VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

KATTA CHAITANYA KRISHNA SAI (1BM23CS144)

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 **KATTA CHAITANYA KRISHNA SAI(1BM23CS144),** 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.

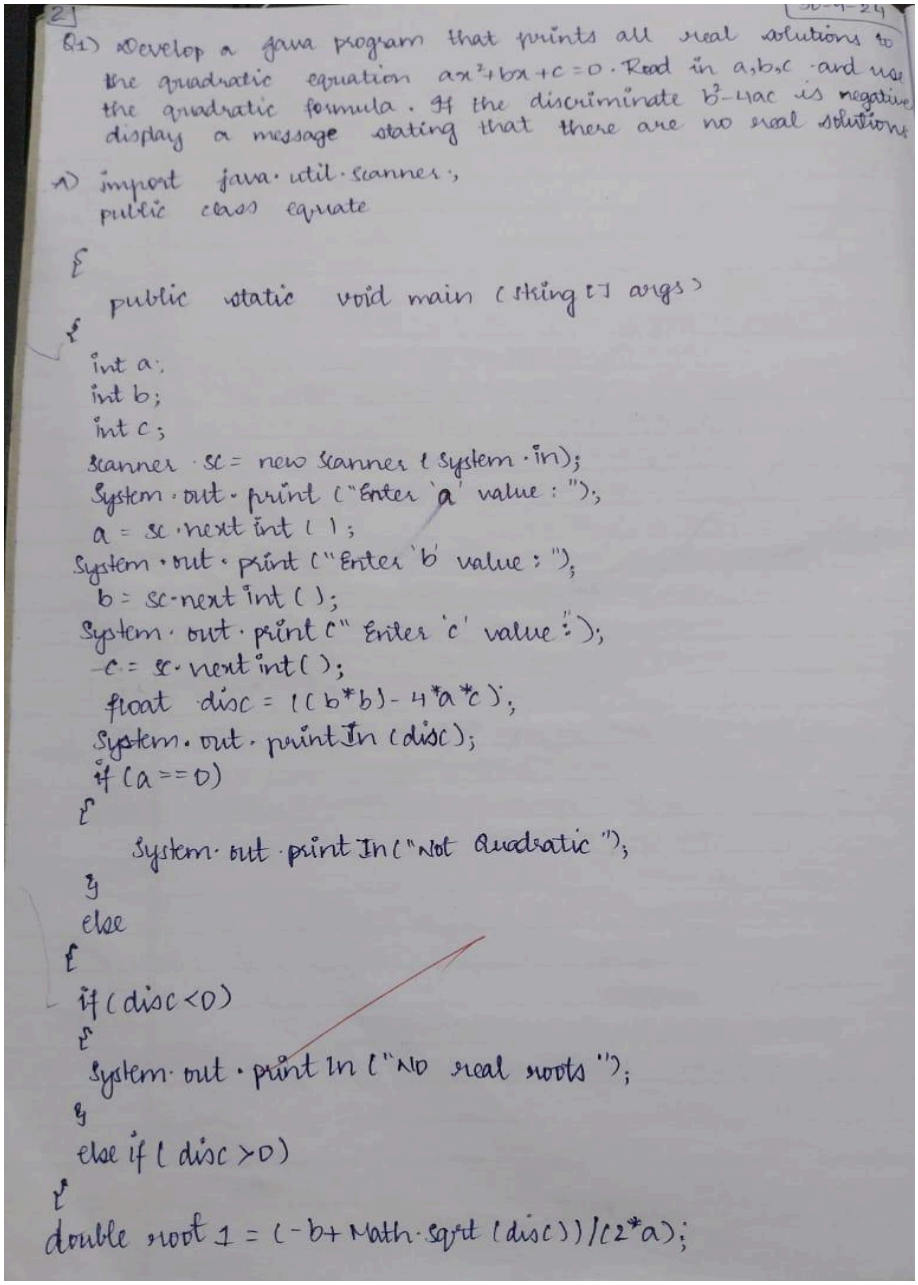
SURABHI S Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/chaitanya-cs23/java-lab-programs-ooj

Program 1
Implement Quadratic Equation

Algorithm:



```
2)
Q1) Develop a java program that prints all real solutions to
the quadratic equation ax²+bx+c=0. Read in a,b,c and use
the quadratic formula. If the discriminate b²-4ac is negative
display a message stating that there are no real solutions.

A) import java.util.Scanner;
   public class equate
{
   public static void main (String[] args)
{
  int a;
  int b;
  int c;
  Scanner sc = new Scanner (System.in);
  System.out.print ("Enter 'a' value : ");
  a = sc.nextint ();
 System.out.print ("Enter 'b' value : ");
  b = sc.nextint ();
 System.out.print (" Enter 'c' value : ");
  c = sc.nextint();
  float disc = ((b*b)-4*a*c);
  System.out.println (disc);
  if (a==0)
{
    System.out.println ("Not Quadratic ");
}
  else
{
  if (disc<0)
{
  System.out.println ("No real roots ");
}
  else if (disc>0)
{
double root1 = (-b+Math.sqrt (disc))/(2*a);
```

```
double root2 = (-b - Math.sqrt(disc))/(2*a);
    System.out.println("Real roots");
    System.out.println("Roots-1:"+roots1);
    System.out.println("Root-2:"+roots2);
    }
    else
    {
    double root1 = (-b)/(2*a);
System.out.println("Real and equal");
    System.out.println("Roots-1:"+roots1);
    System.out.println("Roots-2:"+root1);
    }
    }
}
```

OUTPUT:

```
Enter 'a' value : 4
Enter 'b' value : 4
Enter 'c' value : 1
0.0
Real and equal
Root-1 : 0.0
Root-2 : 0.0
```

```
Enter 'a' value : 1
Enter 'b' value : 1
Enter 'c' value : 1
-3.0
No real roots
```

```
Enter 'a' value : 0
Enter 'b' value : 1
Enter 'c' value : 2
1.0
Not Quadratic
```

```
Enter 'a' value : 3
Enter 'b' value : 8
Enter 'c' value : 2
40.0
Real roots
Root-1 : -0.27924077994138735
Root-2 : -2.3874258867227933
```

01.10

Code:

```
import java.util.Scanner;
public class Quadratic

{
   public static void main(String[] args)
   {
      int a;
      int b;
      int c;
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter 'a' value: ");
      a= sc.nextInt();
      System.out.print("Enter 'b' value: ");
      b=sc.nextInt();
      System.out.print("Enter 'c' value: ");
      c=sc.nextInt();
      float disc = ((b*b)-4*a*c);
      System.out.println(disc);
      if (a==0)
      {
         System.out.println("Not Quadratic");
      }
      else
      {
         if (disc<0)
         {
         System.out.println("No real roots ");
         }
         else if (disc>0)
         {
         double root1= (-b + Math.sqrt(disc))/(2*a);
         double root2= (-b - Math.sqrt(disc))/(2*a);
         System.out.println("Real roots ");
         System.out.println("Root-1: "+root1);
         System.out.println("Root-2: "+root2);
         }
         else
         {
         double root1=(-b)/(2*a);
            System.out.println("Real and equal");
         System.out.println("Root-1: "+root1);
         System.out.println("Root-2: "+root1);
         }
         System.out.println("k.chaitanya");
         System.out.println("1BM23CS144");
```

```
        }
    }
}
```

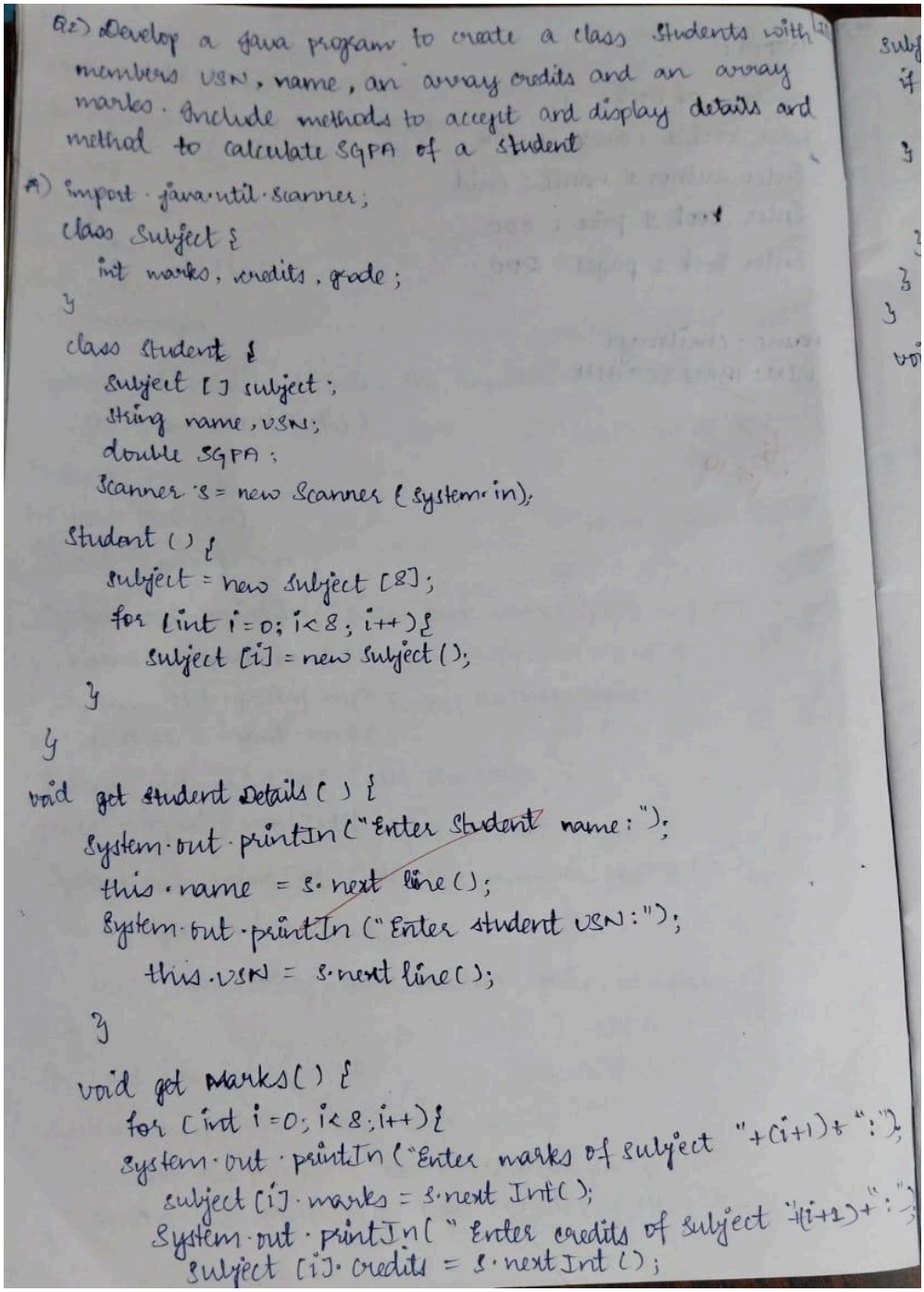
```
D:\CS144>java equate
Enter 'a' value: 4
Enter 'b' value: 4
Enter 'c' value: 1
0.0
Real and equal
Root-1: 0.0
Root-2: 0.0
k.chaitanya
1BM23CS144
```

```
D:\CS144>java equate
Enter 'a' value: 1
Enter 'b' value: 1
Enter 'c' value: 1
-3.0
No real roots
k.chaitanya
1BM23CS144

D:\CS144>java equate
Enter 'a' value: 0
Enter 'b' value: 1
Enter 'c' value: 2
1.0
Not Quadratic
```

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

D:\CS144>javac equate.java

D:\CS144>java equate
Enter 'a' value: 3
Enter 'b' value: 8
Enter 'c' value: 2
40.0
Real roots
Root-1: -0.2792407799438735
Root-2: -2.3874258867227933
k.chaitanya
1BM23CS144

