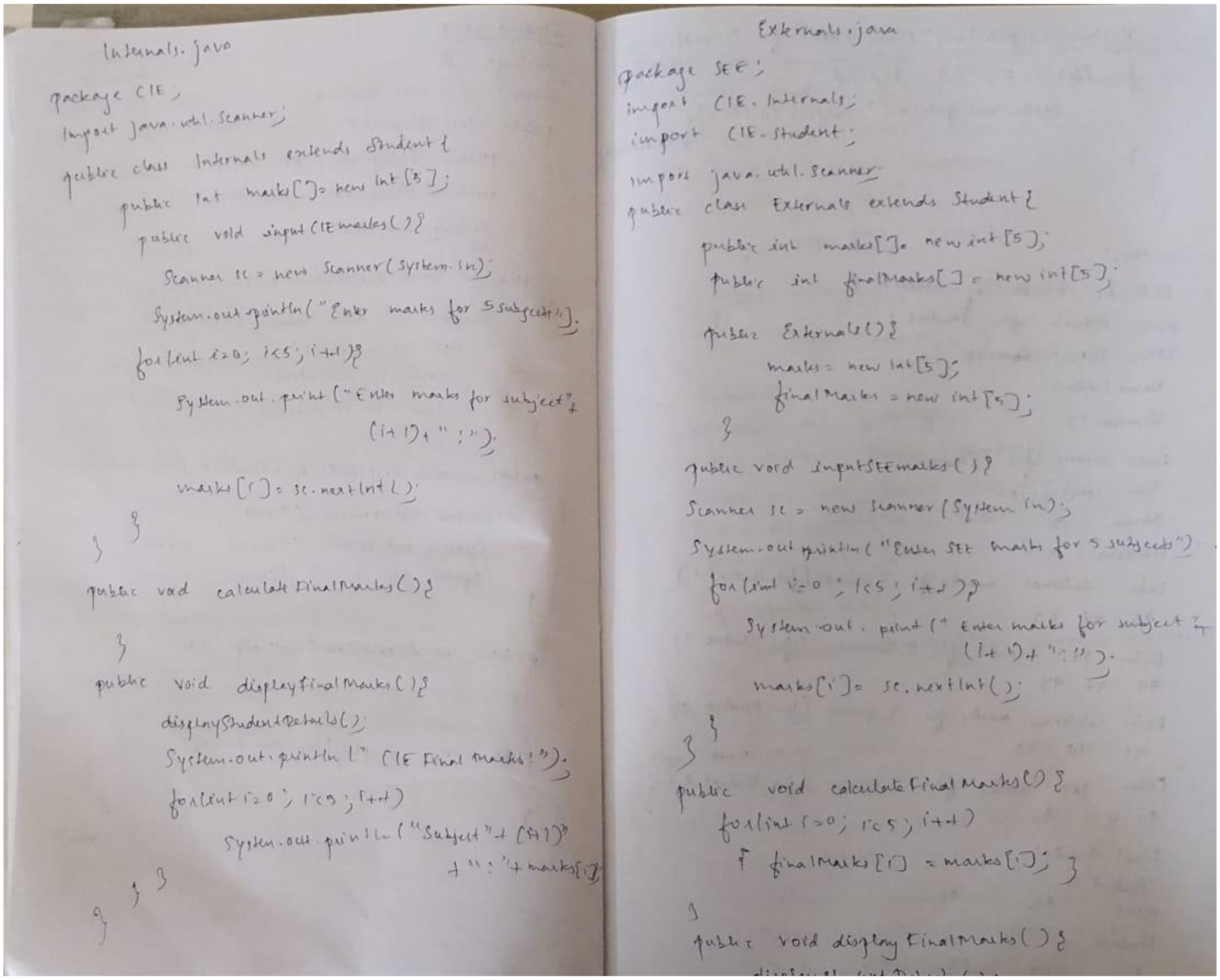
Main.java

```
import CIE. Internals;
Import SEE. ~~Internal~~ Externals;
Class Main{
 public static void main(String args[]){
      Internals internalStudent = new Internals();

      internalStudent.usn = "1BM23CS148";
      internal Student.name = "Keerthana H Bhat";
      internalStudent.sem = 3;

      internalStudent.inputCIEmarks();
      internal Student.calculateFinalMarks();
      internal Student. displayFinalMarks();

     Externals externalStudent = new Externals();

       externalStudent.usn = "1BM23CS148";
       external Student.name = "Keerthana H Bhat";
       externalStudent.sem = 3;

       external Student. inputSEEmarks();
       externalStudent. calculateFinalMarks();
       externalStudent. displayFinalMarks();
    }
 }
```

[illegible]

Student.java

```
package CIE;
import java.util.Scanner;
public class Student {
     public String usn = new String();
     public String name = new String();
     public int sem;

  public void inputStudentDetails()
 {    Scanner sc = new Scanner (System.in);
        usn = sc.nextLine();
        name = sc.nextLine();
        sem = sc.nextInt();
  }

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

  public void calculateFinalMarks(){

   }

  public void displayFinalMarks(){

   }
}
```



Internals.java

```
package CIE;
Import java.util.Scanner;
public class Internals extends Student {
    public int marks[]= new int [5];
      public void inputCIEmarks(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter marks for 5 subjects");
      for(int i=0; i<5; i++){
          System.out.print("Enter marks for subject"+
                              (i+1)+":");

          marks[i]= sc.nextInt();
      }
  }
public void calculateFinalMarks(){

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

Externals.java

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

      public Externals(){
            marks= new int[5];
            finalMarks = new int[5];
      }
    public void inputSEEmarks(){
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter SEE marks for 5 subjects")
      for(int i=0; i<5; i++){
          System.out.print("Enter marks for subject"+
                              (i+1)+":");
          marks[i]= sc.nextInt();
      }
    }
  public void calculateFinalMarks(){
      for(int i=0; i<5; i++)
        { finalMarks[i] = marks[i]; }
    }
    public void displayFinalMarks(){
```

```
    System.out.println(" Final Marks for 5 subjects");
    for (int i=0; i<5; i++){
            System.out.println(" Subject "+ (i+1)+ ": "+
                                                    finalMarks[i]
      }
   }
}
```

OUTPUT:-

```
Enter the number of students: 2
Enter details for Student 1
USN: 1BM23CS~~[illegible]~~148
 Name: ABC
 Semester: 3
Enter details for Student: 2
 USN: 1BM23CS150
 Name: XYZ
Semester: 3
Enter internal marks for 3 courses (for Student 1)
 45  46  47
Enter SEE Marks for 3 courses  (for Student 1)
95  96  97
Enter internal marks for 3 courses (for Student 2)
 41  42  43
Enter SEE marks for 3 courses (for Student 2)
90  91   92.
Final Marks:
 Student :1
92.5      94       95.5
Student :2
 86     87.5     89.
```

Code:
Main class:

```
import CIE.Internals;
import SEE.Externals;

class Main {

    public static void main(String args[]) {

        Internals internalStudent = new Internals();

       internalStudent.usn = "1BM23CS148";
        internalStudent.name= "Keerthana H Bhat";

        internalStudent.sem=3;

        internalStudent.inputCIEmarks();
        internalStudent.calculateFinalMarks();
        internalStudent.displayFinalMarks();

        Externals externalStudent = new Externals();

             externalStudent.usn = "1BM23CS148";
        externalStudent.name = "Keerthana H Bhat";

        externalStudent.sem=3;

        externalStudent.inputSEEmarks();
        externalStudent.calculateFinalMarks();
        externalStudent.displayFinalMarks();
    }
}
```

In CIE package:

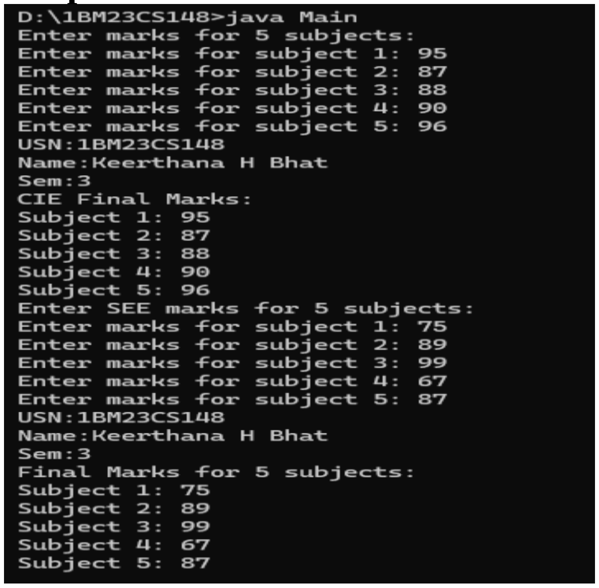
```
package CIE;

import java.util.Scanner;

public class Internals extends Student {

    public int marks[] = new int[5];

       public void inputCIEmarks() {

       Scanner sc = new Scanner(System.in);
       System.out.println("Enter marks for 5 subjects:");


        for (int i = 0; i < 5; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }

            }



            public void calculateFinalMarks() {
            }


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

      }
}
```

In SEE Package:

```
package SEE;
import CIE.Internals;
import CIE.Student;

import java.util.Scanner;

public class Externals extends Student {

   public int marks[] = new int[5];
   public int finalMarks[] = new int[5];


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

       public void inputSEEmarks() {

Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter SEE marks for 5 subjects:");

            for (int i = 0; i < 5; i++) {
          System.out.print("Enter marks for subject " + (i + 1) + ": ");
          marks[i] = sc.nextInt();
        }
    }

       public void calculateFinalMarks() {

        for (int i = 0; i < 5; i++) {
                finalMarks[i] = marks[i];        }
    }


    public void displayFinalMarks() {

        displayStudentDetails();

            System.out.println("Final Marks for 5 subjects:");

        for (int i = 0; i < 5; i++) {
          System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);  // Print final marks
        }
    }
}
```

Output:

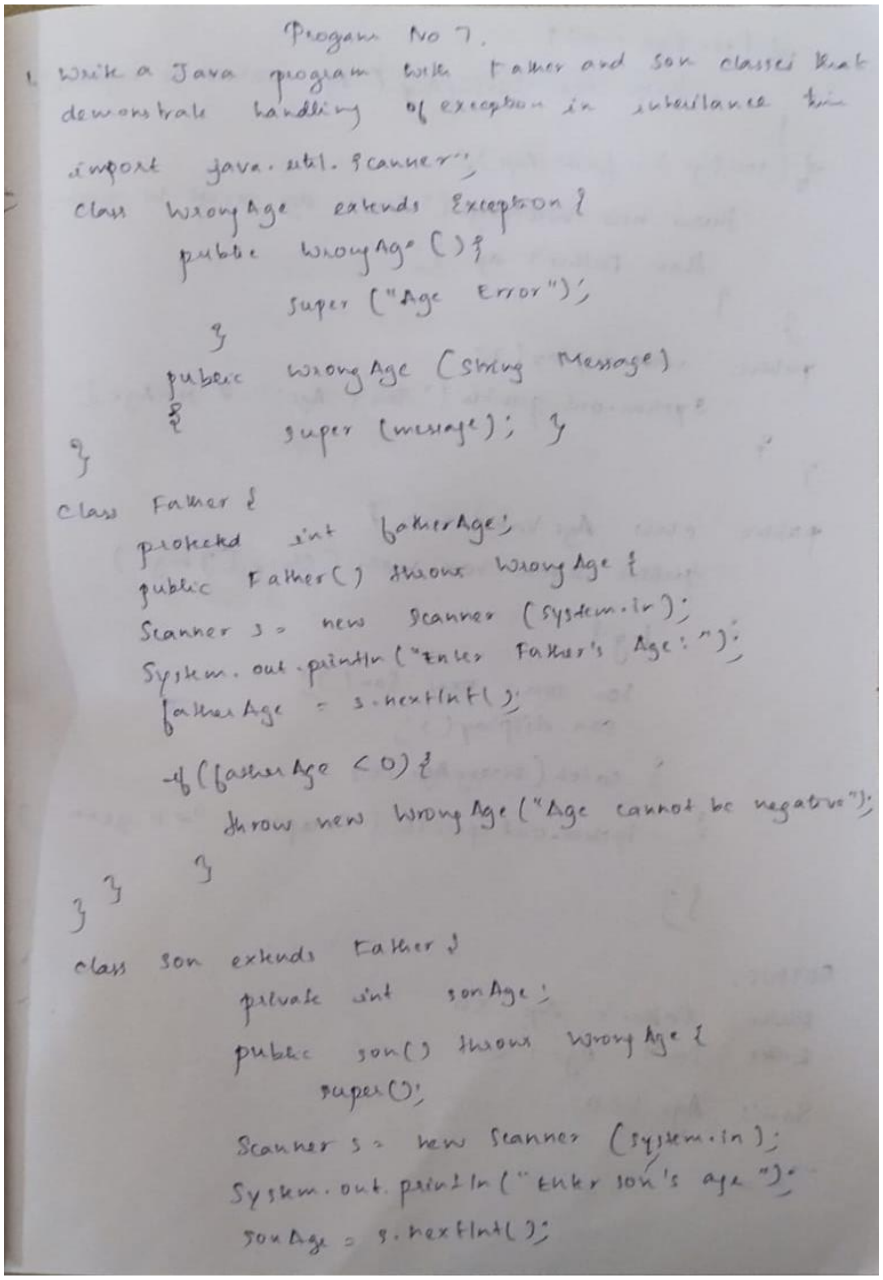


```
D:\1BM23CS148>java Main
Enter marks for 5 subjects:
Enter marks for subject 1: 95
Enter marks for subject 2: 87
Enter marks for subject 3: 88
Enter marks for subject 4: 90
Enter marks for subject 5: 96
USN:1BM23CS148
Name:Keerthana H Bhat
Sem:3
CIE Final Marks:
Subject 1: 95
Subject 2: 87
Subject 3: 88
Subject 4: 90
Subject 5: 96
Enter SEE marks for 5 subjects:
Enter marks for subject 1: 75
Enter marks for subject 2: 89
Enter marks for subject 3: 99
Enter marks for subject 4: 67
Enter marks for subject 5: 87
USN:1BM23CS148
Name:Keerthana H Bhat
Sem:3
Final Marks for 5 subjects:
Subject 1: 75
Subject 2: 89
Subject 3: 99
Subject 4: 67
Subject 5: 87
```

Program 7

Write a program that demonstrates handling of exceptions in inheritance trees. Create a base class called "Father" and a 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 causes both father and son's age and throws an exception if son's age is >=father's age.

Algorithm:



Program No 7.

1. Write a Java program with Father and Son classes that demonstrate handling of exception in inheritance tree

```
import java.util.Scanner;
class WrongAge extends Exception {
        public WrongAge () {
                super ("Age Error");
        }
        public WrongAge (String Message)
        {       super (message); }
}
class Father {
    protected int fatherAge;
    public Father() throws WrongAge {
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter Father's Age: ");
     fatherAge = s.nextInt();

     if (fatherAge < 0) {
        throw new WrongAge ("Age cannot be negative");
      }
 } }
class Son extends Father {
        private int sonAge;
        public son() throws WrongAge {
               super();
        Scanner s = new Scanner (System.in);
        System.out.println (" Enter son's age ");
        sonAge = s.nextInt();
 if (sonAge < 0){
        throw new WrongAge("Age cannot be negative")
 }
if (sonAge >= fatherAge){
     throw new WrongAge(" Son's age cannot be greater
       than Father's age");
     }
 }
public  void display(){
        System.out.println(" Son's Age:" + sonAge);
   }
}
public  class  Age Validation {
        public static void main (String [] args)
     {      try {
              Son son = new Son();
               son.display();
          } catch(WrongAge  e)
      {    System.out.println("Exception: "+ e.getM

      }}
```

