VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Chadive Muralidhar Reddy (1BM23CS072)

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 Chadive Muralidhar Reddy **(1BM23CS072),** 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.

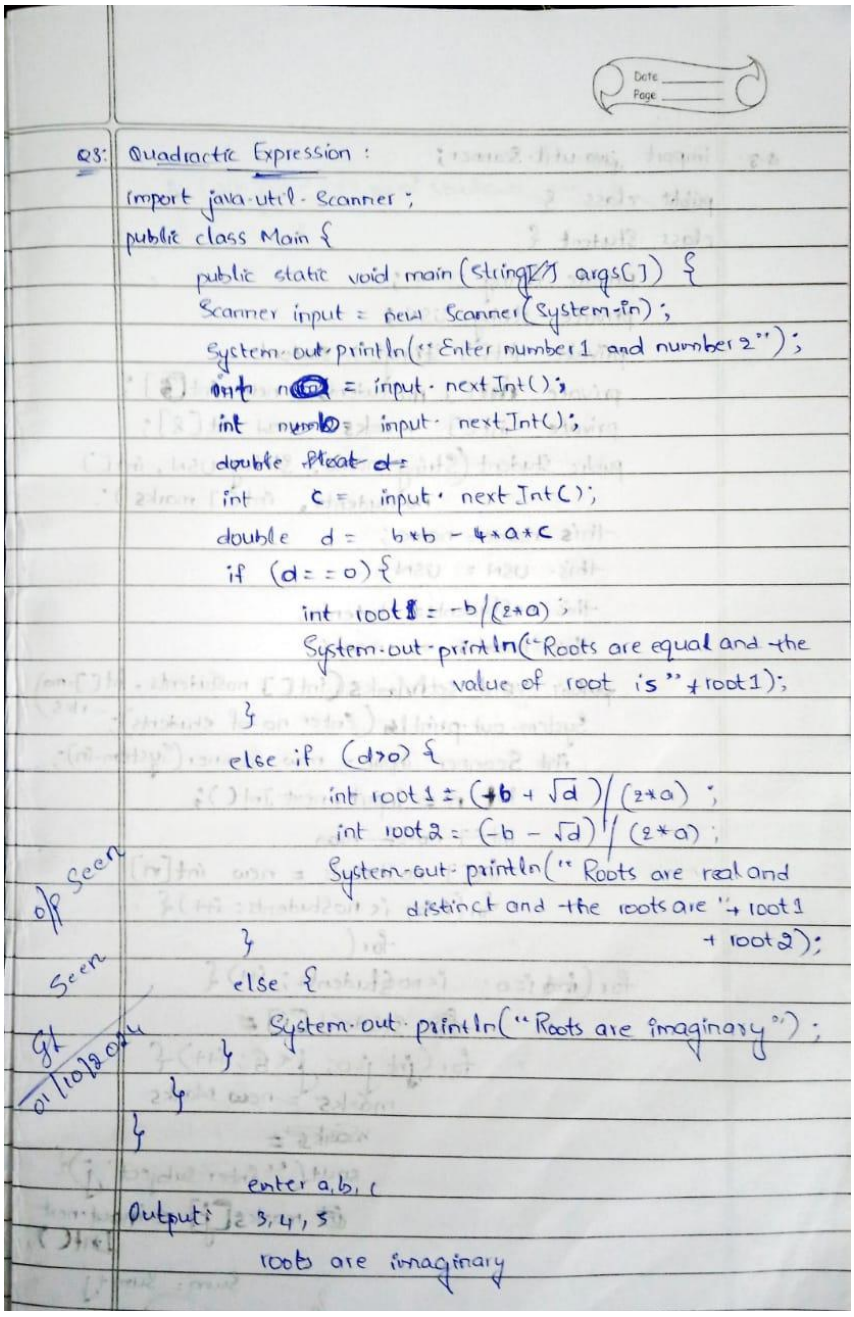
Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
murali1231944/JAVA_LAB

Program 1
Implement Quadratic Equation

Algorithm:



Code:

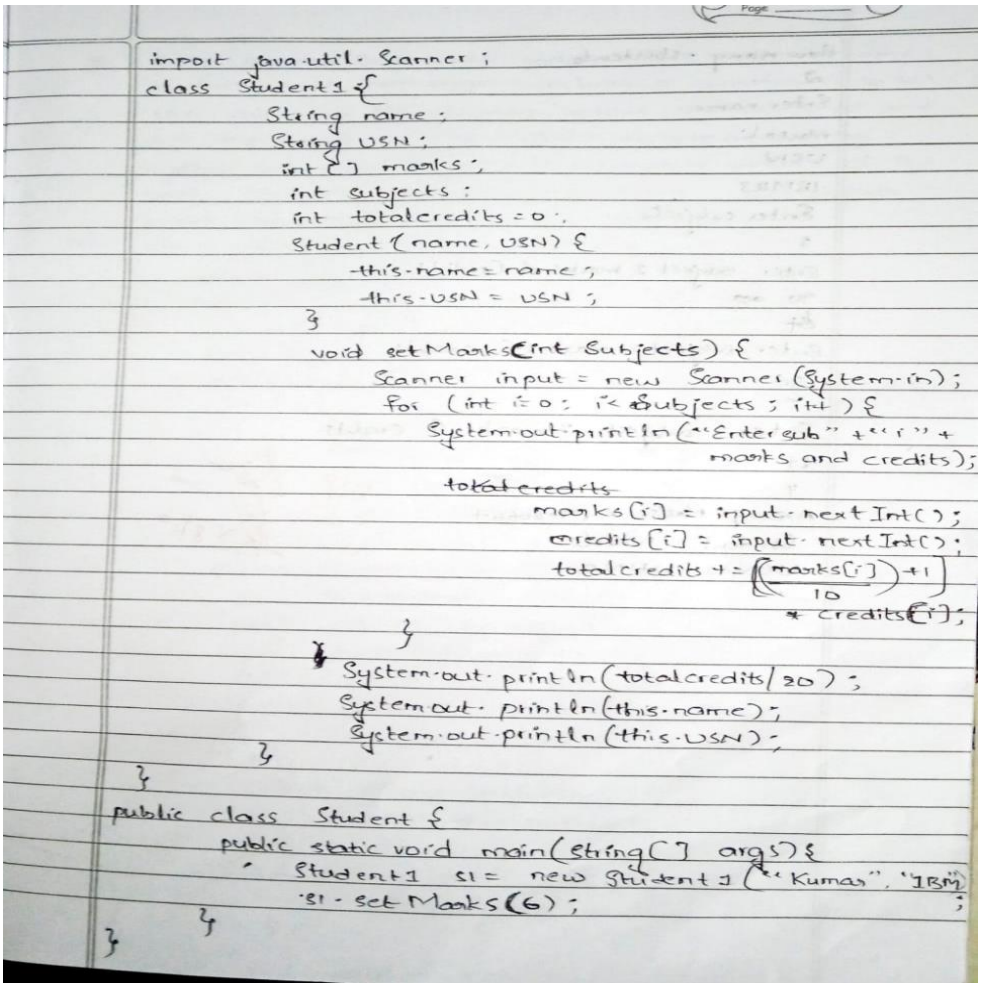
```
package LAB1;
import java.util.Scanner;
public class Quad{
public static void main(String [] args){
Scanner input = new Scanner(System.in);
System.out.println("enter a b c constants in the quad equaion");
int a = input.nextInt();
```

```
int b = input.nextInt();
int c = input.nextInt();
float det = b*b -4*a*c;
if (det < 0){
System.out.println("roots are imaginary");
}
else {
double root1 = (-b+Math.sqrt(det) )/ (2*a);
double root2 =( -b - Math.sqrt(det))/ (2*a);
if (det == 0){
System.out.println("roots are same ");
System.out.println("root 1 = " + root1 +"\n root2 -"+root2);
}
else{
System.out.println("roots are real and distinct ");
System.out.println("root 1 = " + root1 +"\n root2 -"+root2);
}

}
}
}
```

```
C:\Users\mural\OneDrive\Desktop\java_lab\LAB1>javac Quad.java

C:\Users\mural\OneDrive\Desktop\java_lab\LAB1>java Quad
Error: Could not find or load main class Quad
Caused by: java.lang.NoClassDefFoundError: LAB1/Quad (wrong name: Quad)

C:\Users\mural\OneDrive\Desktop\java_lab\LAB1>java Quad.java
enter a b c constants in the quad equaion
1
2
3
roots are imaginary

C:\Users\mural\OneDrive\Desktop\java_lab\LAB1>
```

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.



Code:

```
package LAB2;
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the number of students:");
      int n = sc.nextInt();
      sc.nextLine(); // Consume leftover newline

      Student[] students = new Student[n];
      for (int i = 0; i < n; i++) {
         System.out.println("Enter the name and the USN:");
         String name = sc.nextLine();
         String USN = sc.nextLine();

         System.out.println("Enter the number of subjects:");
         int n1 = sc.nextInt();
         int[] marks = new int[n1];
```

```
            int[] credits = new int[n1];

            for (int j = 0; j < n1; j++) {
                System.out.println("Enter the marks for subject " + (j + 1) + ":");
                marks[j] = sc.nextInt();
                System.out.println("Enter the credits for subject " + (j + 1) + ":");
                credits[j] = sc.nextInt();
            }
            sc.nextLine(); // Consume leftover newline

            students[i] = new Student(name, USN, credits, marks);
        }

        // Display SGPA for each student
        for (Student student : students) {
            student.s_gpa();
        }

        sc.close();
    }
}

class Student {
    String name;
    String USN;
    int[] credits;
    int[] marks;

    Student(String name, String USN, int[] credits, int[] marks) {
        this.name = name;
        this.USN = USN;
        this.credits = credits;
        this.marks = marks;
    }

    public void s_gpa() {
        int totalCredits = 0;
        int totalWeightedMarks = 0;

        for (int i = 0; i < credits.length; i++) {
            totalCredits += credits[i];
            totalWeightedMarks += marks[i] * credits[i];
        }

        double sgpa = (double) totalWeightedMarks / totalCredits;
        System.out.println("NAME = " + name +
                "\nUSN = " + USN +
                "\nSGPA = " + sgpa);
    }
}
```