D:\CS144>
```

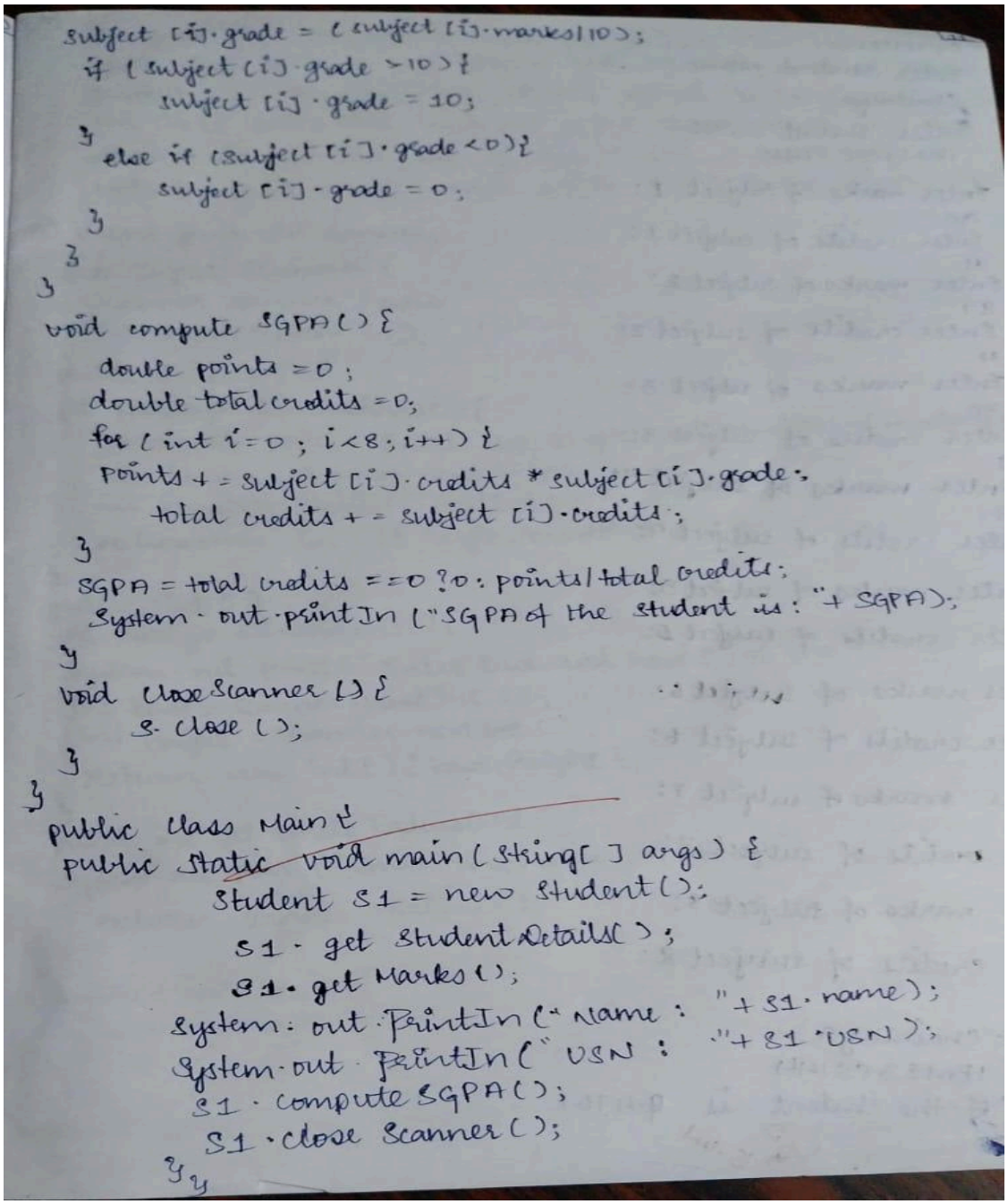
Program 2
SGPA Calculator

Algorithm:

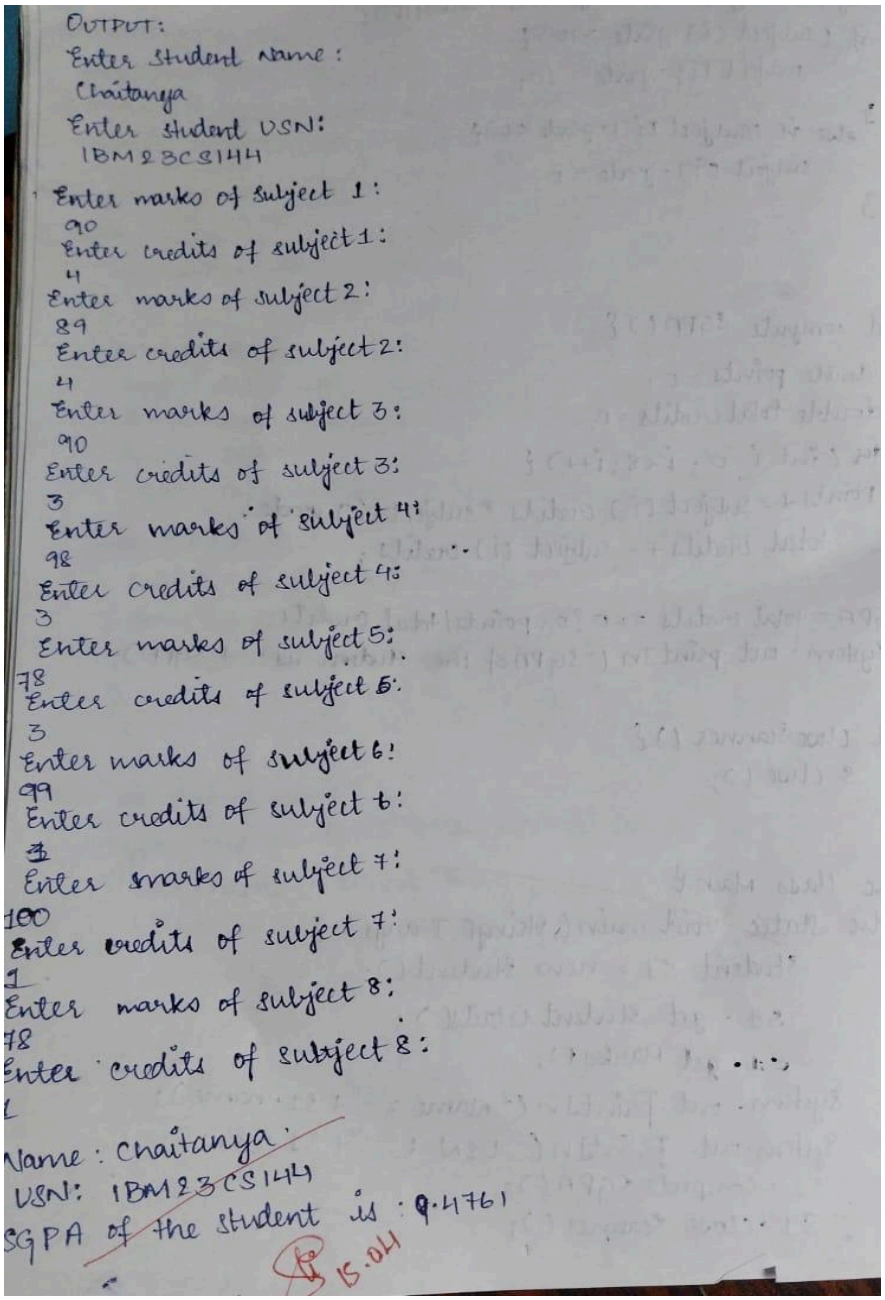


Q2) Develop a java program to create a class Students with members USN, name, an array credits and an array marks. Include methods to accept and display details and method to calculate SGPA of a student

```
import java.util.Scanner;
class Subject {
    int marks, credits, grade;
}
class Student {
    Subject [] subject;
    String name, USN;
    double SGPA;
    Scanner s = new Scanner (System.in);
    Student () {
        subject = new Subject [8];
        for (int i=0; i<8; i++){
            subject [i] = new Subject ();
        }
    }
void getStudentDetails () {
    System.out.println("Enter Student name:");
    this.name = s.nextLine();
    System.out.println("Enter student USN:");
        this.USN = s.nextLine();
    }
    void getMarks() {
        for (int i=0; i<8; i++){
        System.out.println("Enter marks of subject "+(i+1)+":");
            subject [i].marks = s.nextInt();
        System.out.println("Enter credits of subject "+(i+1)+":");
            subject [i].credits = s.nextInt();
```



```
        subject[i].grade = (subject[i].marks/10);
        if (subject[i].grade > 10){
            subject[i].grade = 10;
        } else if (subject[i].grade < 0){
            subject[i].grade = 0;
        }
    }
}
void computeSGPA(){
    double points = 0;
    double totalcredits = 0;
    for (int i = 0; i < 8; i++) {
    points += subject[i].credits * subject[i].grade;
        totalcredits += subject[i].credits;
    }
    SGPA = totalcredits == 0 ? 0 : points/totalcredits;
    System.out.println("SGPA of the student is: " + SGPA);
}
void closeScanner(){
    s.close();
}
}
public class Main{
public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
    System.out.println("Name : " + s1.name);
    System.out.println("USN : " + s1.USN);
    s1.computeSGPA();
    s1.closeScanner();
}}
```



OUTPUT:

Enter student name:
Chaitanya
Enter student USN:
1BM23CS144
Enter marks of subject 1:
90
Enter credits of subject 1:
4
Enter marks of subject 2:
89
Enter credits of subject 2:
4
Enter marks of subject 3:
90
Enter credits of subject 3:
3
Enter marks of subject 4:
98
Enter credits of subject 4:
3
Enter marks of subject 5:
78
Enter credits of subject 5:
3
Enter marks of subject 6:
99
Enter credits of subject 6:
3
Enter marks of subject 7:
100
Enter credits of subject 7:
1
Enter marks of subject 8:
78
Enter credits of subject 8:
1

Name: Chaitanya
USN: 1BM23CS144
SGPA of the student is: 9.4761

15.04

Code:

```
import java.util.Scanner;

class Subject {
   int marks, credits, grade;
}

class Student {
   Subject subject[];
   String name, usn;
   double SGPA;
   Scanner s = new Scanner(System.in);

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

   void getStudentDetails() {
      System.out.println("Enter student name:");
      this.name = s.nextLine();
      System.out.println("Enter student USN:");
      this.usn = s.nextLine();
   }

   void getMarks() {
      for (int i = 0; i < 8; i++) {
         System.out.println("Enter marks of subject " + (i + 1) + ":");
         subject[i].marks = s.nextInt();
         System.out.println("Enter credits of subject " + (i + 1) + ":");
         subject[i].credits = s.nextInt();
         subject[i].grade = (subject[i].marks / 10) + 1;
         if (subject[i].grade > 10) {
            subject[i].grade = 10;
         } else if (subject[i].grade < 0) {
            subject[i].grade = 0;
         }
      }
   }

   void computeSGPA() {
      double points = 0;
      double totalCredits = 0;
      for (int i = 0; i < 8; i++) {
         int sub = subject[i].credits * subject[i].grade;
```

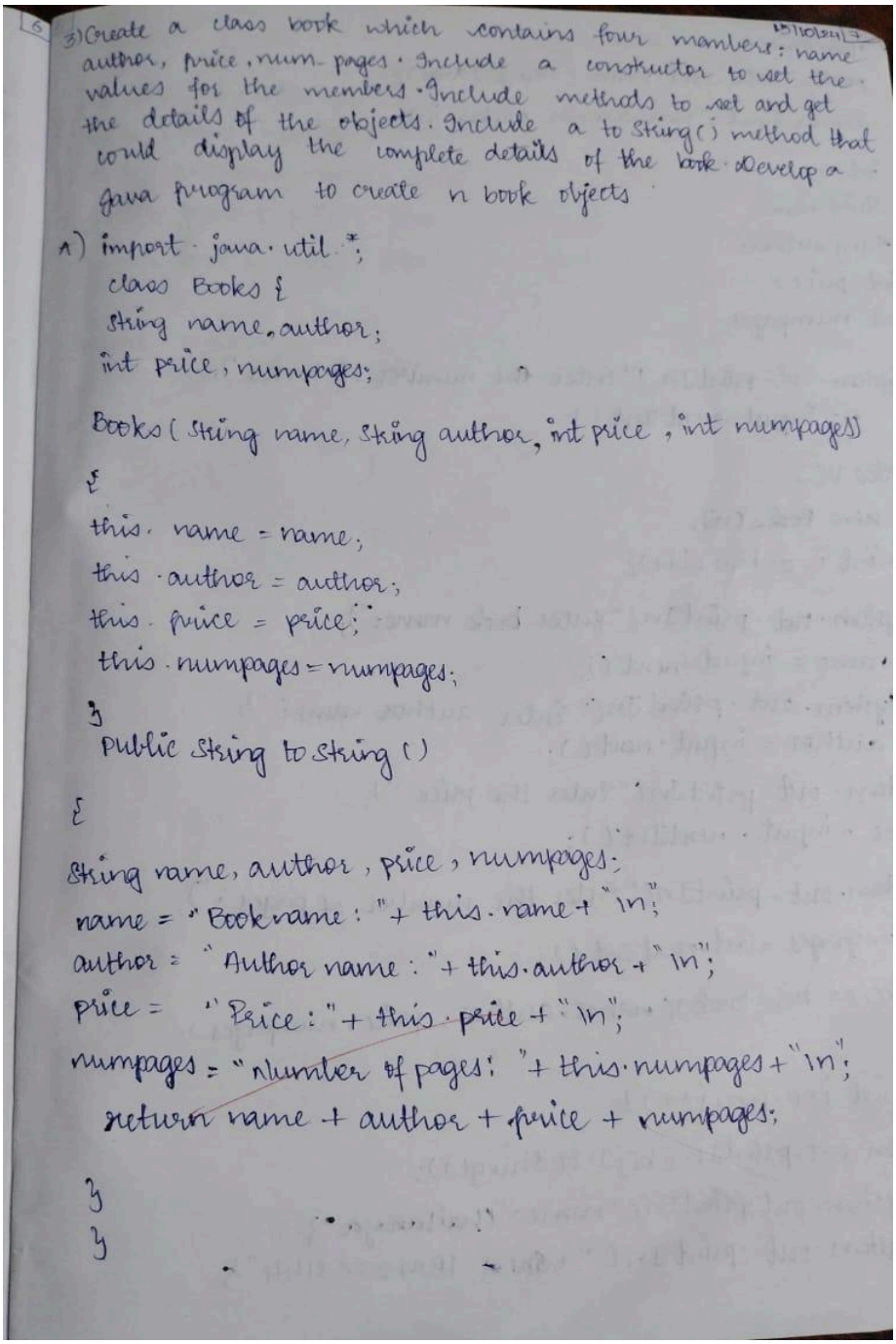
```
            points += sub;
            totalCredits += subject[i].credits;
        }
        SGPA = points / totalCredits;
        System.out.println("SGPA of the student is: " + SGPA);
    }
}

public class sgpa {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        System.out.println("Name: " + s1.name);
        System.out.println("USN: " + s1.usn);
        s1.computeSGPA();
    }
}
```