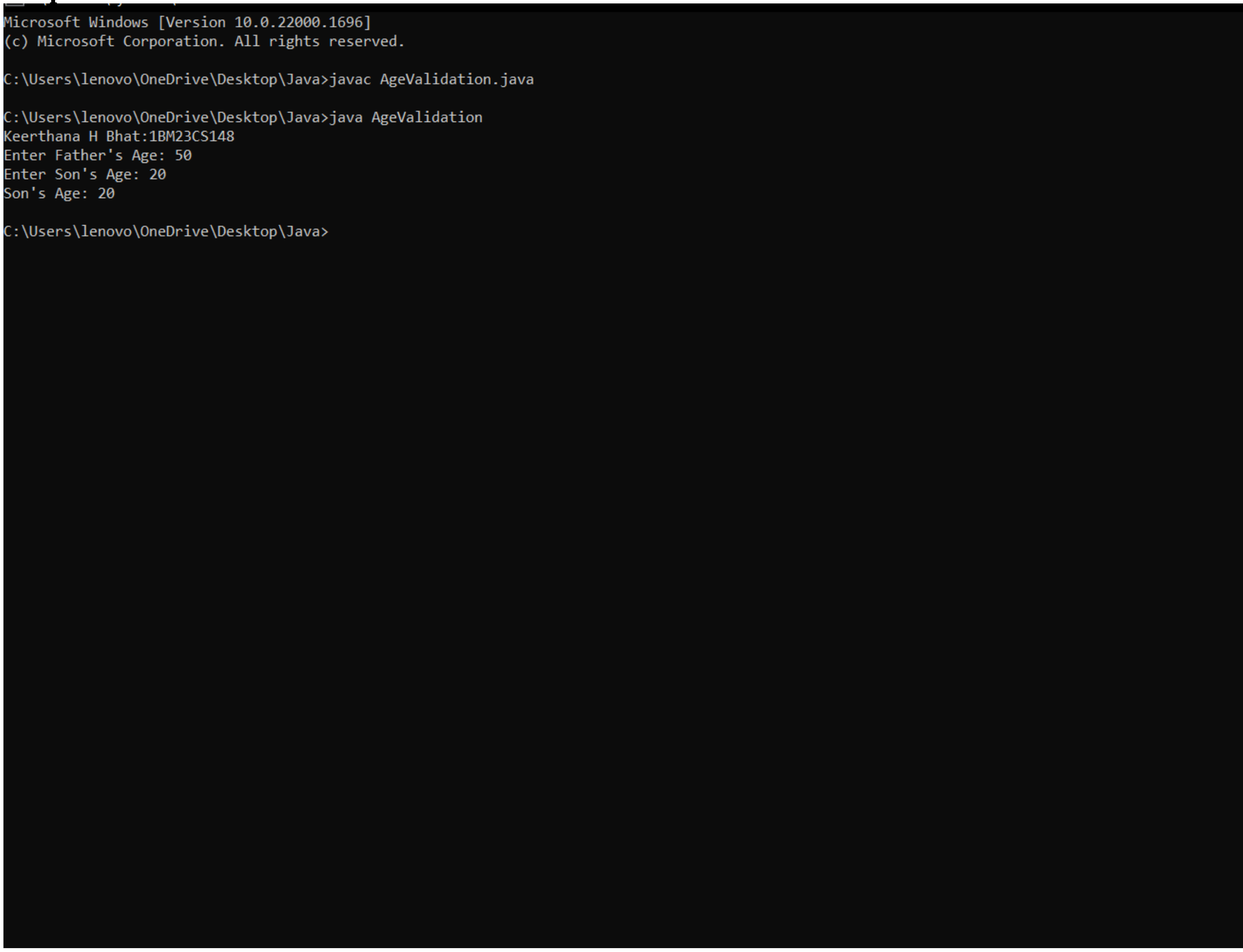
OUTPUT:

```
Enter  Father's Age : 50
Enter  Son's  Age : 20

Son's Age : 20.
```

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("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 {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();

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

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

public class ExceptionProgram {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println(e.getMessage());
        }
finally{
System.out.println("Name: Keerthana H Bhat");
System.out.println("USN: 1BM23CS148");
    }
}
}
```

Output:

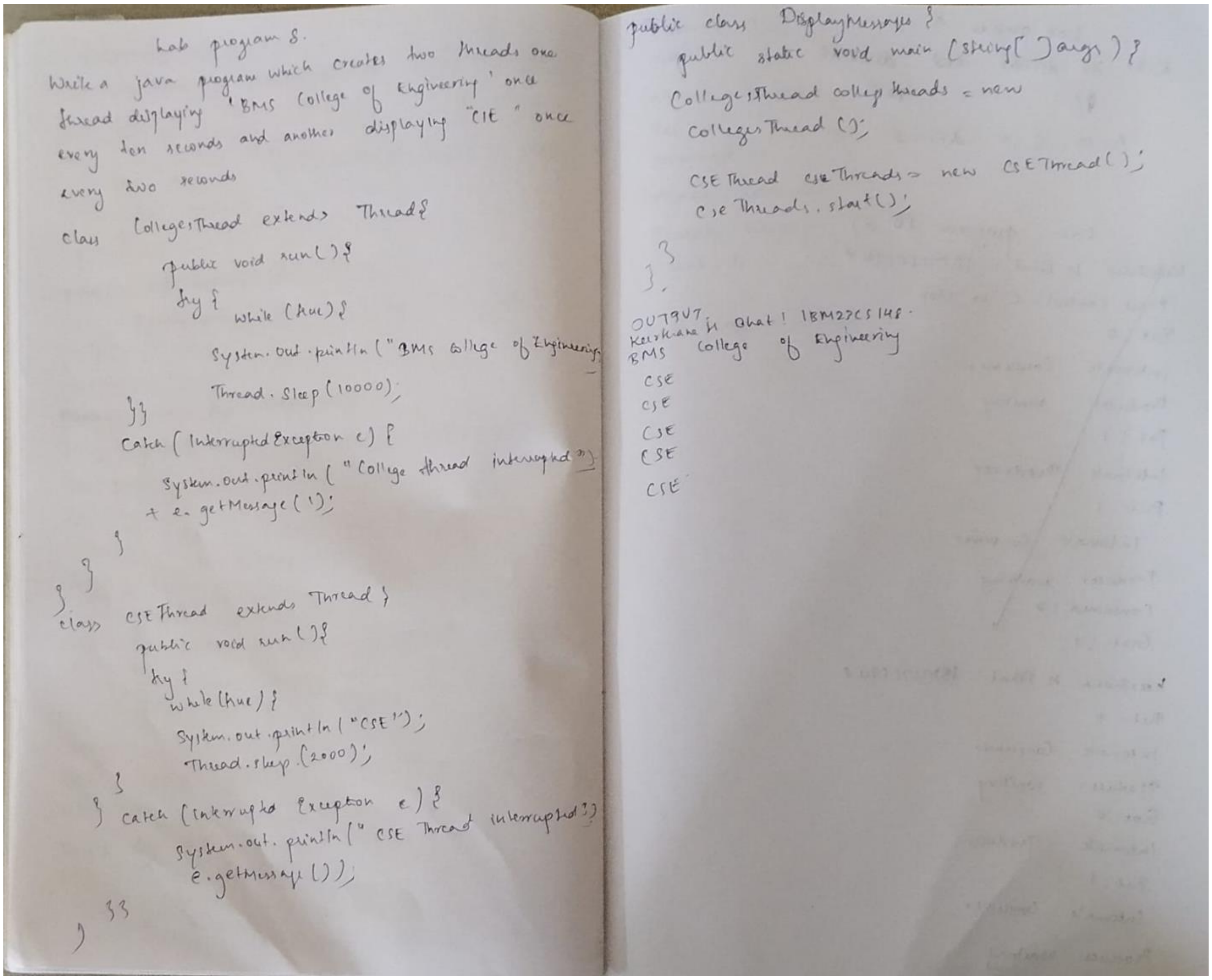


```
Microsoft Windows [Version 10.0.22000.1696]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo\OneDrive\Desktop\Java>javac AgeValidation.java

C:\Users\lenovo\OneDrive\Desktop\Java>java AgeValidation
Keerthana H Bhat:1BM23CS148
Enter Father's Age: 50
Enter Son's Age: 20
Son's Age: 20

C:\Users\lenovo\OneDrive\Desktop\Java>
```

Program 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.

Algorithm:



Lab program 8.

Write a java program which creates two threads one thread displaying 'BMS College of Engineering' once every ten seconds and another displaying "CIE" once every two seconds