Output:

```
C:\Users\mural\OneDrive\Desktop\java_lab\LAB2>javac Main.java

C:\Users\mural\OneDrive\Desktop\java_lab\LAB2>java Main.java
Enter the number of students:
1
Enter the name and the USN:
m
23
Enter the number of subjects:
3
Enter the marks for subject 1:
23
Enter the credits for subject 1:
2
Enter the marks for subject 2:
23
Enter the credits for subject 2:
2
Enter the marks for subject 3:
23
Enter the credits for subject 3:
2
NAME = m
USN = 23
SGPA = 23.0
```

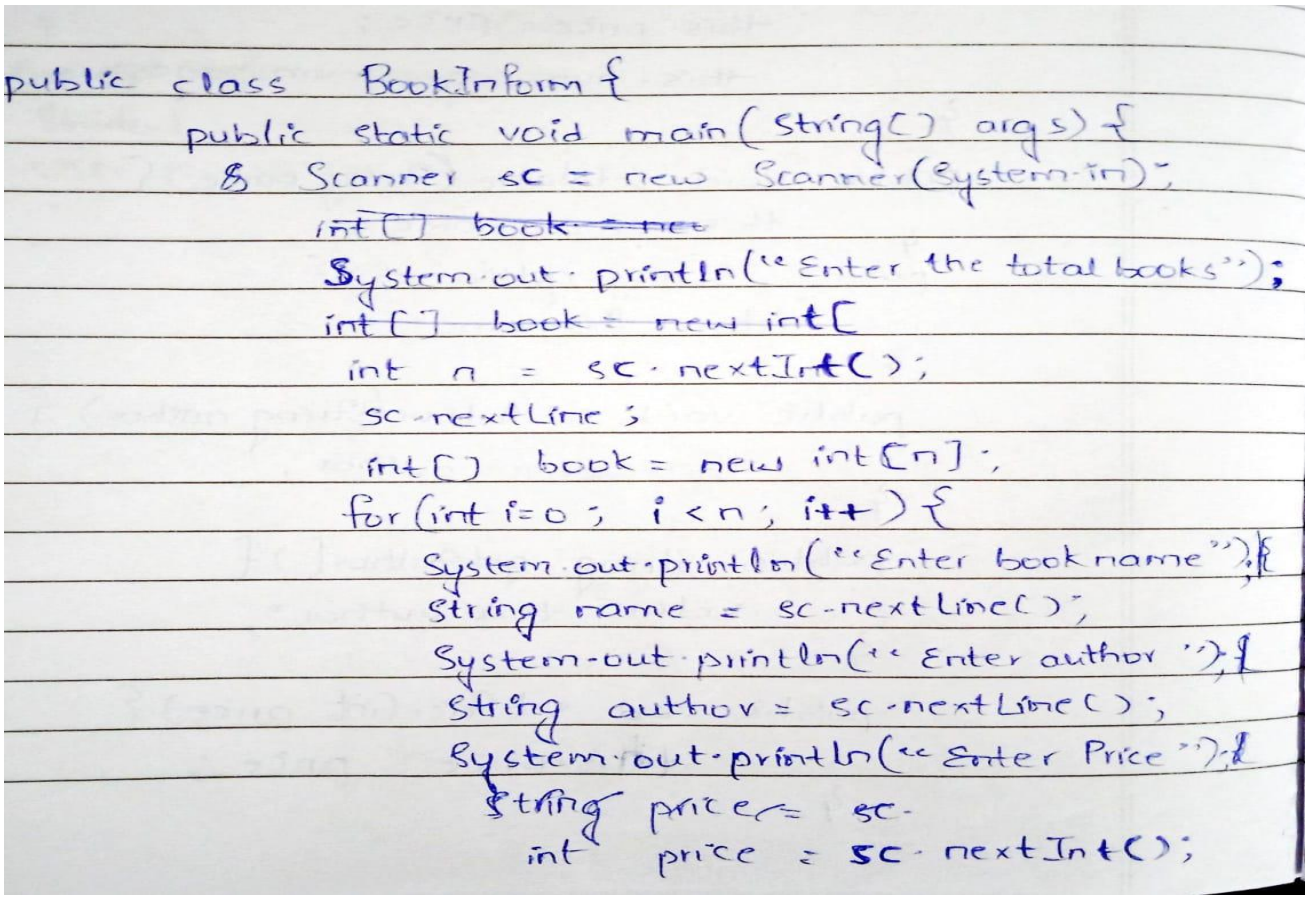
Program 3

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

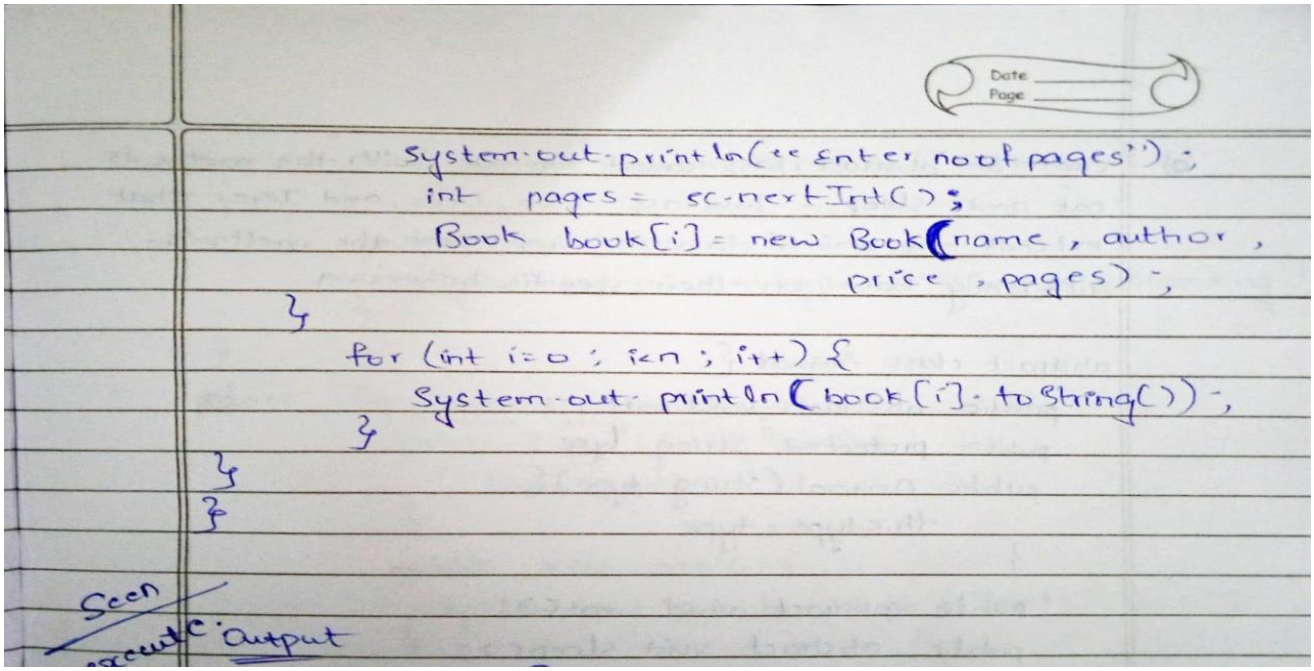
Algorithmn:

```
import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private int price;
    private int num_pages;
    public Book(String name, String author, int price, int
                num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return this.name;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getAuthor() {
        return this.author;
    }
    public void setPrice(int price) {
        this.price = price;
    }
```

```
    public int getPrice() {
        return this.price;
    }
    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }
    public int getNumPages() {
        return this.num_pages;
    }
    @Override
    public String toString() {
        return "Name = " + this.name + "\n" +
               "Author = " + this.author + "\n" +
               "Price = " + this.price + "\n" +
               "NumPages = " + this.num_pages;
    }
```



```
public class BookInform {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] book = new
        System.out.println("Enter the total books");
        int[] book = new int[
        int n = sc.nextInt();
        sc.nextLine;
        int[] book = new int[n];
        for(int i=0; i<n; i++) {
            System.out.println("Enter book name");
            String name = sc.nextLine();
            System.out.println("Enter author");
            String author = sc.nextLine();
            System.out.println("Enter Price");
            String price = sc.
            int price = sc.nextInt();
```



```
            System.out.println("Enter noof pages");
            int pages = sc.nextInt();
            Book book[i] = new Book(name, author,
                                    price, pages);
        }
        for (int i=0; i<n; i++) {
            System.out.println(book[i].toString());
        }
    }
}
```