```
D:\1BM23CS144>javac sgpa.java

D:\1BM23CS144>java sgpa
Enter student name:
K.Chaitanya
Enter student USN:
1BM23CS144
Enter marks of subject 1:
90
Enter credits of subject 1:
4
Enter marks of subject 2:
89
Enter credits of subject 2:
4
Enter marks of subject 3:
90
Enter credits of subject 3:
3
Enter marks of subject 4:
98
Enter credits of subject 4:
3
Enter marks of subject 5:
78
Enter credits of subject 5:
3
Enter marks of subject 6:
99
Enter credits of subject 6:
1
Enter marks of subject 7:
100
Enter credits of subject 7:
1
Enter marks of subject 8:
78
Enter credits of subject 8:
1
Name: K.Chaitanya
USN: 1BM23CS144
SGPA of the student is: 9.4
```

Program 3

Method Overriding

Algorithm:



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

A)

```
import java.util.*;
class Books {
String name,author;
int price, 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 input = new Scanner (System.in);

int n;
String name;
String author;
int price;
int numpages;

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

Books b[];
b = new Books[n];
for(int i=0; i<n; i++){

System.out.println("Enter book name:");
  name = input.next();
  System.out.println("Enter author name:");
  author = input.next();
System.out.println("Enter the price:");
price = input.nextInt();
System.out.println("Enter the number of pages:");
  numpages = int.nextInt();
    b[i] = new Books(name, author, price, numpages);
  }
for(int j=0; j<n; j++){
System.out.println(b[j].toString());
  System.out.println("name: Chaitanya");
  System.out.println("USN: 1BM23CS144");
}
}
}
```

OUTPUT:

```
Number of books: 1
Enter book1 name: BOOK
Enter author1 name: enid
Enter book 1 price: 100
Enter book 1 pages: 200

name: Chaitanya
USN: 1BM23CS144
```

15.10

Code:

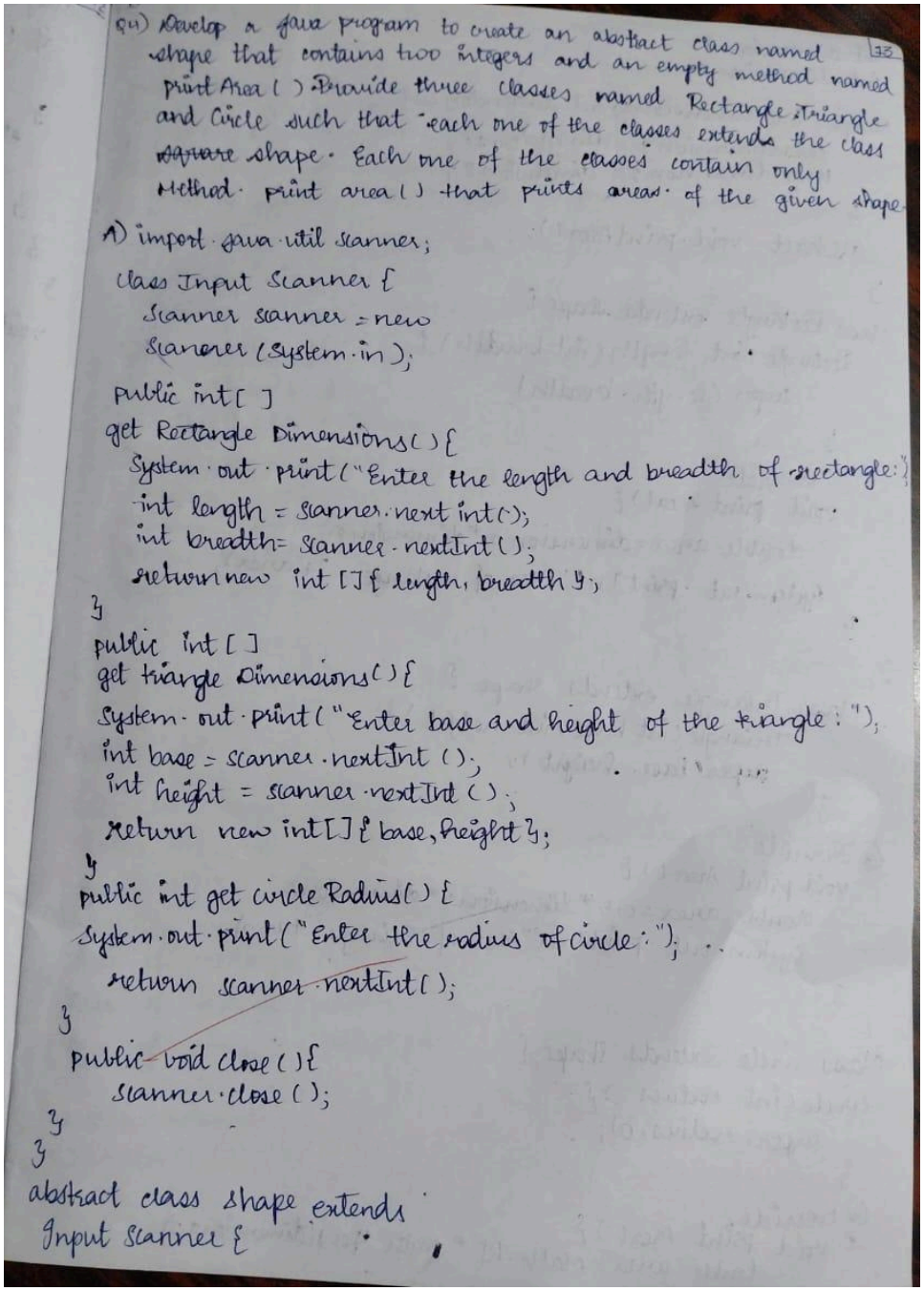
```
import java.util.Scanner;

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

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

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

      Books[] b = new Books[n];

      for (int i = 0; i < n; i++) {
         System.out.println("Enter the book name:");
         String name = s.nextLine();

         System.out.println("Enter the author:");
         String author = s.nextLine();

         System.out.println("Enter the price:");
         int price = s.nextInt();

         System.out.println("Enter the number of pages:");
         int numPages = s.nextInt();
         s.nextLine();
```

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

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

        System.out.println("Name:K.chaitanya");
        System.out.println("USN:1BM23CS!144");

        s.close();
    }
}
```

```
D:\1BM23CS144>java Main
Enter the number of books
1
Enter the book name:
air
Enter the author:
raj
Enter the price:
300
Enter the number of pages:
230
Book name: air
Author name: raj
Price: 300
Number of pages: 230

Name:K.chaitanya
USN:1BM23CS!144
```

Program 4
Abstract Class

Algorithm:



Q4) 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 method print area() that prints areas of the given shape.

```
1) import java.util.Scanner;
class Input Scanner {
  Scanner scanner = new
  Scanner(System.in);
  Public int[ ]
  get Rectangle Dimensions() {
    System.out.print("Enter the length and breadth of rectangle:");
    int length = scanner.nextInt();
    int breadth = scanner.nextInt();
    return new int[]{length, breadth};
  }
  public int[ ]
  get triangle Dimensions() {
  System.out.print("Enter base and height of the triangle:");
  int base = scanner.nextInt();
  int height = scanner.nextInt();
  return new int[]{base, height};
  }
  public int get circle Radius() {
  System.out.print("Enter the radius of circle:");
    return scanner.nextInt();
  }
  public void close() {
    scanner.close();
  }
}
abstract class shape extends
 Input Scanner {
```

```
int dimension1;
int dimension2;

Shape(int dimension1, int dimension2){
    this.dimension1 = dimension1;
    this.dimension2 = dimension2;
}
abstract void printArea();
}
class Rectangle extends Shape {
  Rectangle(int length, int breadth) {
      super(length, breadth);
  }
  @Override
  void printArea(){
    double area = dimension1 * dimension2;
    System.out.println("Area of rectangle = " + area);
  }
}
  class Triangle extends Shape {
      Triangle(int base, int height) {
        super(base, height);
  }
@Override
  void printArea() {
    double area = 0.5 * dimension1 * dimension2;
    System.out.println("Area of Triangle = "+area);
  }
}
class circle extends Shape {
circle(int radius ){
   super(radius, 0);
}
@Override
void printArea() {
    double area = Math.PI * Math.pow(dimension1,2);
    System.out.println("Area of circle = "+area);
}
public class area {
    public static void main(String[] args) {
    Input Scanner = new Input Scanner();
int[] rectangle Dimensions = Input Scanner.getRectangleDimensions();
    Shape rectangle = new Rectangle(rectangle Dimensions[0], rectangle Dimensions[1]);
int[] triangle Dimensions = input Scanner.getTriangle Dimensions();
    Shape Triangle = new
    Triangle(Triangle Dimensions[0], triangle Dimension[1]);
int circle Radius = Input Scanner.getCircle Radius();
    Shape circle = new
    circle(circle Radius);
System.out.println()
rectangle.printArea();
triangle.printArea();
Circle.printArea();
Input Scanner.close();
}
}
}
```

OUTPUT

```
Enter the length and breadth of rectangle : 10  9
Enter the base and height of triangle : 8  10
Enter the radius of circle : 7

Area of rectangle = 90.0
Area of triangle = 40.0
Area of circle = 153.93804
```

NAME: K. CHAITANYA
USN = 1BM23CS144

Code:

```
import java.util.Scanner;

class InputScanner {
    Scanner scanner = new Scanner(System.in);

    public int[] getRectangleDimensions() {
        System.out.print("Enter the length and breadth of the rectangle: ");
        int length = scanner.nextInt();
        int breadth = scanner.nextInt();
        return new int[]{length, breadth};
    }

    public int[] getTriangleDimensions() {
        System.out.print("Enter the base and height of the triangle: ");
        int base = scanner.nextInt();
        int height = scanner.nextInt();
        return new int[]{base, height};
    }

    public int getCircleRadius() {
        System.out.print("Enter the radius of the circle: ");
        return scanner.nextInt();
    }

    public void close() {
        scanner.close();
    }
}

abstract class Shape extends InputScanner {
    int dimension1;
    int dimension2;

    Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        super(length, breadth);
    }
```

```
    @Override
    void printArea() {
        double area = dimension1 * dimension2;
        System.out.println("Area of rectangle = " + area);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }

    @Override
    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of triangle = " + area);
    }
}

class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
    }

    @Override
    void printArea() {
        double area = Math.PI * Math.pow(dimension1, 2);
        System.out.println("Area of circle = " + area);
    }
}

public class Area {
    public static void main(String[] args) {
        InputScanner inputScanner = new InputScanner();

        int[] rectangleDimensions = inputScanner.getRectangleDimensions();
        Shape rectangle = new Rectangle(rectangleDimensions[0], rectangleDimensions[1]);

        int[] triangleDimensions = inputScanner.getTriangleDimensions();
        Shape triangle = new Triangle(triangleDimensions[0], triangleDimensions[1]);

        int circleRadius = inputScanner.getCircleRadius();
        Shape circle = new Circle(circleRadius);

        System.out.println();
        rectangle.printArea();
        triangle.printArea();
```