```
class CollegesThread extends Thread{
    public void run(){
     try {
         while (true){
            System.out.println("BMS college of Engineering
            Thread.Sleep(10000);
    }}
    Catch (InterruptedException e) {
       System.out.println ("College thread interrupted")
     + e.getMessage());
    }
  }
}
class CSEThread extends Thread }
    public void run(){
     try {
       while(true){
         System.out.println("CSE");
         Thread.sleep.(2000);
     }
   } catch (interrupted Exception e){
         System.out.println(" CSE Thread interrupted")
          e.getMessage());
    }}
)
```

```
public class DisplayMessages {
   public static void main (String[ ]args){
   CollegesThread college threads = new
    CollegesThread ();

    CSE Thread cseThreads = new CSEThread();
    cse Threads.start();

  }
}.
```

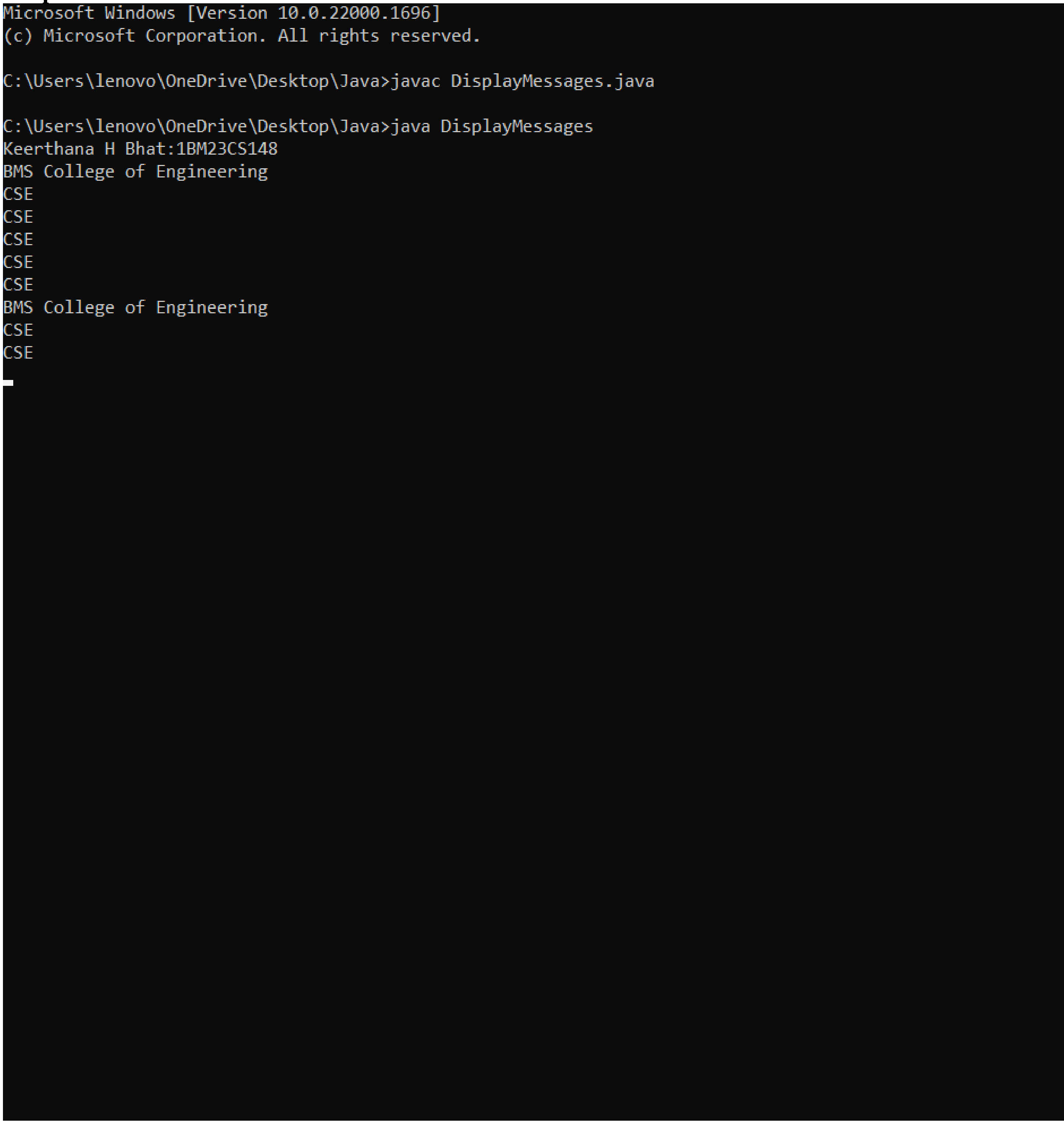
OUTPUT Hi what! 1BM22CS148. Keerthana

```
BMS College of Engineering
 CSE
 CSE
 CSE
 CSE
 CSE
```

Code:

```
class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted: " + e.getMessage());
        }
    }
}

// Thread to display "CSE" every 2 seconds
class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted: " + e.getMessage());
        }
    }
}

// Main class to run the threads
public class DisplayMessages {
    public static void main(String[] args) {
        // Create threads
            printf("Keerthana H Bhat:1BM23CS148");
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

        // Start threads
        collegeThread.start();
        cseThread.start();
    }
}
```

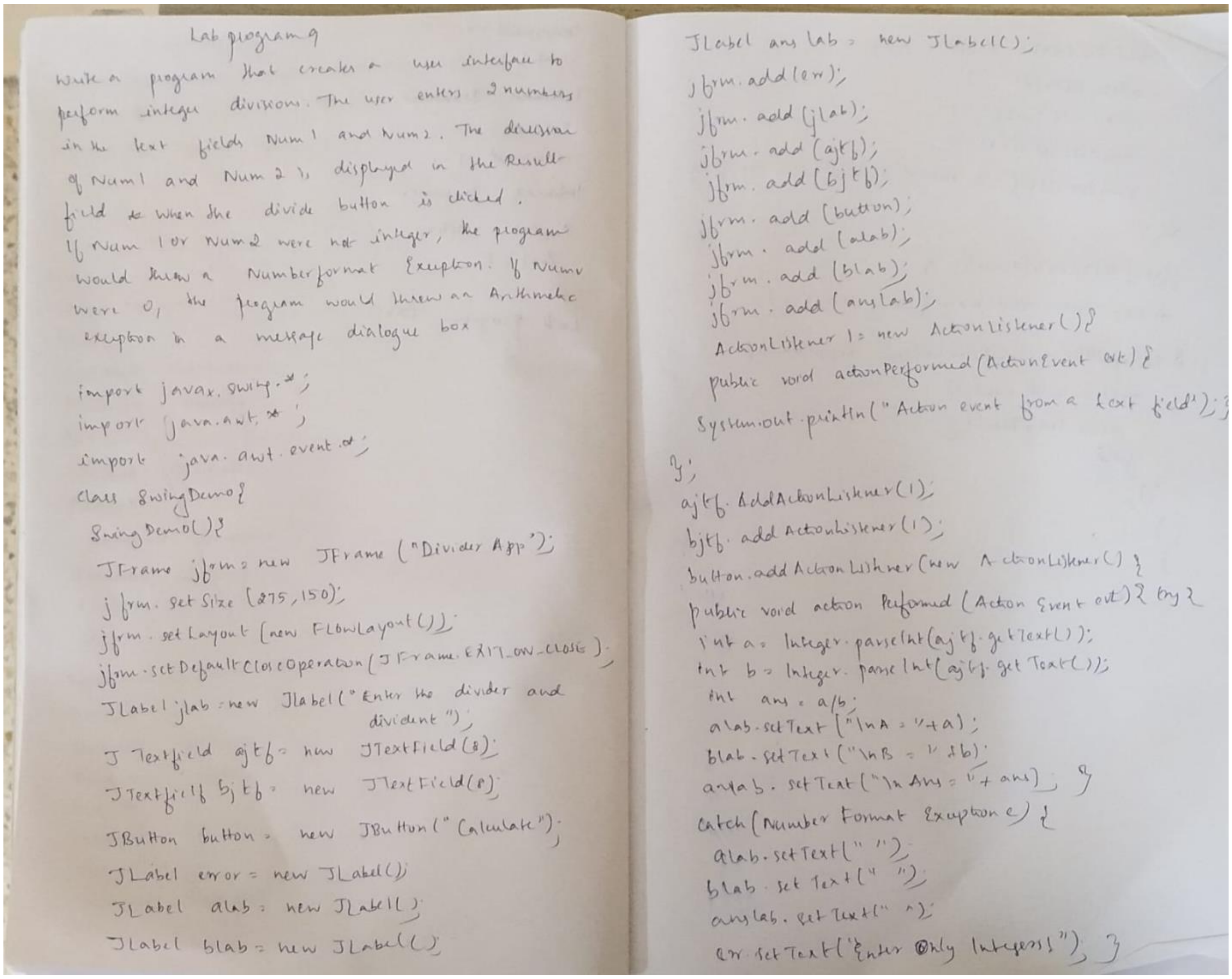
Output:



```
Microsoft Windows [Version 10.0.22000.1696]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo\OneDrive\Desktop\Java>javac DisplayMessages.java

C:\Users\lenovo\OneDrive\Desktop\Java>java DisplayMessages
Keerthana H Bhat:1BM23CS148
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

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

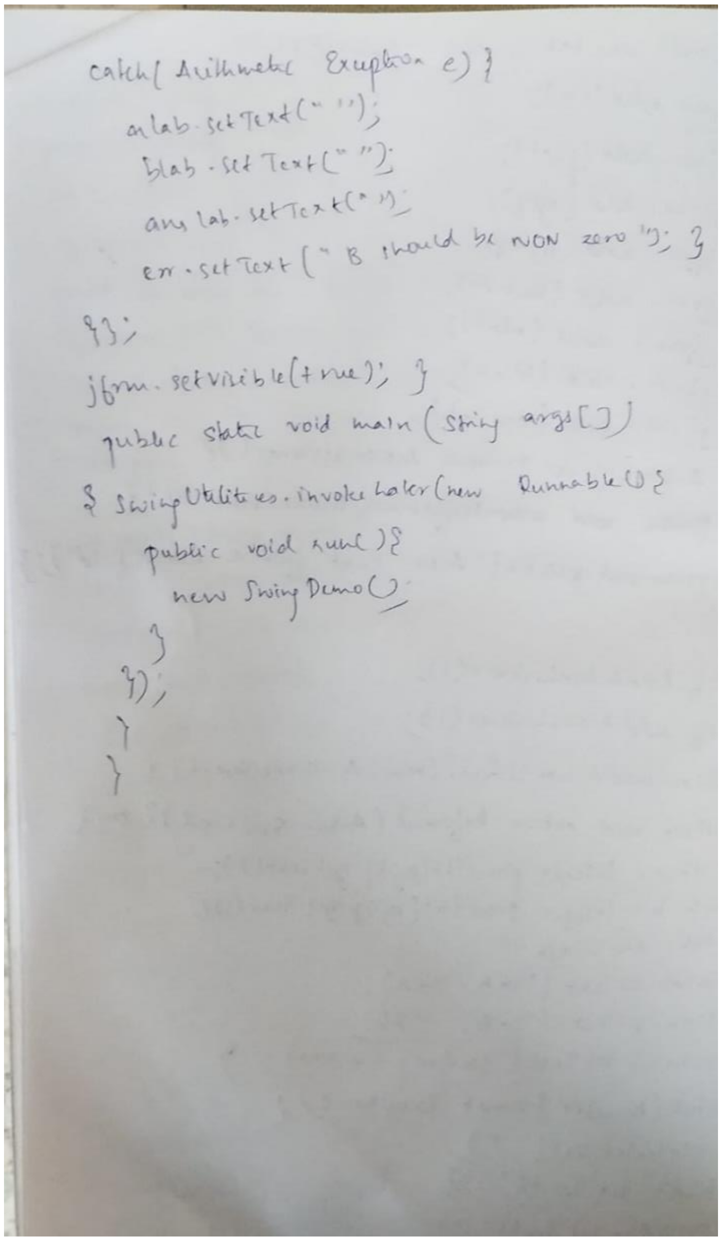
Algorithm:



Lab program 9

Write a program that creates a user interface to perform integer divisions. The user enters 2 numbers in the text fields Num 1 and Num 2. The division of Num1 and Num 2 is displayed in the Result field when the divide button is clicked. If Num 1 or Num 2 were not integer, the program would throw a Numberformat Exception. If Num2 were 0, the program would threw an Arithmetic exception in a message dialogue box

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
 SwingDemo(){
  JFrame jfrm = new JFrame ("Divider App");
  jfrm. set Size (275,150);
  jfrm. setLayout (new FLowLayout());
  jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  JLabel jlab = new Jlabel("Enter the divider and
                                  divident");
  J Textfield ajtf = new JTextField(8);
  JTextfielf bjtf = new JTextField(8);
  JButton button = new JButton("Calculate");
  JLabel error = new JLabel();
  JLabel alab = new JLabel();
  JLabel blab = new JLabel();
JLabel anslab = new JLabel();
jfrm.add(err);
 jfrm. add (jlab);
 jfrm. add (ajtf);
 jfrm. add (bjtf);
 jfrm. add (button);
 jfrm. add (alab);
 jfrm. add (blab);
 jfrm. add (anslab);
  ActionListener l = new ActionListener(){
  public void actionPerformed(ActionEvent evt) {
 System.out.println("Action event from a text field"); }
};
ajtf. AddActionListner(l);
bjtf. add ActionListener(l);
button.add ActionListner(new ActionListner() {
public void action Performed (Action Event evt) { try {
 int a = Integer.parseInt(ajtf.getText());
 int b = Integer. parseInt(ajtf. get Text());
  int  ans = a/b;
  alab.setText ("\nA = "+a);
  blab.setText ("\nB = "+b);
 anslab. set Text ("\n Ans = "+ ans); }
catch(Number Format Exception e) {
 alab.setText(" ");
 blab. set Text(" ");
 anslab. set Text(" ");
 err.setText("Enter Only Integers!"); }
```



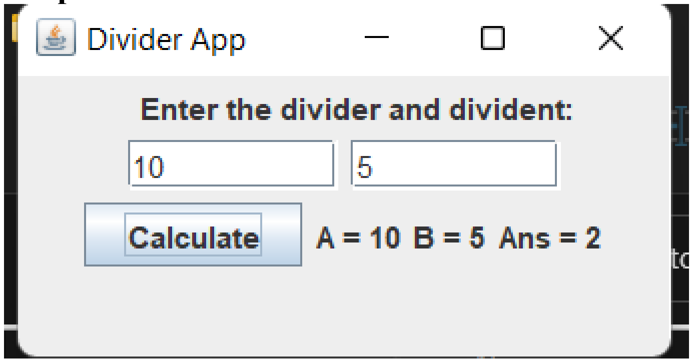
```
catch( Arithmetic Exception e) {
   alab.setText(" ");
    blab.setText(" ");
    anslab.setText(" ");
    err.setText(" B should be NON zero "); }
}};
jfrm.setVisible(true); }
 public static void main(String args[])
{ SwingUtilities.invokeLater(new Runnable(){
     public void run(){
       new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

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(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();