Seen
Output

Code:

```
package LAB3;
```

```
import java.util.Scanner;
class Book{
    private String name;
    private String author;
    private int num_pages;
    private int price;
    public Book(String name,String author,int price,int num_pages){
        this.name =name;
        this.author =author;
        this.price =price;
        this.num_pages=num_pages;
    }
    public void setName(String name){
        this.name =name;
    }
    public void setAuthor(String author){
        this.author =author;
    }
    public void setPrice(int price){
        this.price =price;
    }
    public void setName(int num_pages){
        this.num_pages =num_pages;
    }
    public String getName(){
        return this.name;
    }
    public String getAuthor(){
        return this.author;
    }
    public int getPrice(){
        return this.price;
    }
    public int getNumPages(){
        return this.num_pages;
    }
    public String  toString(){
        return "Name= "+this.name+" Author= "+this.author+" Price= "+this.price+" Numpages=
"+this.num_pages;
}
}
public class BookInform{
    public static void main(String[] args){
        Scanner sc =new Scanner(System.in);
        System.out.println("Enter no of books");
        int n= sc.nextInt();
        sc.nextLine();
        Book[] book=new Book[n];
        for (int i=0;i<n;i++){
            System.out.println("Enter book name");
            String name=sc.nextLine();
            System.out.println("Enter author name");
            String author =sc.nextLine();
```

```
            System.out.println("Enter Price");
            int price = sc.nextInt();
            System.out.println("Enter no of book pages");
            int pages =sc.nextInt();
            book[i]=new Book(name,author,price,pages);
           sc.nextLine();

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

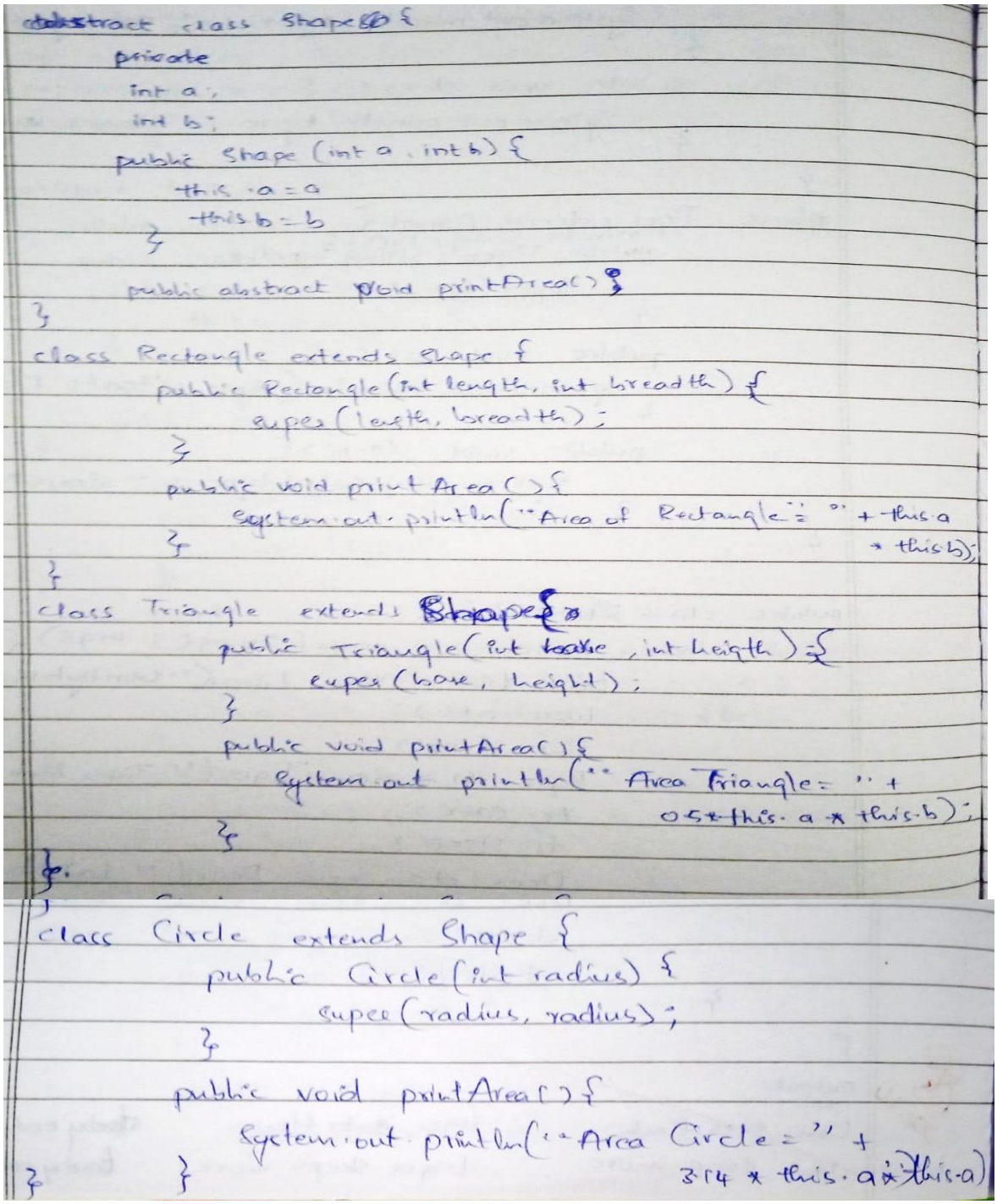
Output:

```
C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB3/BookInform.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB3.BookInform
Enter no of books
2
Enter book name
harrypotter
Enter author name
jk rowling
Enter Price
120
Enter no of book pages
679
Enter book name
lord of the rings
Enter author name
stephenwolf
Enter Price
456
Enter no of book pages
980
Name= harrypotter Author= jk rowling Price= 120 Numpages= 679
Name= lord of the rings Author= stephenwolf Price= 456 Numpages= 980
```

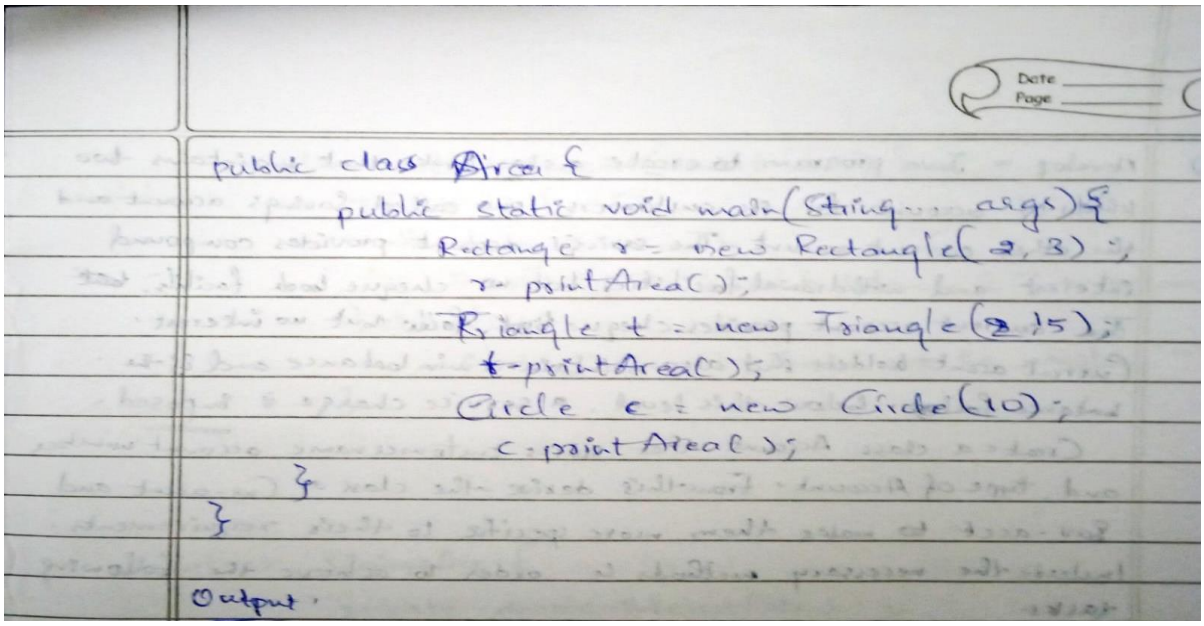
Program 4

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 contains only the method **printArea( )** that prints the area of the given shape

Algorithm:



```
abstract class Shape {
    private
    int a;
    int b;
    public Shape (int a, int b) {
        this.a = a
        this.b = b
    }
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(int length, int breadth) {
        super(length, breadth);
    }
    public void printArea() {
        System.out.println("Area of Rectangle = " + this.a
            * this.b);
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }
    public void printArea() {
        System.out.println("Area Triangle = " +
            0.5 * this.a * this.b);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, radius);
    }
    public void printArea() {
        System.out.println("Area Circle = " +
            3.14 * this.a * this.a);
    }
}
```



Code:

```
package LAB4;

abstract class Shape{
  int a;
  int b;
  public Shape(int a,int b){
     this.a =a;
     this.b =b;
  }
  public abstract void printArea();
}
class Triangle extends Shape{
  public Triangle(int base,int height){
     super(base,height);
  }

  public void printArea(){
     System.out.println("Area of the Triangle "+this.a*this.b*0.5);
  }
}
class Rectangle extends Shape{
  public Rectangle(int breadth,int lenght){
     super(breadth,lenght);
  }

  public void printArea(){
     System.out.println("Area of the Rectangle "+this.a*this.b);
  }
}
class Circle extends Shape{
  public Circle(int radius){
```