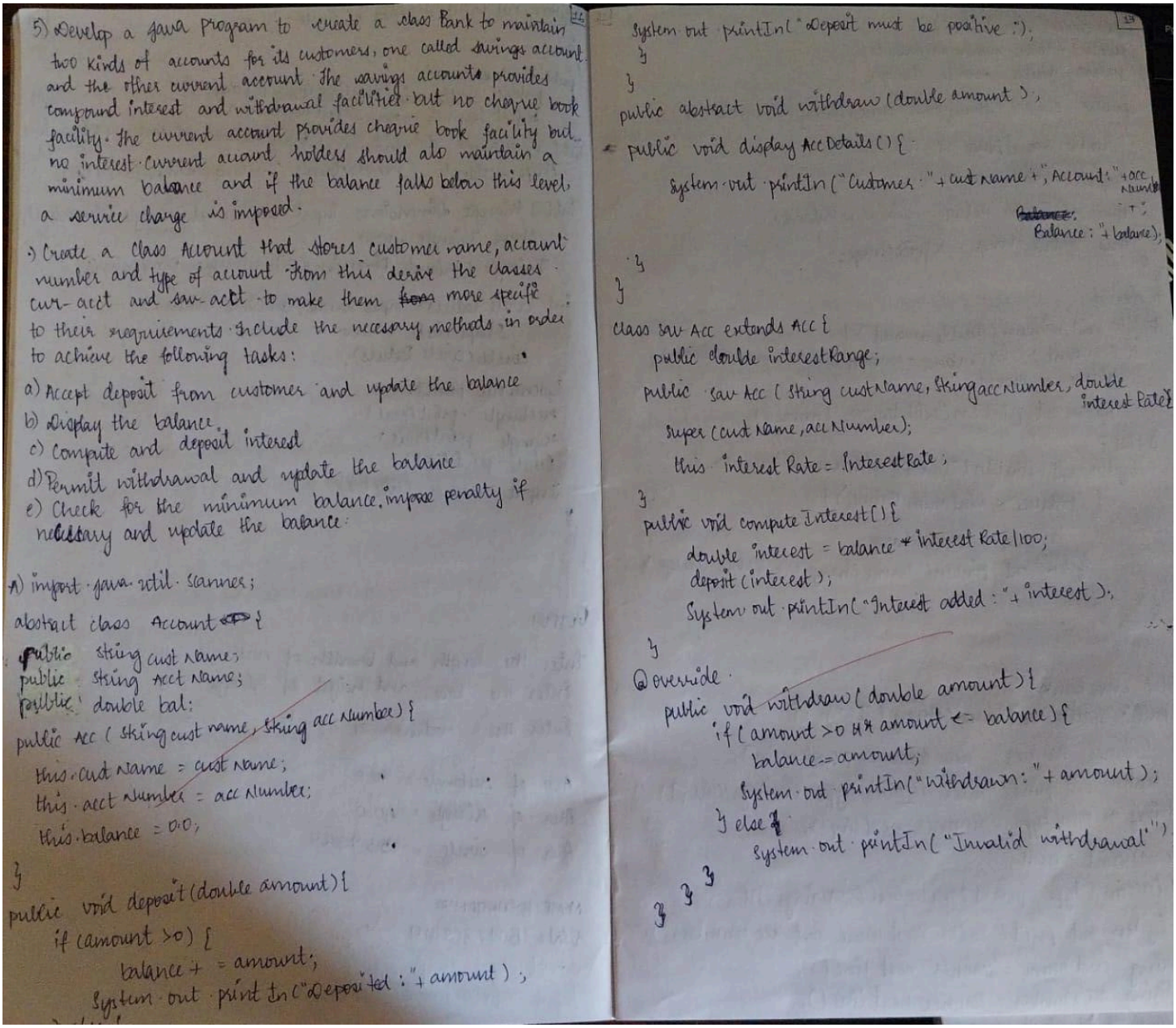
```
        circle.printArea();

        System.out.println("\nName: K.chaitanya");
        System.out.println("UN:1BM23CS144");

        inputScanner.close();
    }
}
```

```
D:\144>java Area
Enter the length and breadth of the rectangle: 10 9
Enter the base and height of the triangle: 8 10
Enter the radius of the circle: 7

Area of rectangle = 90.0
Area of triangle = 40.0
Area of circle = 153.93804002589985

Name: K.chaitanya
UN:1BM23CS144
```

Program 5

Bank Account

Algorithm:



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

- Create a class Account that stores customer name, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance
b) Display the balance
c) Compute and deposit interest
d) Permit withdrawal and update the balance
e) Check for the minimum balance, impose penalty if necessary and update the balance

```
A) import java.util.Scanner;
abstract class Account {
public String custName;
public String acctName;
public double bal;
public Acc (String custname, String accNumber) {
  this.custName = custName;
  this.acctNumber = accNumber;
  this.balance = 0.0;
}
public void deposit(double amount){
   if (amount >0) {
        balance += amount;
      System.out.println("Deposited :"+amount);
      System.out.println("Deposit must be positive");
   }
}
public abstract void withdraw(double amount);
public void displayAccDetails(){
    System.out.println("Customer :"+custName+", Account:"+accNumber+
                       "Balance: "+balance);
}
}
class savAcc extends Acc{
   public double interestRange;
   public savAcc(String custName, String accNumber, double interestRate){
     super(custName, accNumber);
     this.interestRate = interestRate;
   }
   public void computeInterest(){
      double interest = balance * interestRate/100;
      deposit(interest);
      System.out.println("Interest added :"+interest);
   }
   @override
   public void withdraw(double amount){
      if(amount >0 && amount <= balance){
         balance -= amount;
         System.out.println("withdrawn:"+amount);
      }else{
         System.out.println("Invalid withdrawal");
      }
   }
}
```

```
class curAcc extends Acc {
public double minimumBalance;
public double serviceCharge;
public curAcc (String custName, String accNumber, double minimumBalance,
    double serviceCharge){
Super (custName, accNumber);
this.minimumBalance = minimumBalance;
this.serviceCharge = serviceCharge;
}
@override
public void withdraw (double amount) {
  if (amount > 0 && (balance - amount) >= minimumBalance) {
    balance -= amount;
    System.out.println ("Withdrawn:" + amount);
} else {
  System.out.println ("Invalid withdrawal");
    if (balance < minimumBalance) {
        balance -= serviceCharge;
      System.out.println ("Service Charge:" + serviceCharge);
}
}
}
}

public class Bank {
  public static void main (String [] args) {
    Scanner scanner = new Scanner (System.in);
    System.out.print ("Enter account type (savings/current):");
String accountType = scanner.nextLine();
Acc account = null;
if (accountType.equalsIgnoreCase ("savings")){
  System.out.print ("Enter cust name and acc number:");
String custName = scanner.nextLine();
String accNumber = scanner.nextLine();
account = new SavAcc (custName, accNumber, 5.0);
} else if (accountType.equalsIgnoreCase ("current")) {
  System.out.println ("Enter cust name and acc number:");
  String custName = scanner.nextLine();
  String accNumber = scanner.nextLine();
  account = new CurAcc (custName, accNumber, 500.0, 50.0);
} else {
  System.out.println ("Invalid account");
  scanner.close();
  return;
}
while (true) {
  System.out.println ("\n1. Deposit 2. Withdraw 3. Compute Interest
                       4. Display 5. Exit");
  int choice = scanner.nextInt();
switch (choice) {
case 1:
System.out.print ("Enter deposit amount:");
  account.deposit (scanner.nextDouble());
  break;
case 2:
  System.out.print ("Enter withdrawal amount:");
  account.withdraw (scanner.nextDouble());
  break;
case 3:
  if (account instanceof SavAcc) {
      ((SavAcc) account).computeInterest();
} else {
  System.out.println ("Interest can only be computed for
                       Savings Acc");
}
  break;
  case 4:
    account.displayAccDetails();
    break;
  case 5:
    scanner.close();
    System.out.println ("NAME: K.CHAITANYA");
    System.out.println ("USN: 1BM23CS144");
    return;
```

OUTPUT:

Enter Account type (savings/Current): savings

Enter cust Name And acc Number: A

123

1. Deposit 2. withdraw 3. Compute Interest 4. Display 5. Exit

1

Enter deposit amount: 1000

Deposited: 1000.0

1. Deposit 2. withdraw 3. Compute Interest 4. Display 5. Exit

2

Enter withdrawal amount: 200

withdrawn: 200.0

3.

Deposited: 40.0

Interest added: 40.0

4.

Customer: a, Account: 123, Balance: 840.0

5

Name: K. Chaitanya

USN: 1BM23CS144

Enter Account type (savings/Current): Current

Enter Cust Name And Acc number: B

213

1. Deposit 2. withdraw 3. Compute Interest 4. Display 5. Exit

1

Enter Deposit amount: 1000

Deposited: 1000.0

2

Enter withdrawal amount: 250

withdrawn: 250.

3

Interest can only be computed for saving Account

4

Customer: B, Account: 213, Balance: 750.0

28/10

Code:

```
import java.util.Scanner;

abstract class Account {
    protected String custName;
    protected String acctNumber;
    protected double balance;

    public Account(String custName, String acctNumber) {
        this.custName = custName;
        this.acctNumber = acctNumber;
        this.balance = 0.0;
    }

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

    public abstract void withdraw(double amount);

    public void displayAcctDetails() {
        System.out.println("Customer: " + custName);
        System.out.println("Account Number: " + acctNumber);
        System.out.println("Balance: " + balance);
    }
}

class SavAcc extends Account {
    private double interestRate;

    public SavAcc(String custName, String acctNumber, double interestRate) {
        super(custName, acctNumber);
        this.interestRate = interestRate;
    }

    public void computeInterest() {
        double interest = balance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest added: " + interest);
    }

    @Override
```

```
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Invalid withdrawal amount.");
        }
    }
}

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

    public CurAcc(String custName, String acctNumber, double minimumBalance, double
serviceCharge) {
        super(custName, acctNumber);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    @Override
    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= minimumBalance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient funds.");
        }

        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " applied due to low balance.");
        }
    }
}

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

        System.out.print("Enter customer name: ");
        String custName = scanner.nextLine();

        System.out.print("Enter account number: ");
        String acctNumber = scanner.nextLine();
```

```
System.out.print("Enter account type (savings/current): ");
String accountType = scanner.nextLine();

Account account = null;

if (accountType.equalsIgnoreCase("savings")) {
    System.out.print("Enter interest rate: ");
    double interestRate = scanner.nextDouble();
    account = new SavAcc(custName, acctNumber, interestRate);

} else if (accountType.equalsIgnoreCase("current")) {
    System.out.print("Enter minimum balance: ");
    double minimumBalance = scanner.nextDouble();
    System.out.print("Enter service charge: ");
    double serviceCharge = scanner.nextDouble();
    account = new CurAcc(custName, acctNumber, minimumBalance, serviceCharge);

} else {
    System.out.println("Invalid account type.");
    scanner.close();
    return;
}

while (true) {
    System.out.println("\nMENU");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Compute interest for Savings Account");
    System.out.println("4. Display account details");
    System.out.println("5. Exit");

    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter the deposit amount: ");
            double depositAmount = scanner.nextDouble();
            account.deposit(depositAmount);
            break;

        case 2:
            System.out.print("Enter the withdrawal amount: ");
            double withdrawalAmount = scanner.nextDouble();
            account.withdraw(withdrawalAmount);
            break;
```