JLabel blab = new JLabel();
JLabel anslab = new JLabel();
// add in order :)
jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field"); }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) { try{
int a = Integer.parseInt(ajtf.getText()); int b =
Integer.parseInt(bjtf.getText()); int ans = a/b;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!"); }
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!"); }
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){ // create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
}
```

Output:



Divider App

Enter the divider and divident:

10

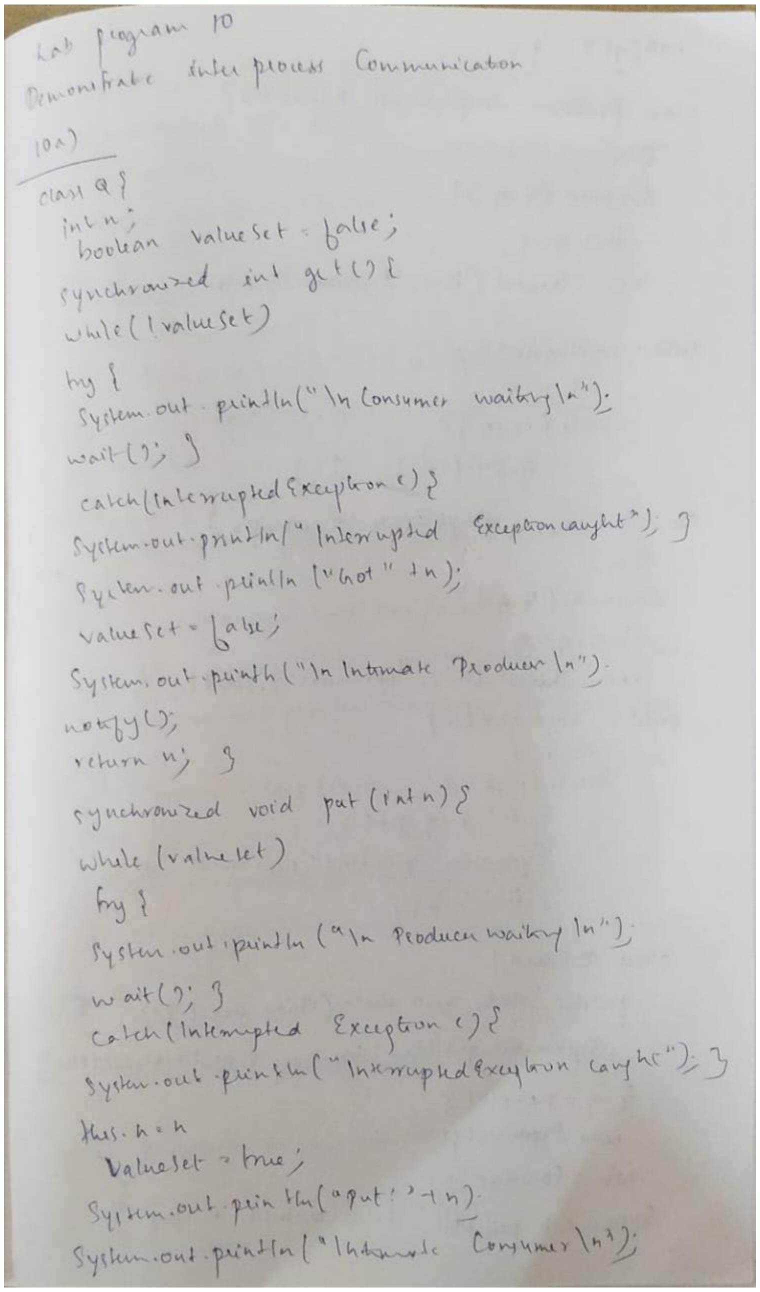
5

A = 10 B = 5 Ans = 2

Program 10(a)

Demonstrate Inter process Communication

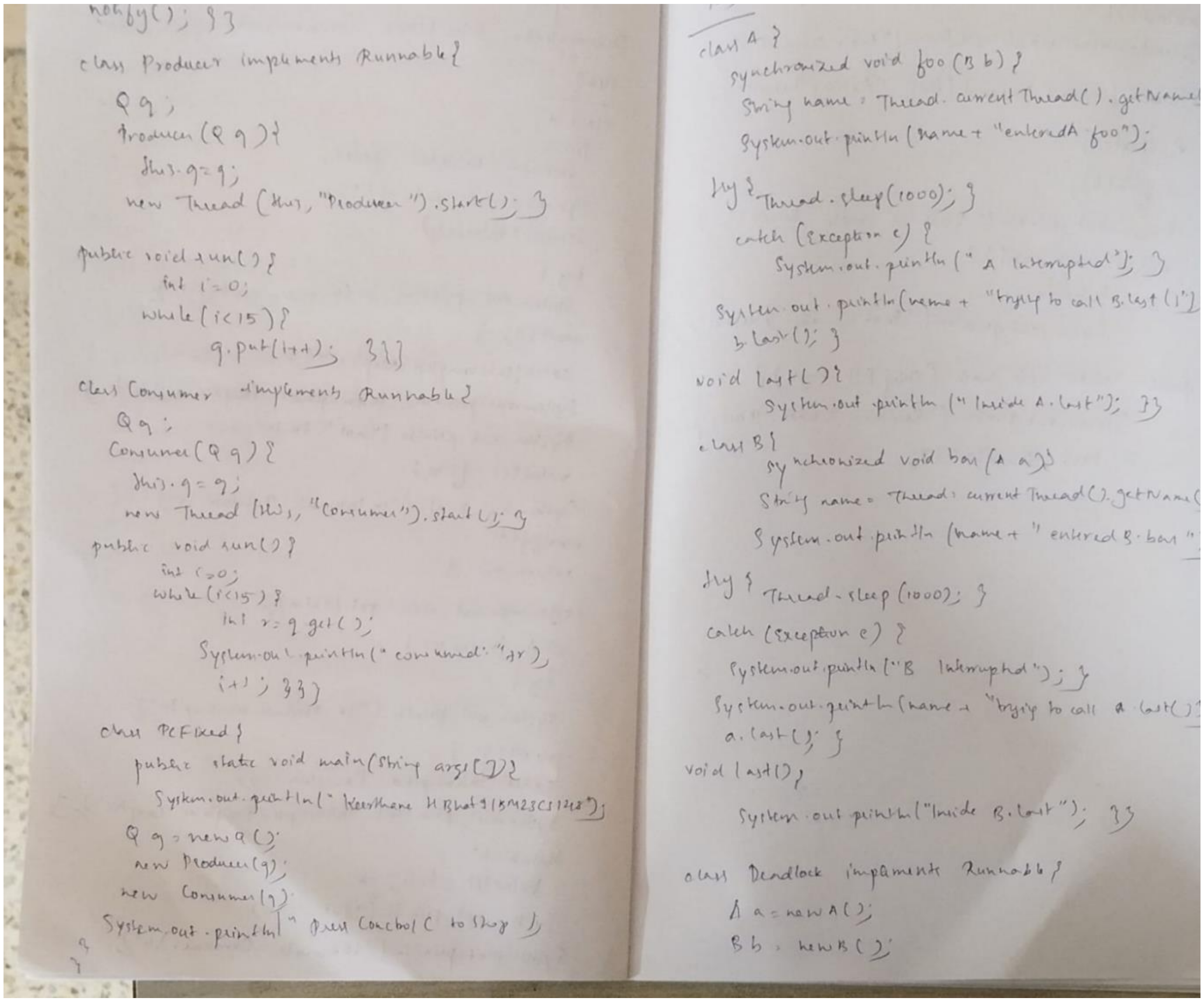
Algorithm:



Lab program 10
Demonstrate inter process Communication

10a)

```
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(" Interrupted Exception 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(Interrupted Exception e) {
  System.out.println("InterruptedException caught"); }
  this.n = n
    ValueSet = true;
   System.out.println("put!" + n).
  System.out.println("Intimate 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("Keerthana H Bhat 1BM23CS148");
    Q q=new Q();
    new Producer(q);
   new Consumer(q);
  System.out.println(" Press Control C to stop");
 }
}
```

```
class A {
  synchronized void foo(B b) {
   String name = Thread.currentThread().getName
   System.out.println(name+ "enteredA 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 B.last"); }}