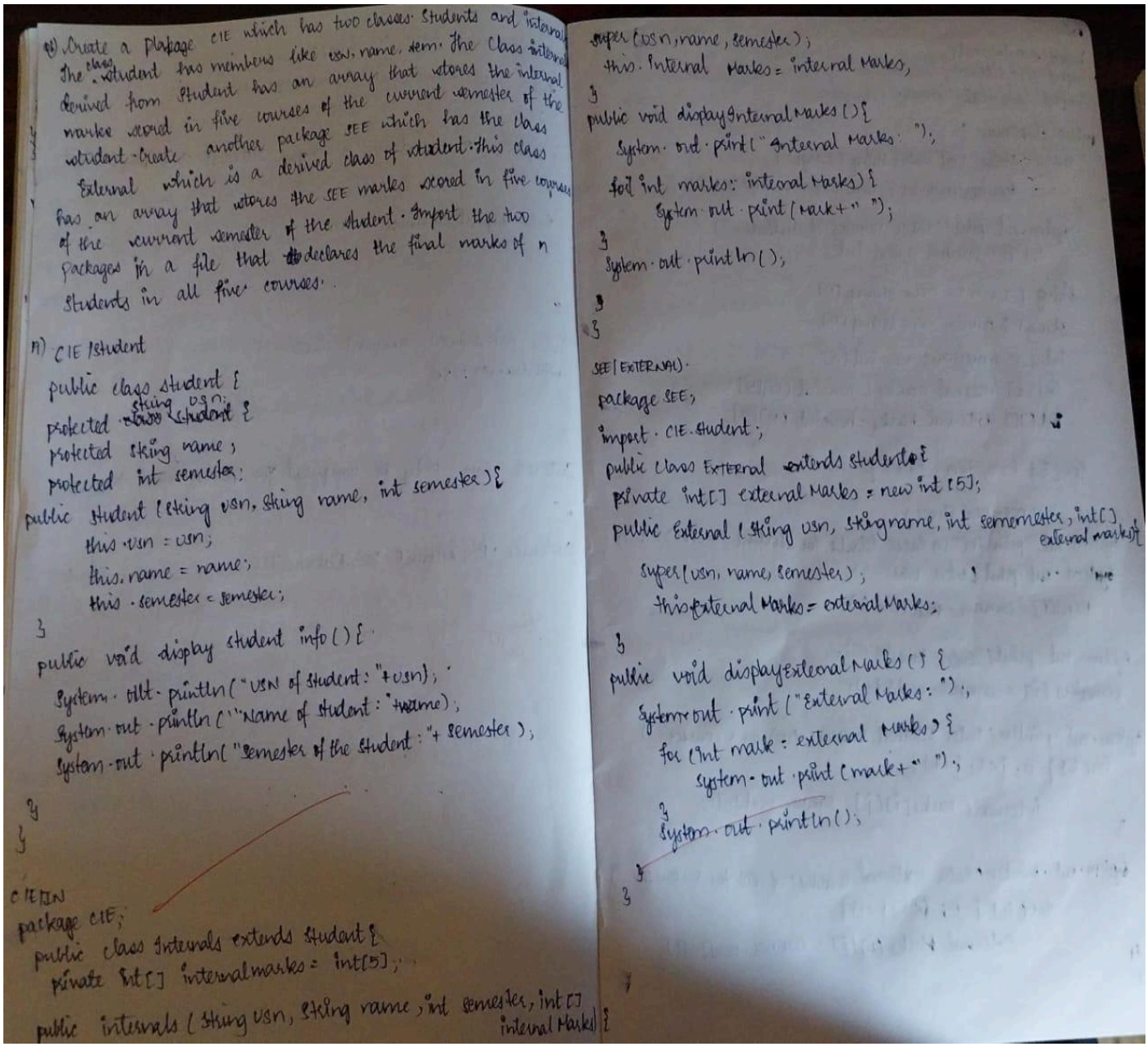
```
                case 3:
                    if (account instanceof SavAcc) {
                        ((SavAcc) account).computeInterest();
                    } else {
                        System.out.println("Interest can only be computed for Savings Account.");
                    }
                    break;

                case 4:
                    account.displayAcctDetails();
                    break;

                case 5:
                    System.out.println("K.Chaitanya");
                    System.out.println("USN:1BM23CS144");
                    scanner.close();
                    return;

                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```

```
D:\1BM23CS144>javac Bank.java

D:\1BM23CS144>java Bank
Enter customer name: A
Enter account number: 123
Enter account type (savings/current): savings
Enter interest rate: 5

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 1
Enter the deposit amount: 1000
Deposited: 1000.0

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 2
Enter the withdrawal amount: 200
Withdrawn: 200.0

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 3
Deposited: 40.0
Interest added: 40.0

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 40
Invalid choice. Please try again.

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 4
Customer: A
Account Number: 123
Balance: 840.0
```

```
D:\1BM23CS144>javac Bank.java

D:\1BM23CS144>java Bank
Enter customer name: B
Enter account number: 213
Enter account type (savings/current): current
Enter minimum balance: 1000
Enter service charge: 10

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 1
Enter the deposit amount: 1000
Deposited: 1000.0

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 2
Enter the withdrawal amount: 250
Invalid withdrawal amount or insufficient funds.

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 3
Interest can only be computed for Savings Account.

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 4
Customer: B
Account Number: 213
Balance: 1000.0

MENU
1. Deposit
2. Withdraw
3. Compute interest for Savings Account
4. Display account details
5. Exit
Enter your choice: 5
K.Chaitanya
USN:1BM23CS144
```

Program 6
Packages

Algorithm:



Main.

```
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();

        String[] usns = new String[n];
        String[] names = new String[n];
        int[] semesters = new int[n];
        int[][] internalMarks = new int[n][5];
        int[][] externalMarks = new int[n][5];

        for (int i = 0; i < n; i++) {
            scanner.nextLine();
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            System.out.print("Enter USN: ");
            usns[i] = scanner.nextLine();

            System.out.print("Enter name: ");
            names[i] = scanner.nextLine();

            System.out.print("Enter semester: ");
            semesters[i] = scanner.nextInt();

            System.out.println("Enter internal marks (out of 50) for 5 courses: ");
            for (int j = 0; j < 5; j++) {
                internalMarks[i][j] = scanner.nextInt();
            }

            System.out.println("Enter external marks (out of 100) for 5 courses: ");
            for (int j = 0; j < 5; j++) {
                externalMarks[i][j] = scanner.nextInt();
            }
        }

        for (int i = 0; i < n; i++) {
            Internals studentInternal = new Internal(usns[i], names[i],
                    semesters[i], internalMarks[i]);
            External studentExternal = new External(usns[i], names[i],
                    semesters[i], externalMarks[i]);

            studentInternal.displayStudentInfo();
            studentInternal.displayInternalMarks();
            studentExternal.displayExternalMarks();

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

OUTPUT:

```
Enter no. of students: 1

Enter details for student 1:
Enter USN: 1BM23CS144
Enter name: Chaitanya
Enter Semester: 3
Enter internal marks for 5 subjects:
Sub 1: 30
Sub 2: 30
Sub 3: 30
Sub 4: 30
Sub 5: 30
Enter external marks for 5 subjects:
Sub 1: 80
Sub 2: 80
```

Sub 4: 80
Sub 5: 80

Final marks for each Student

Details of Student 1:
USN: 1BM23CS144
Name: Chaitanya
Sub 1: 110
Sub 2: 110
Sub 3: 110
Sub 4: 110
Sub 5: 110;

NAME: K. CHAITANYA
USN: 1BM23CS144

Code:

```
package CIE;

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

   public Internals(String usn, String name, int semester, int[] internalMarks) {
      super(usn, name, semester);
      this.internalMarks = internalMarks;
   }

   public void displayInternalMarks() {
      System.out.println("Internal Marks (out of 50):");
      for (int i = 0; i < 5; i++) {
         System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
      }
   }
}
```

```
package CIE;

public class Student {
   protected String usn;
   protected String name;
   protected int semester;

   public Student(String usn, String name, int semester) {
      this.usn = usn;
      this.name = name;
      this.semester = semester;
   }

   public void displayStudentInfo() {
      System.out.println("Student Details:");
      System.out.println("USN: " + usn);
      System.out.println("Name: " + name);
      System.out.println("Semester: " + semester);
   }
}
```

```
package SEE;

import CIE.Internals;

public class External extends Internals {
   protected int[] externalMarks = new int[5];
```

```
    public External(String usn, String name, int semester, int[] internalMarks, int[] externalMarks) {
        super(usn, name, semester, internalMarks);
        this.externalMarks = externalMarks;
    }

    public void displayExternalMarks() {
        System.out.println("External Marks (out of 100):");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
    }
}


import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();

        String[] usns = new String[n];
        String[] names = new String[n];
        int[] semesters = new int[n];
        int[][] internalMarks = new int[n][5];
        int[][] externalMarks = new int[n][5];

        for (int i = 0; i < n; i++) {
            scanner.nextLine();

            System.out.println("\nEnter details for student " + (i + 1) + ":");

            System.out.print("Enter USN: ");
            usns[i] = scanner.nextLine();

            System.out.print("Enter Name: ");
            names[i] = scanner.nextLine();

            System.out.print("Enter Semester: ");
            semesters[i] = scanner.nextInt();

            System.out.println("Enter Internal marks (out of 50) for 5 courses:");
```

```
            for (int j = 0; j < 5; j++) {
                internalMarks[i][j] = scanner.nextInt();
            }

            System.out.println("Enter External marks (out of 100) for 5 courses:");
            for (int j = 0; j < 5; j++) {
                externalMarks[i][j] = scanner.nextInt();
            }
        }

        for (int i = 0; i < n; i++) {
            Internals studentInternal = new Internals(usns[i], names[i], semesters[i], internalMarks[i]);
            External studentExternal = new External(usns[i], names[i], semesters[i], internalMarks[i],
externalMarks[i]);

            studentInternal.displayStudentInfo();
            studentInternal.displayInternalMarks();
            studentExternal.displayExternalMarks();

            System.out.println("Final Marks:");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internalMarks[i][j] + externalMarks[i][j];
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }

            System.out.println("Name: K.CHAITANYA");
            System.out.println("USN: 1BM23CS144");
        }

        scanner.close();
    }
}
```

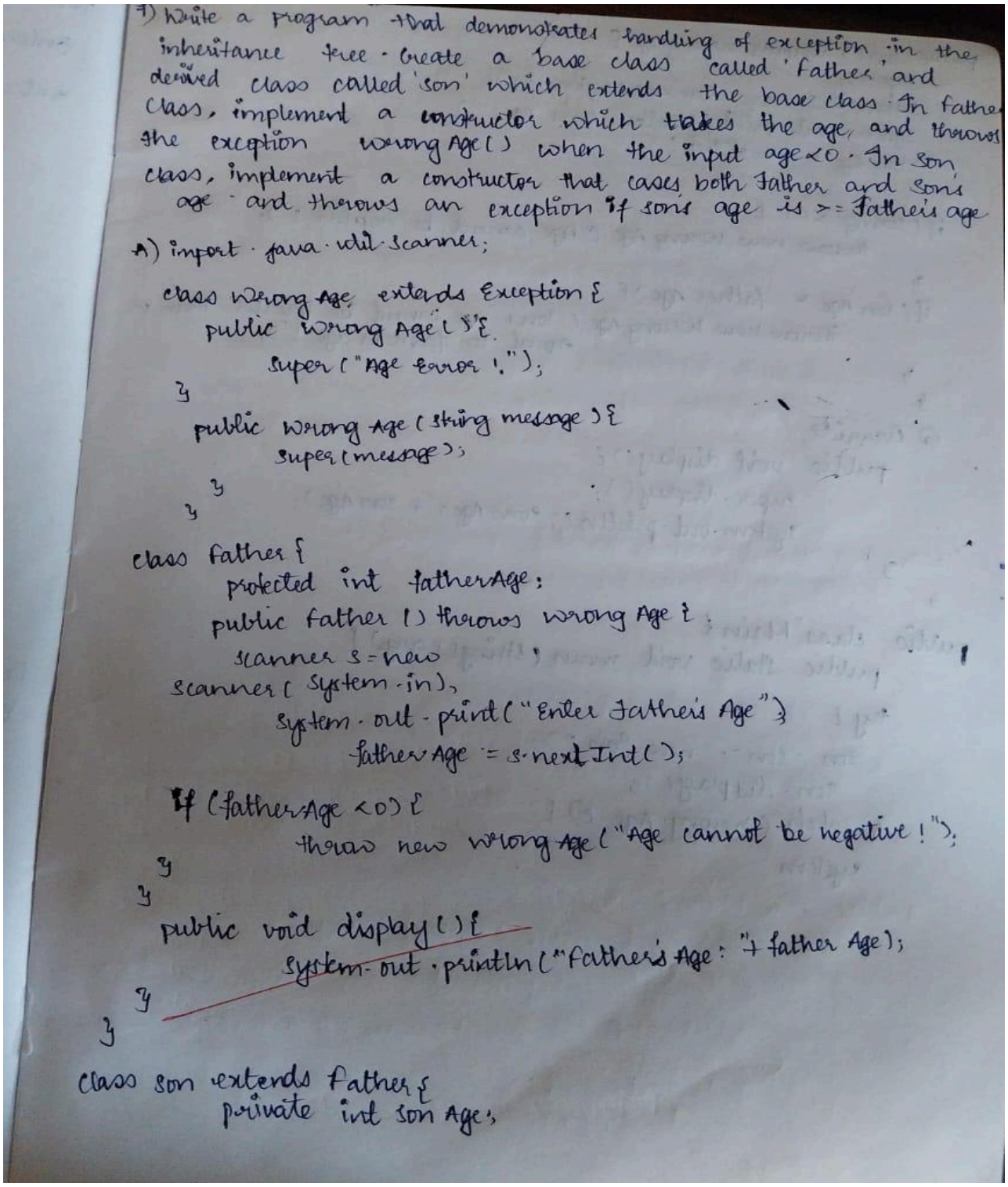
```
PS D:\1BM23CS144-6> & 'C:\Program Files\Java\jdk-23\bin
dhat.java\jdt_ws\1BM23CS144-6_3a8c2446\bin' 'Main'
Enter number of students: 2

Enter details for student 1:
Enter USN: 1BM23CS144
Enter Name: CHAITANYA
Enter Semester: 3
Enter Internal marks (out of 50) for 5 courses:
80
80
80
80
80
Enter External marks (out of 100) for 5 courses:
70
70
70
70
70

Enter details for student 2:
Enter USN: 1BM23CS061
Enter Name: VATSAL
Enter Semester: 3
Enter Internal marks (out of 50) for 5 courses:
80
80
80
80
80
Enter External marks (out of 100) for 5 courses:
70
70
70
70
70
Student Details:
USN: 1BM23CS144
Name: CHAITANYA
Semester: 3
Internal Marks (out of 50):
Course 1: 80
Course 2: 80
Course 3: 80
Course 4: 80
Course 5: 80
External Marks (out of 100):
Course 1: 70
Course 2: 70
Course 3: 70
Course 4: 70
Course 5: 70
Final Marks:
Course 1: 150
Course 2: 150
Course 3: 150
```

```
Enter External marks (out of 100) for 5 courses:
70
70
70
70
70
Student Details:
USN: 1BM23CS144
Name: CHAITANYA
Semester: 3
Internal Marks (out of 50):
Course 1: 80
Course 2: 80
Course 3: 80
Course 4: 80
Course 5: 80
External Marks (out of 100):
Course 1: 70
Course 2: 70
Course 3: 70
Course 4: 70
Course 5: 70
Final Marks:
Course 1: 150
Course 2: 150
Course 3: 150
Course 4: 150
Course 5: 150
Name: K.CHAITANYA
USN: 1BM23CS144
Student Details:
USN: 1BM23CS061
Name: VATSAL
Semester: 3
Internal Marks (out of 50):
Course 1: 80
Course 2: 80
Course 3: 80
Course 4: 80
Course 5: 80
External Marks (out of 100):
Course 1: 70
Course 2: 70
Course 3: 70
Course 4: 70
Course 5: 70
Final Marks:
Course 1: 150
Course 2: 150
Course 3: 150
Course 4: 150
Course 5: 150
Name: K.CHAITANYA
USN: 1BM23CS144
```

Program 7

Exception Handling

Algorithm:



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

A)

```
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 < 0) {
        throw new WrongAge("Age cannot be negative!");
    }
    if(sonAge >= fatherAge){
        throw new WrongAge("Son's age cannot be greater than or
                            equal to father's age!");
    }
}
@Override
    public void display(){
        super.display();
        System.out.println("Son's Age: " + sonAge);
    }
}
public class Main {
    public static void main(String[] args) {
    try {
        Son son = new Son();
        son.display();
    } catch (WrongAge e) {
        System
```

OUTPUT:

Enter father's Age: 35
Enter Son's Age: 35
Exception: Son's Age cannot be greater than or equal to father's Age.

Enter father's Age: 50
Enter Son's Age = 0

Enter father's Age: -23
Exception: Age cannot be negative!

NAME: K. CHAITANYA
USN: 1BM23CS144

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; // Age of the father


   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; // Age of the son


   public Son() throws WrongAge {
      super(); // Call Father constructor
      Scanner s = new Scanner(System.in);
      System.out.print("Enter Son's Age: ");
      sonAge = s.nextInt();
```

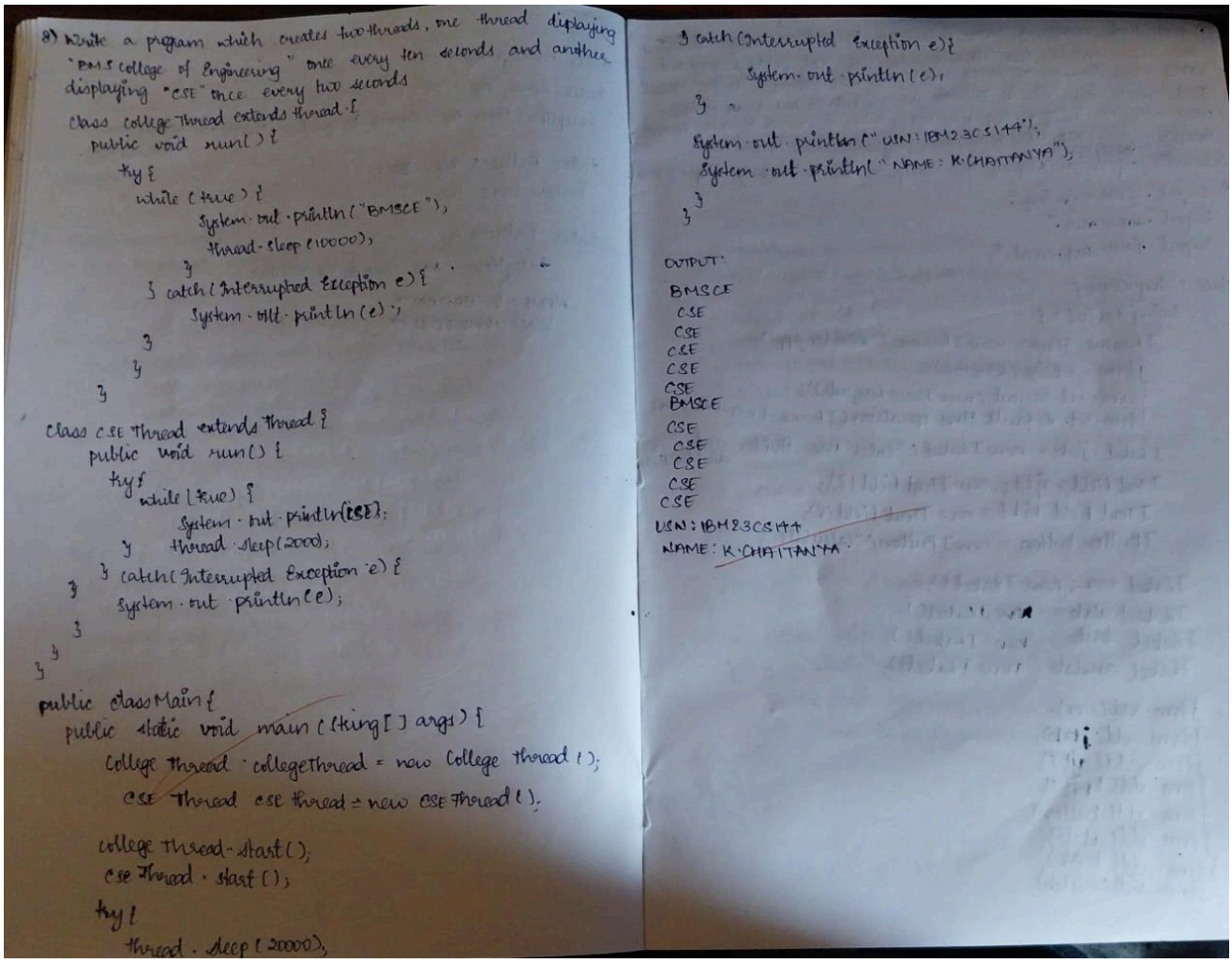
```
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative!");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age!");
        }
    }


    @Override
    public void display() {
        super.display(); // Call Father's display method
        System.out.println("Son's Age: " + sonAge);
    }
}


public class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son(); // Create Son object (also initializes Father)
            son.display(); // Display ages of father and son
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

```
D:\1BM23CS144>javac Main.java

D:\1BM23CS144>java Main
Enter Father's Age: 35
Enter Son's Age: 35
Exception: Son's age cannot be greater than or equal to father's age!

D:\1BM23CS144>javac Main.java

D:\1BM23CS144>java Main
Enter Father's Age: 1
Enter Son's Age: 0
Father's Age: 1
Son's Age: 0

D:\1BM23CS144>javac Main.java

D:\1BM23CS144>java Main
Enter Father's Age: -23
Exception: Age cannot be negative!
```

Program 8

Threads

Algorithm:



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

```
class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMSCE");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

        collegeThread.start();
        cseThread.start();

        try {
            Thread.sleep(20000);
        } catch (InterruptedException e) {
            System.out.println(e);
        }

        System.out.println("USN: 1BM23CS144");
        System.out.println("NAME: K.CHAITANYA");
    }
}
```

OUTPUT:

```
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
USN: 1BM23CS144
NAME: K.CHAITANYA
```

Code:

```
class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
```

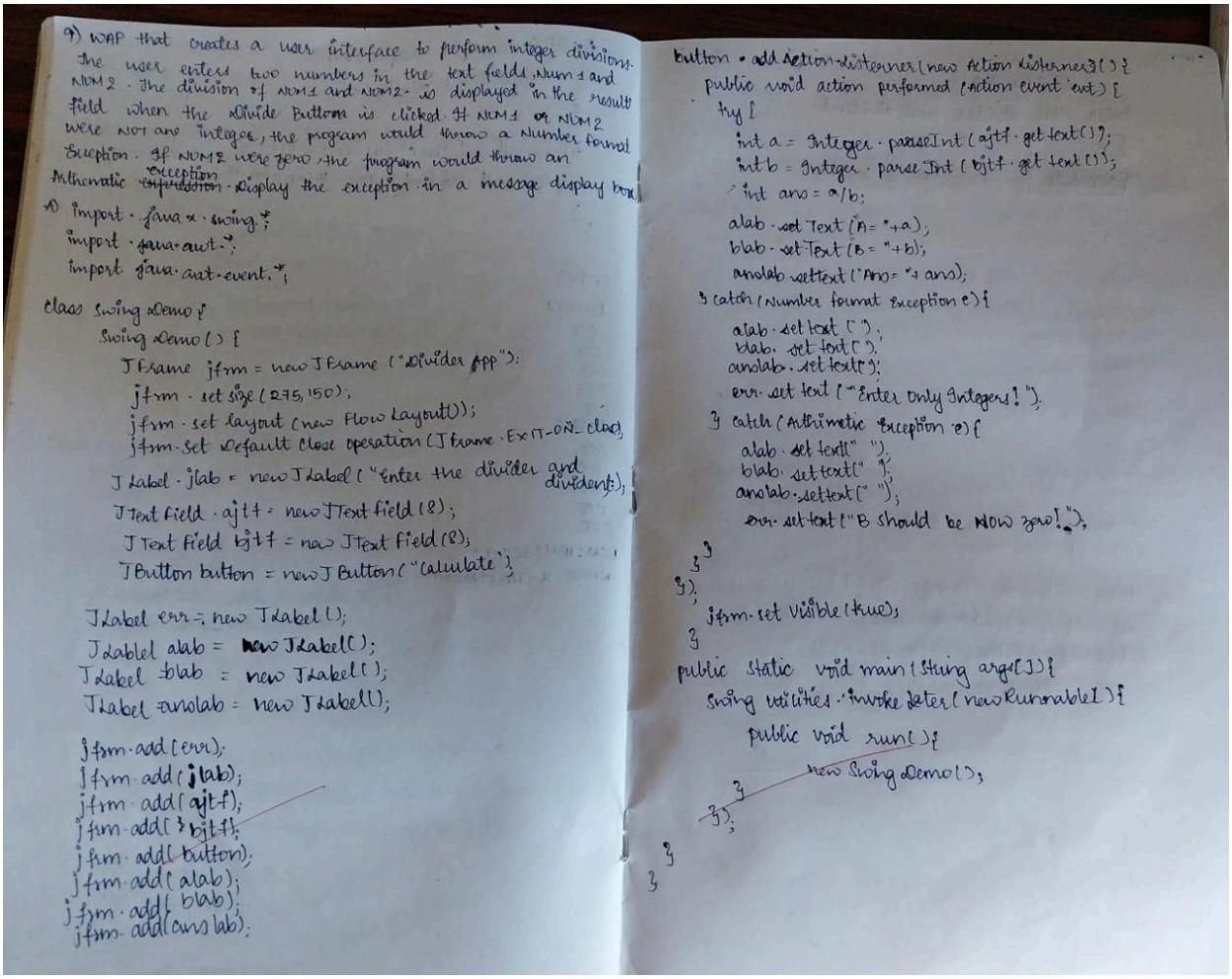
```
class CSEThread extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("CSE");
            Thread.sleep(2000);
         }
      } catch (InterruptedException e) {
         System.out.println(e);
      }
   }
}

public class Main {
   public static void main(String[] args) {
      CollegeThread collegeThread = new CollegeThread();
      CSEThread cseThread = new CSEThread();

      collegeThread.start();
      cseThread.start();

      try {
         Thread.sleep(20000);
      } catch (InterruptedException e) {
         System.out.println(e);
      }

      System.out.println("USN:1BM23CS144");
      System.out.println("Name:K.CHAITANYA");
   }
}
```

```
D:\1BM23CS144-8>javac Main.java

D:\1BM23CS144-8>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
USN:1BM23CS144
Name:K.CHAITANYA
```