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

Code:
class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while(!valueSet)

try {

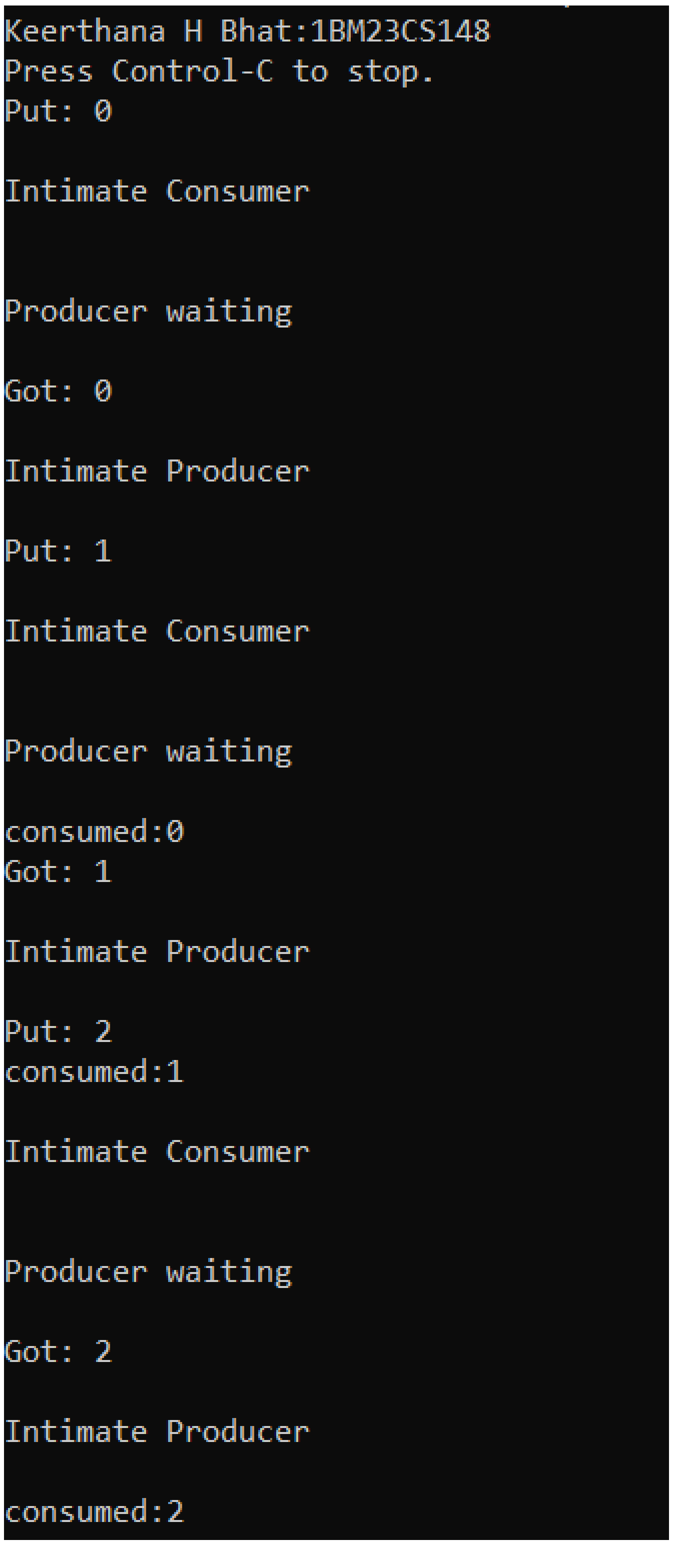
System.out.println("\nConsumer waiting\n");

```
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedExceptioncaught");
}
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("Keerthana H Bhat:1BM23CS148");
Q q = new Q();

new Producer(q);

new Consumer(q);

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

}

}
```

Output:

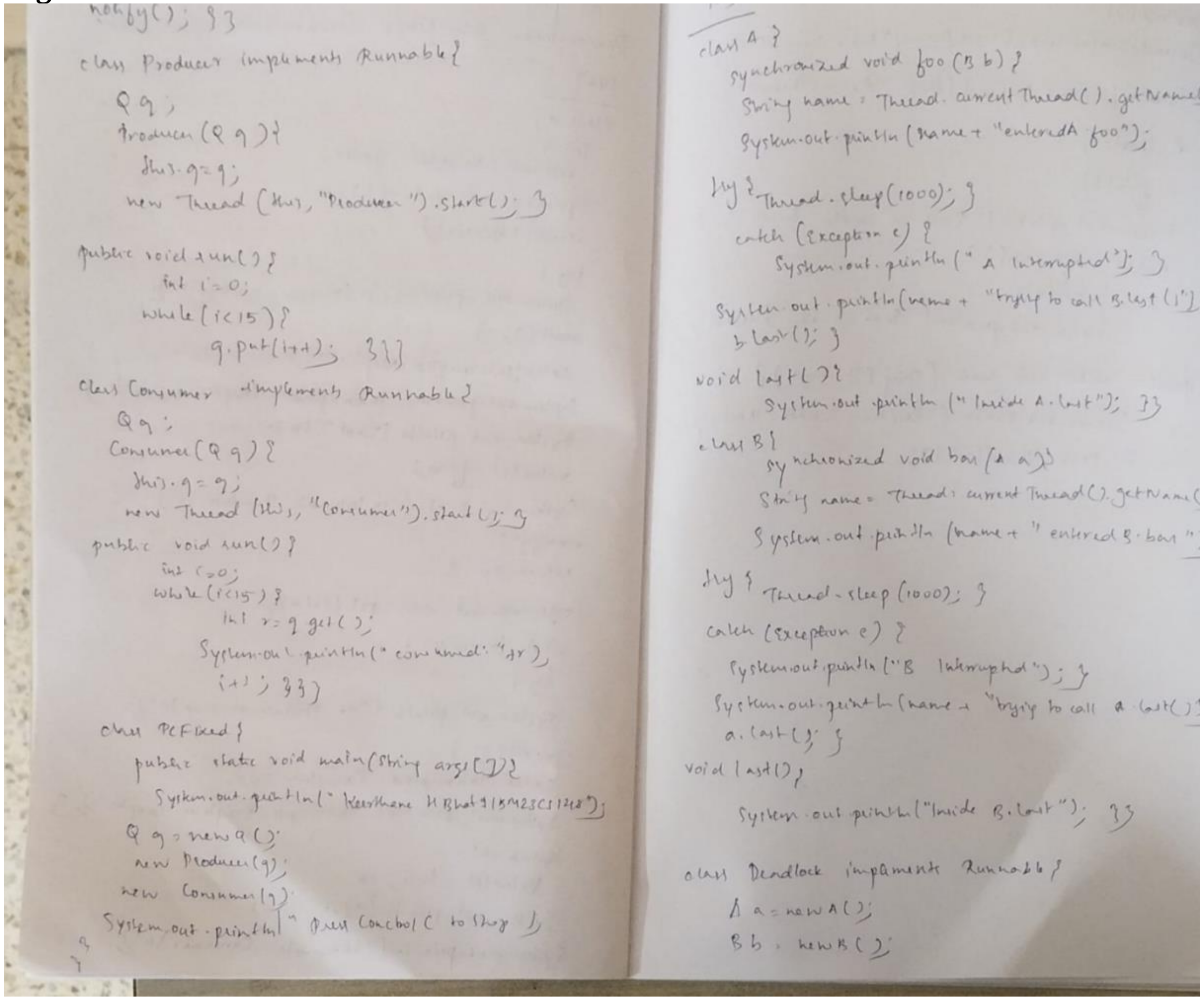


```
Keerthana H Bhat:1BM23CS148
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer

Put: 2
consumed:1

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

consumed:2
```

Program 10(b)