Program 9

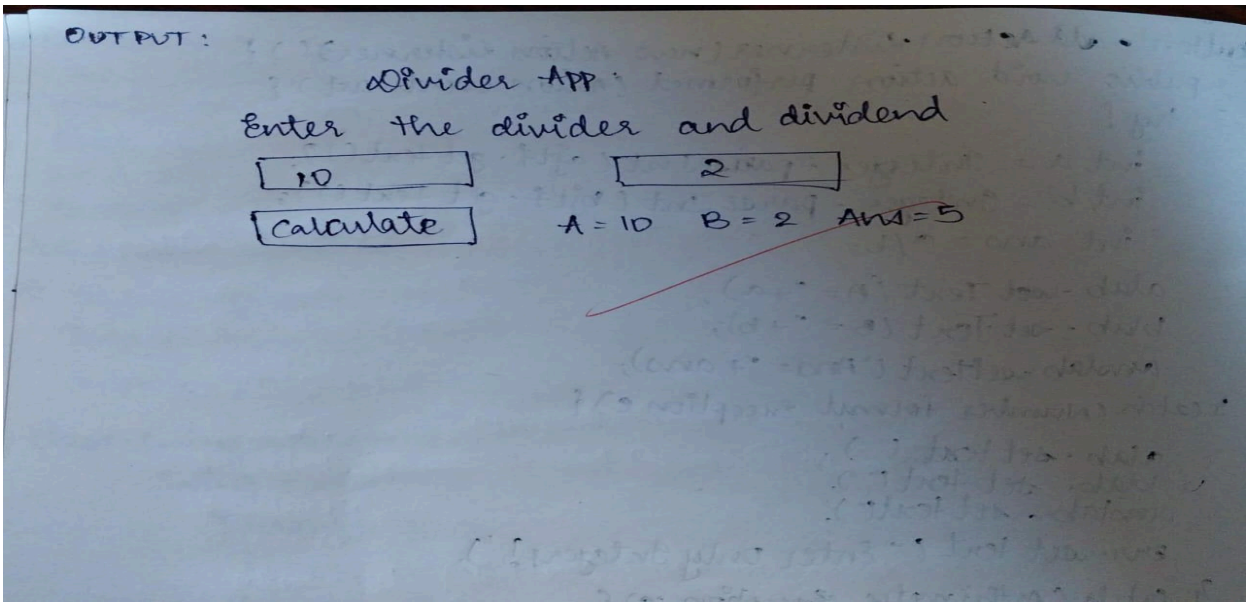
Open End Question 1

Algorithm:



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

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the divider and dividend:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");

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

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

        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("A= "+a);
                    blab.setText("B= "+b);
                    anslab.setText("Ans= "+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 NOW zero!");
                }
            }
        });
        jfrm.setVisible(true);
    }
    public static void main(String args[]) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
}
```



OUTPUT:

Divider App

Enter the divider and dividend

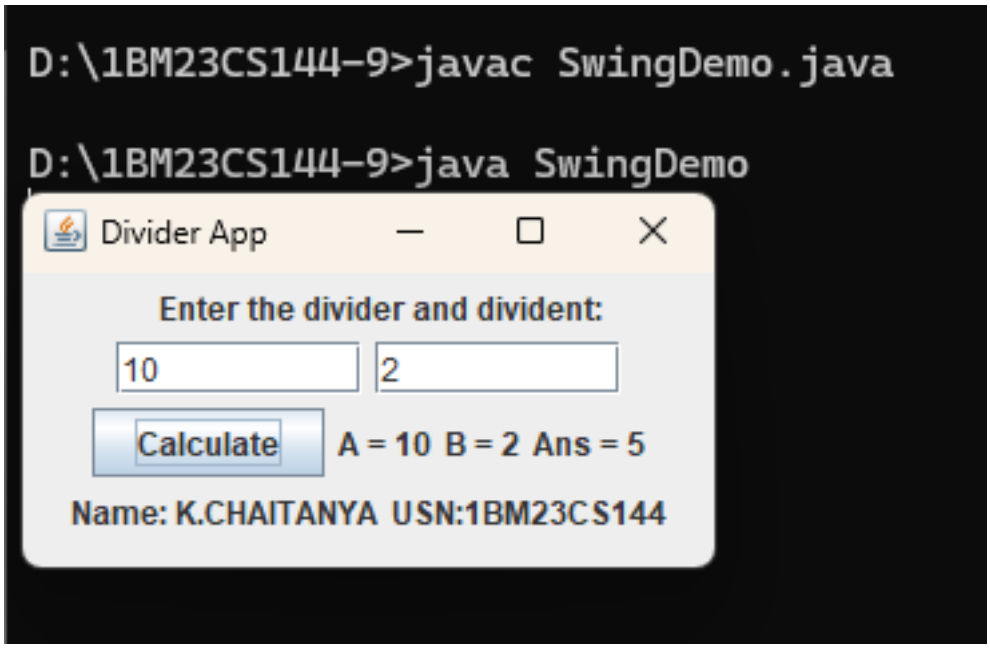
10 2

Calculate A = 10 B = 2 Ans = 5

Code:

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

class SwingDemo {
   SwingDemo() {
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      JLabel jlab = new JLabel("Enter the divider and divident:");
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);
      JButton button = new JButton("Calculate");

      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();
      JLabel nameLabel = new JLabel();
      JLabel rollNoLabel = 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);
      jfrm.add(nameLabel);
      jfrm.add(rollNoLabel);

      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("A = " + a);
               blab.setText("B = " + b);
               anslab.setText("Ans = " + ans);
               err.setText("");  // Clear error label if valid input
            } catch (NumberFormatException e) {
               alab.setText("");
```

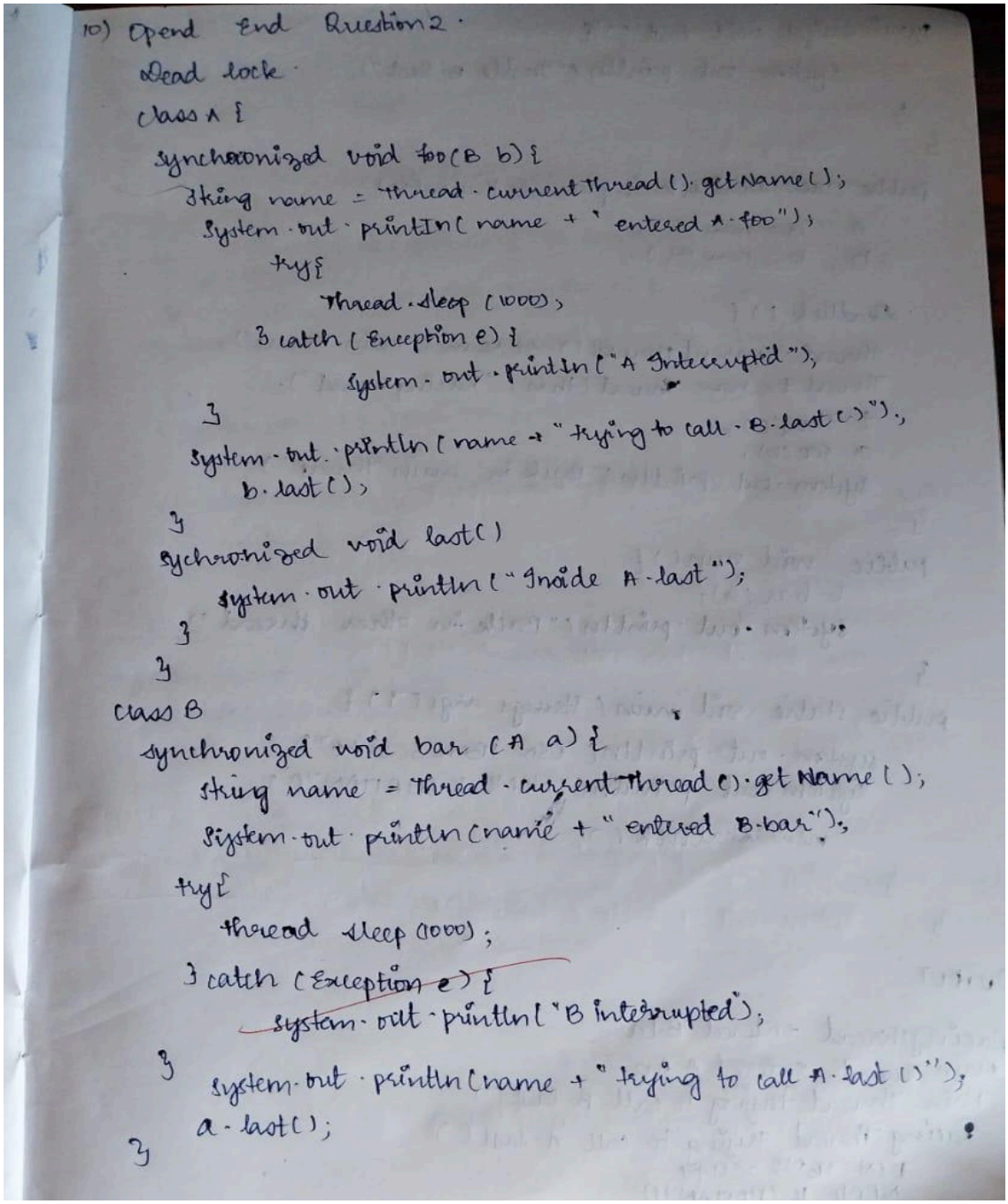
```
                blab.setText("");
                anslab.setText("");
                err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });


    nameLabel.setText("Name: K.CHAITANYA");
    rollNoLabel.setText("USN:1BM23CS144");

    jfrm.setVisible(true);
  }

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



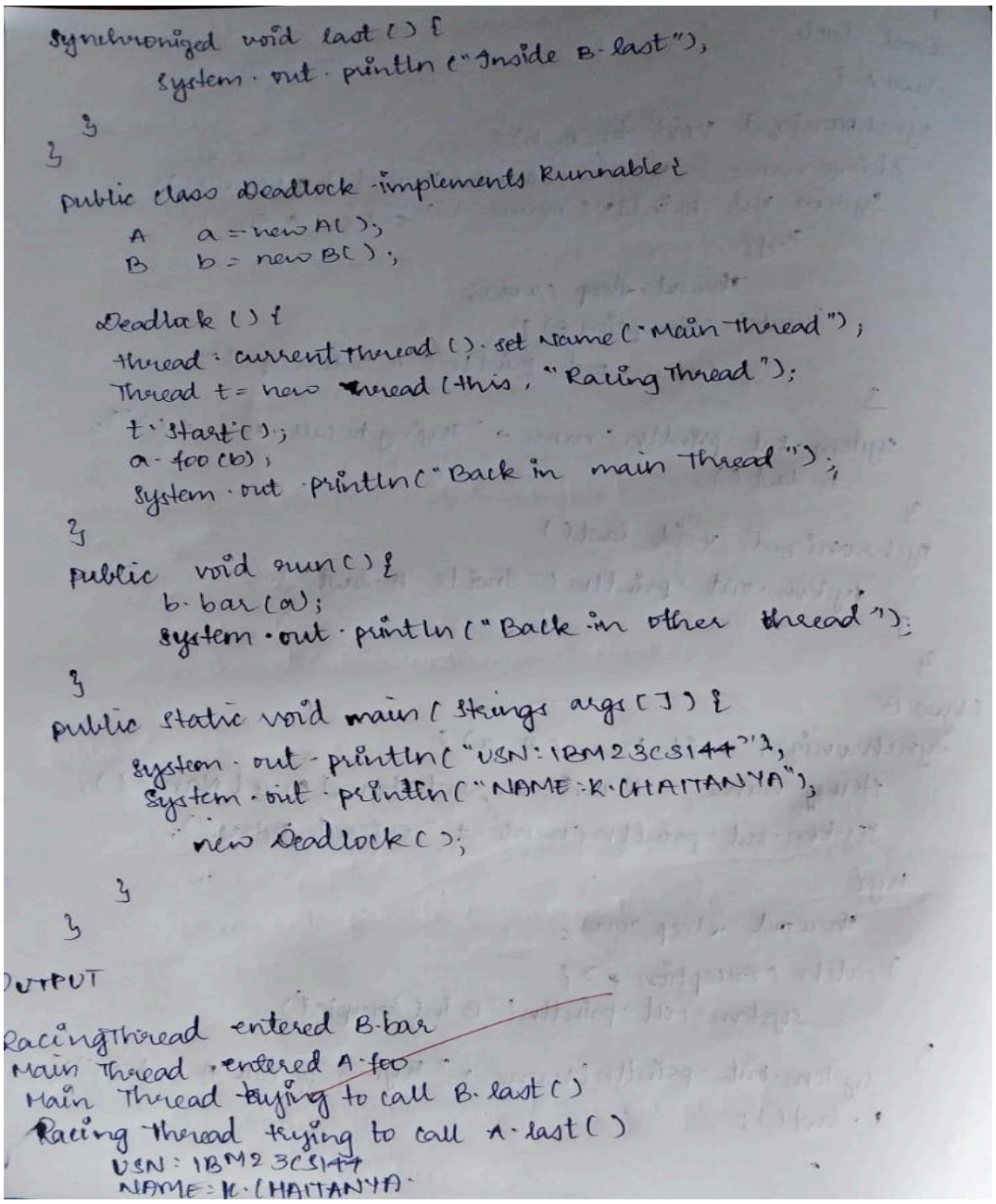
Program 10

Open End Question 2

Algorithm:



```
10) Opend End Question 2.
Dead lock
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();
 }
 synchronized 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();
}
```



Code:

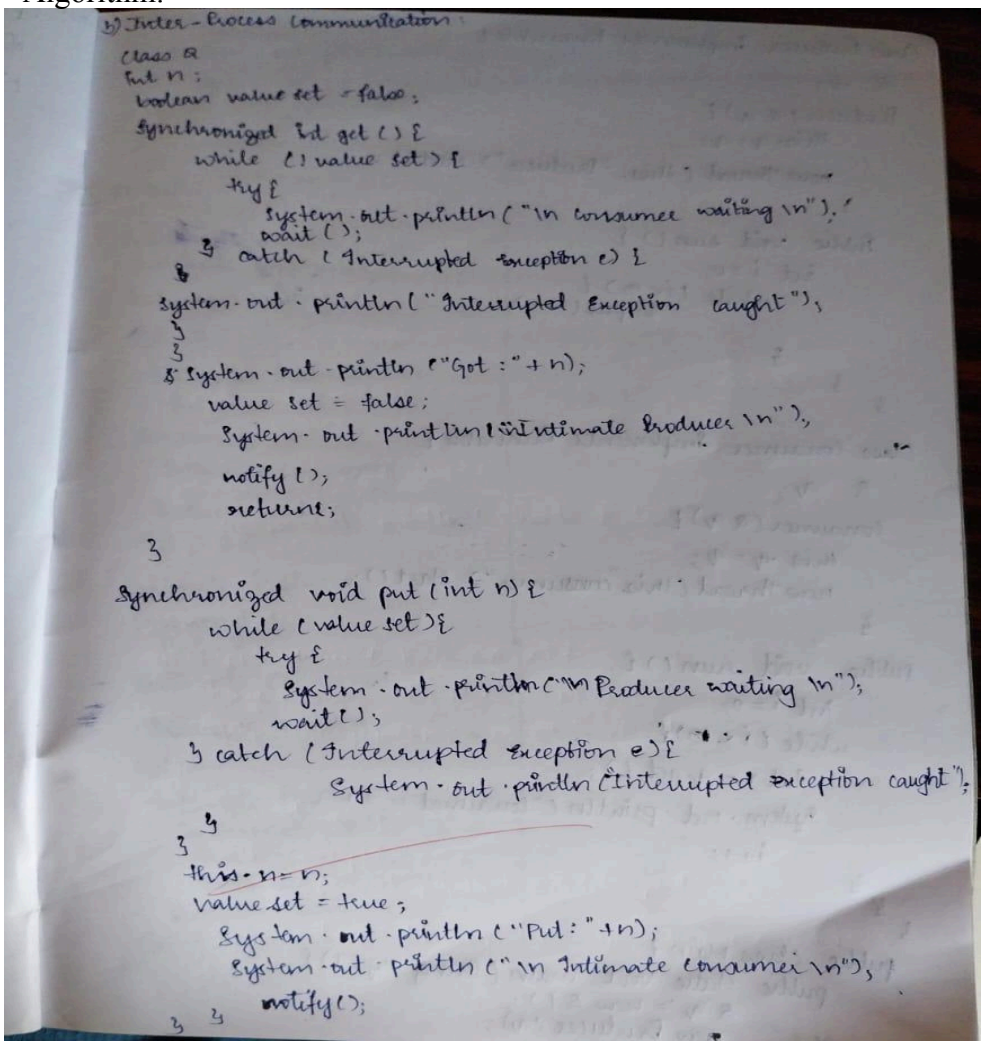
```
class A {
   synchronized void foo(B b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");
      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }
```

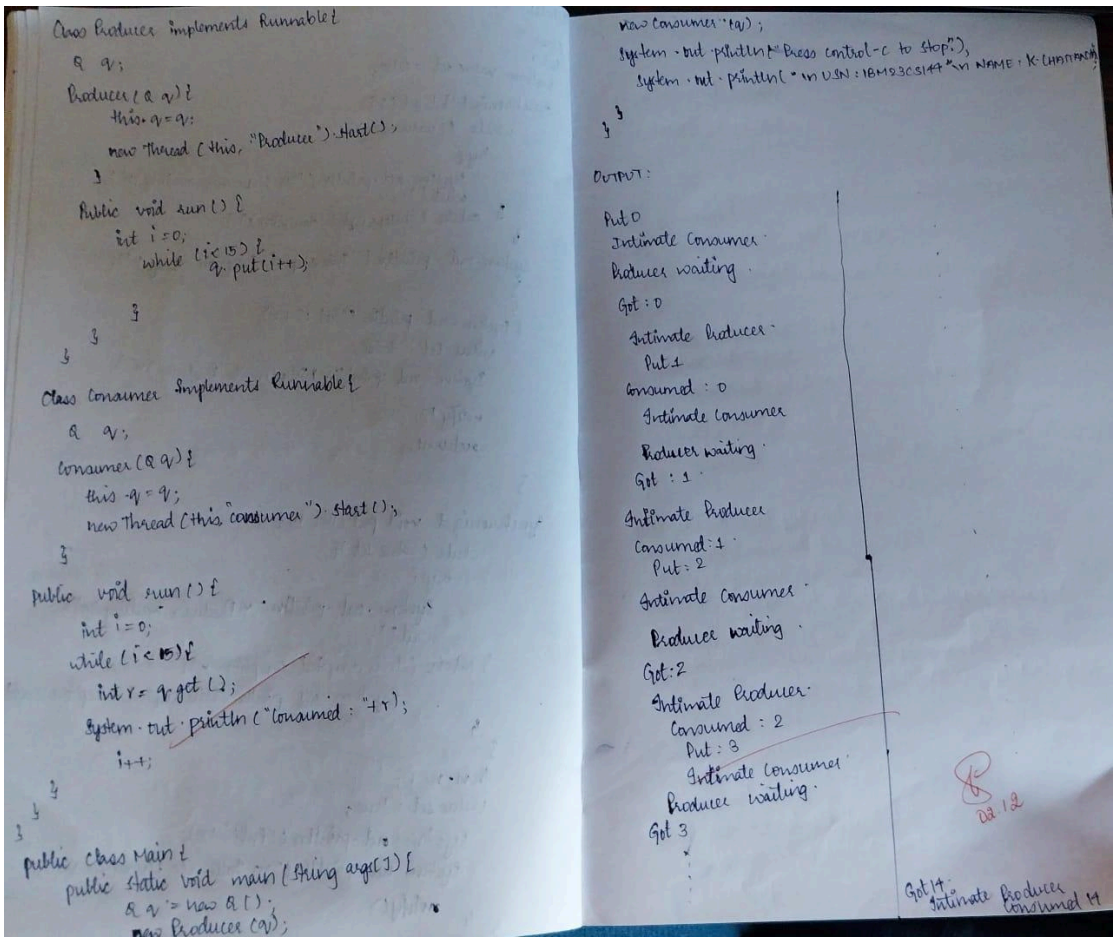
```
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    synchronized 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();
    }

    synchronized void last() {
        System.out.println("Inside B.last");
    }
}

public 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("USN: 1BM23CS144");
        System.out.println("Name:K.CHAITANYA");
```

```
        new Deadlock();
    }
}
```

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS144-DEADLOCK>javac Deadlock.java

D:\1BM23CS144-DEADLOCK>java Deadlock
USN: 1BM23CS144
Name:K.CHAITANYA
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
```

Algorithm:



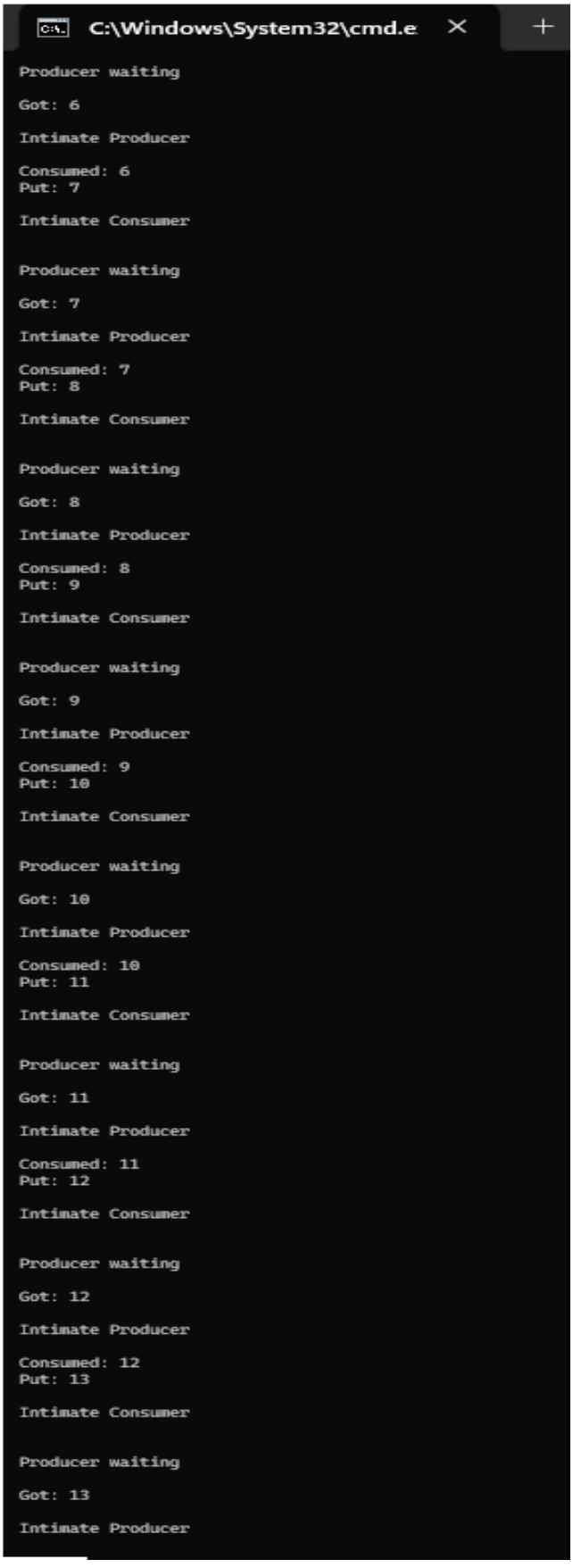


Code:

```
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++
```

```
        }
    }
}

public class Main {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("\nUSN: 1BM23CS144\nName:K.CHAITANYA");
    }}
```


C:\Windows\System32\cmd.e
Producer waiting
Got: 6
Intimate Producer
Consumed: 6
Put: 7
Intimate Consumer
Producer waiting
Got: 7
Intimate Producer
Consumed: 7
Put: 8
Intimate Consumer
Producer waiting
Got: 8
Intimate Producer
Consumed: 8
Put: 9
Intimate Consumer
Producer waiting
Got: 9
Intimate Producer
Consumed: 9
Put: 10
Intimate Consumer
Producer waiting
Got: 10
Intimate Producer
Consumed: 10
Put: 11
Intimate Consumer
Producer waiting
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


C:\Windows\System32\cmd.e
Intimate Consumer
Producer waiting
Got: 9
Intimate Producer
Consumed: 9
Put: 10
Intimate Consumer
Producer waiting
Got: 10
Intimate Producer
Consumed: 10
Put: 11
Intimate Consumer
Producer waiting
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:\1BM23CS144-INTER_PROCESS COMMUNICATION>