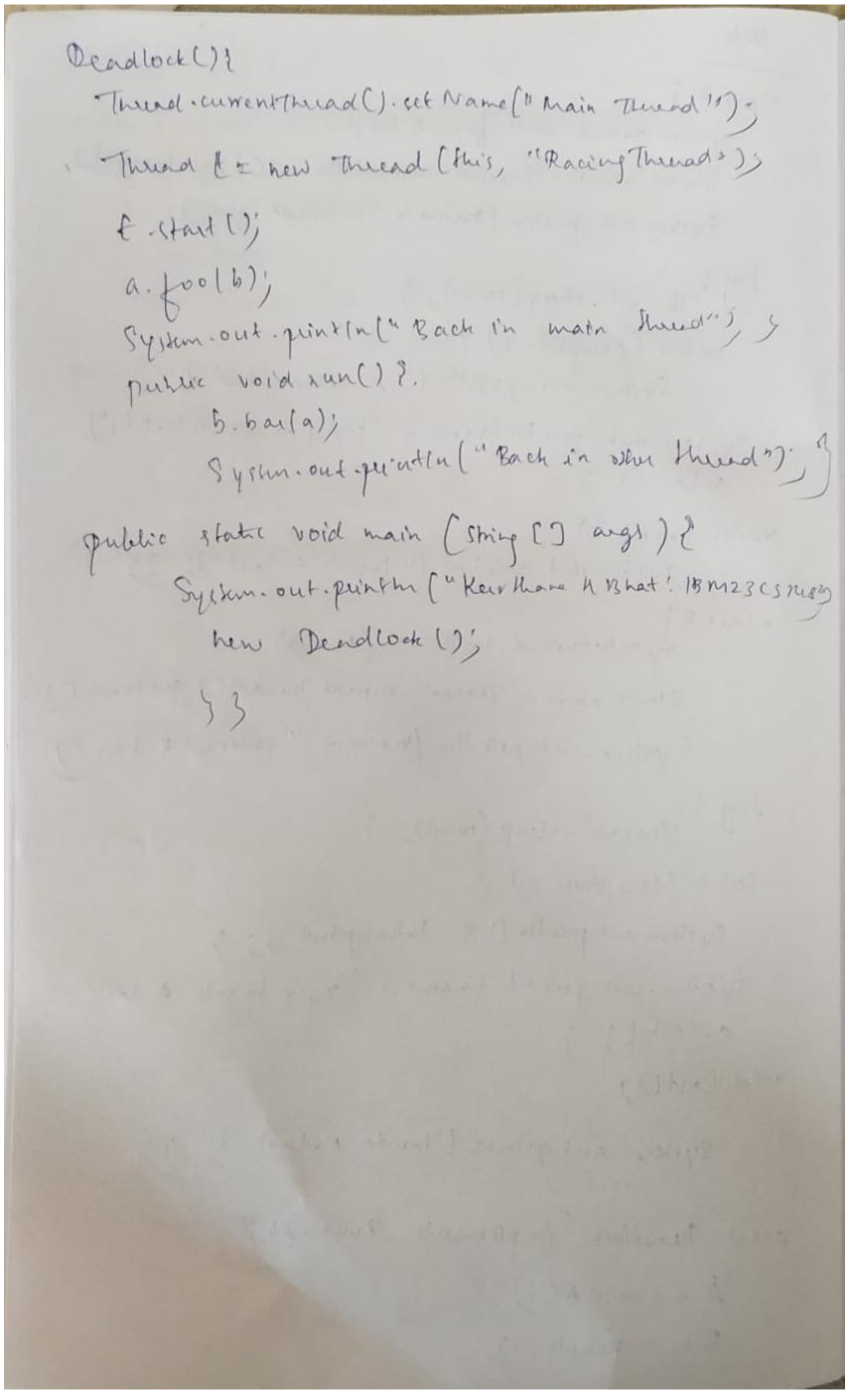
Demonstrate Deadlock

Algorithm:



```
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(" Keerthana H Bhat 1BM23CS128");
  Q q=new Q();
   new Producer(q);
  new Consumer(q);
 System.out.println(" Press Control C to Stop");
}
}
```

```
class A {
  synchronized void foo(B b){
   String name = Thread.currentThread().getName
   System.out.println(name+ "enteredA 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 B.last"); }}

class Deadlock impaments Runnable{
   A a=new A();
   B b= new B();
```



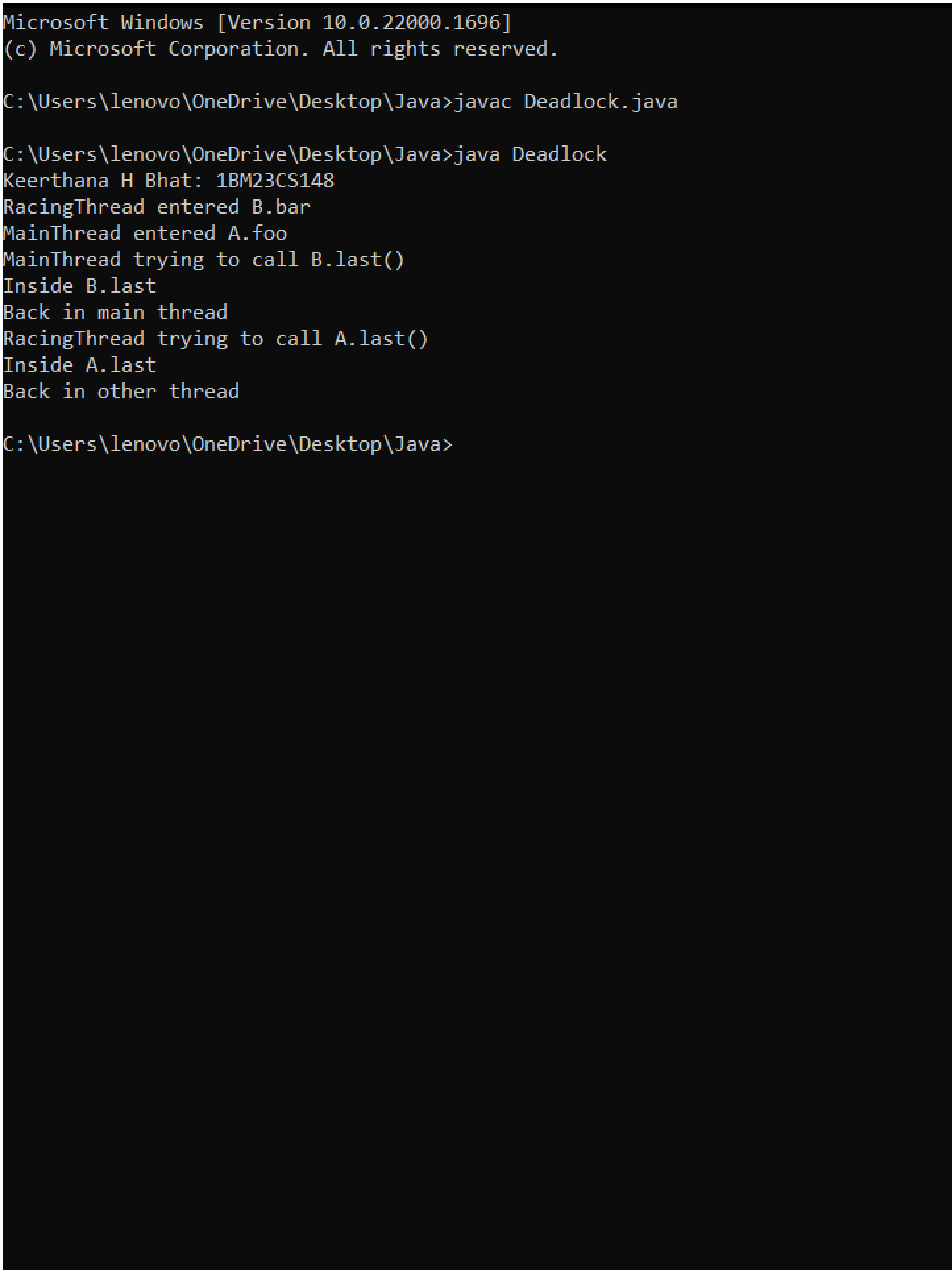
```
Deadlock(){
 Thread.currentThread().setName("Main Thread");
 Thread t = new Thread(this, "Racing Thread");
  t.start();
  a.foo(b);
  System.out.println("Back in main thread"); }
  public void run(){.
       b.bar(a);
       System.out.println("Back in other thread"); }
public static void main (String [] args) {
    System.out.println("Keerthana K Bhat 1BM23CS1489");
     new Deadlock();
    }}
```

Code:

```
class A {

    // Synchronized method in class A
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");

        try {
            Thread.sleep(1000); // Simulate some work
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }

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

    // Non-synchronized method in class A
    void last() {
        System.out.println("Inside A.last");
    }
}

class B {

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

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

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

    // Non-synchronized method in class B
    void last() {
        System.out.println("Inside B.last");
    }
}
```

```
class Deadlock implements Runnable {

    A a = new A();
    B b = new B();

    Deadlock() {
        // Set the name for the main thread
        Thread.currentThread().setName("MainThread");

        // Create and start a new thread
        Thread t = new Thread(this, "RacingThread");
        t.start();

        // Main thread calls A.foo and locks object 'a'
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        // RacingThread calls B.bar and locks object 'b'
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String[] args) {
        System.out.println("Keerthana H Bhat: 1BM23CS148");
        new Deadlock(); // Trigger the deadlock scenario
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.22000.1696]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo\OneDrive\Desktop\Java>javac Deadlock.java

C:\Users\lenovo\OneDrive\Desktop\Java>java Deadlock
Keerthana H Bhat: 1BM23CS148
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
Inside B.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread

C:\Users\lenovo\OneDrive\Desktop\Java>
```