```
    super(radius,0);
  }

  public void printArea(){
    System.out.println("Area of the Circle "+this.a*this.a*3.14);
  }
}
public class Area{
  public static void main(String... args){
    Rectangle r = new Rectangle(2,4);
    r.printArea();
    Triangle t = new Triangle(2,5);
    t.printArea();
    Circle c= new Circle(5);
    c.printArea();
  }
}
```

Output:

```
C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB4/Area.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB4.Area
Area of the Rectangle 8
Area of the Triangle 5.0
Area of the Circle 78.5
```

Program 5

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

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

a) Accept deposit from customer and update the balance.

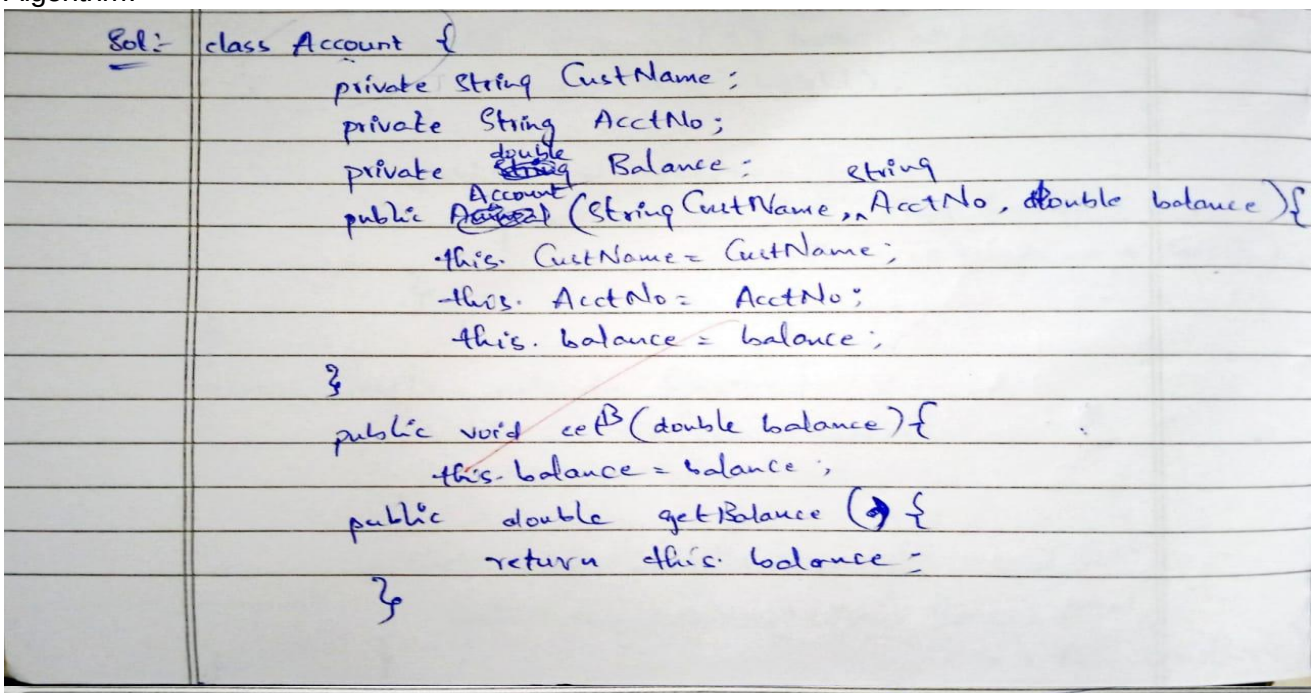
b) Display the balance.

c) Compute and deposit interest.

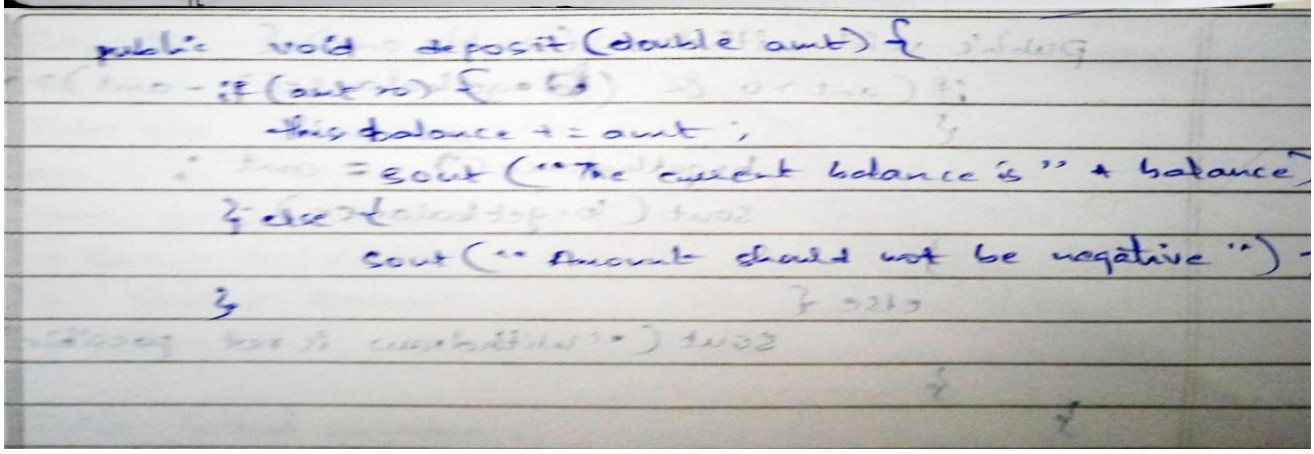
d) Permit withdrawal and update the balance.

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

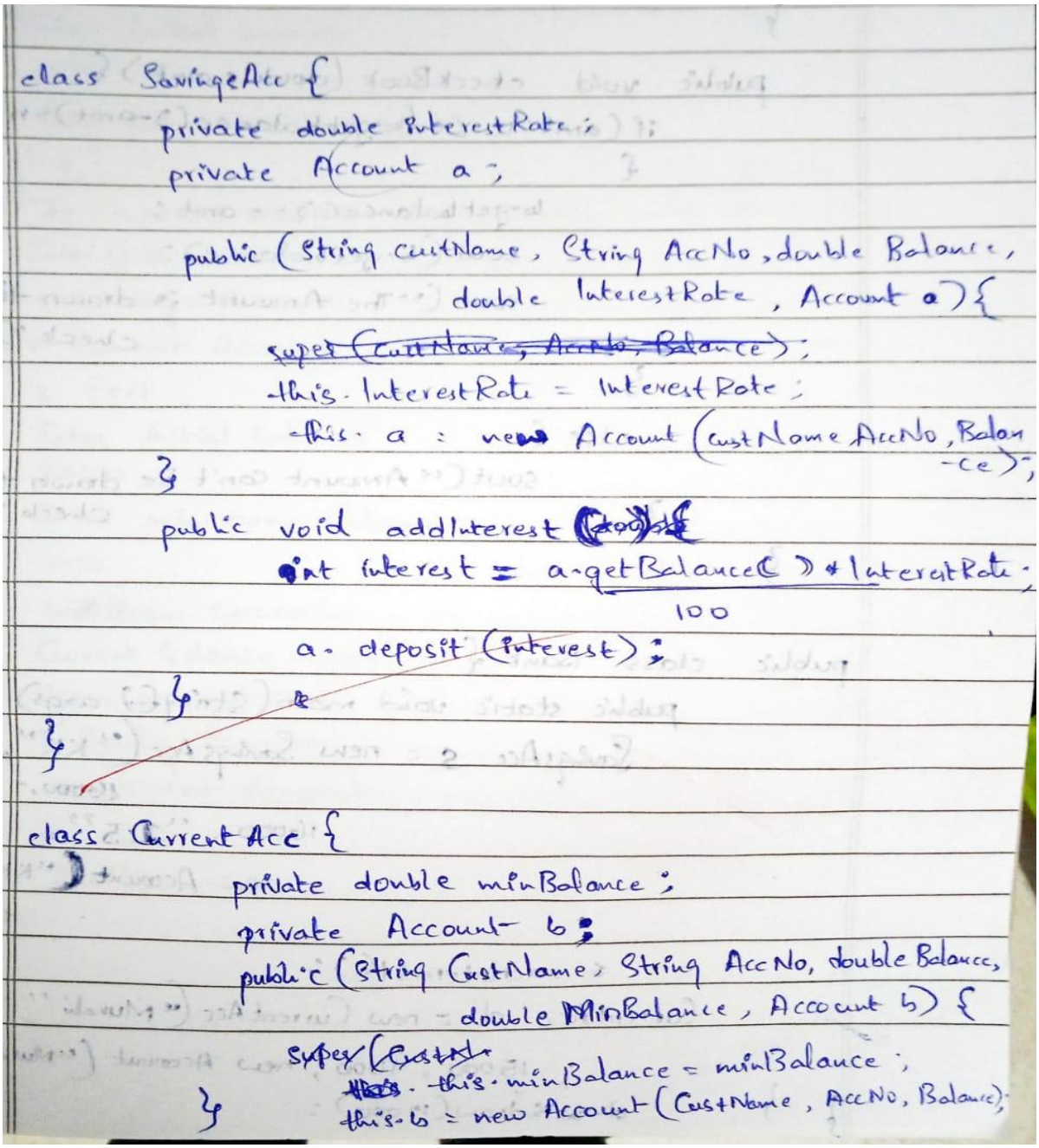
Algorithm:



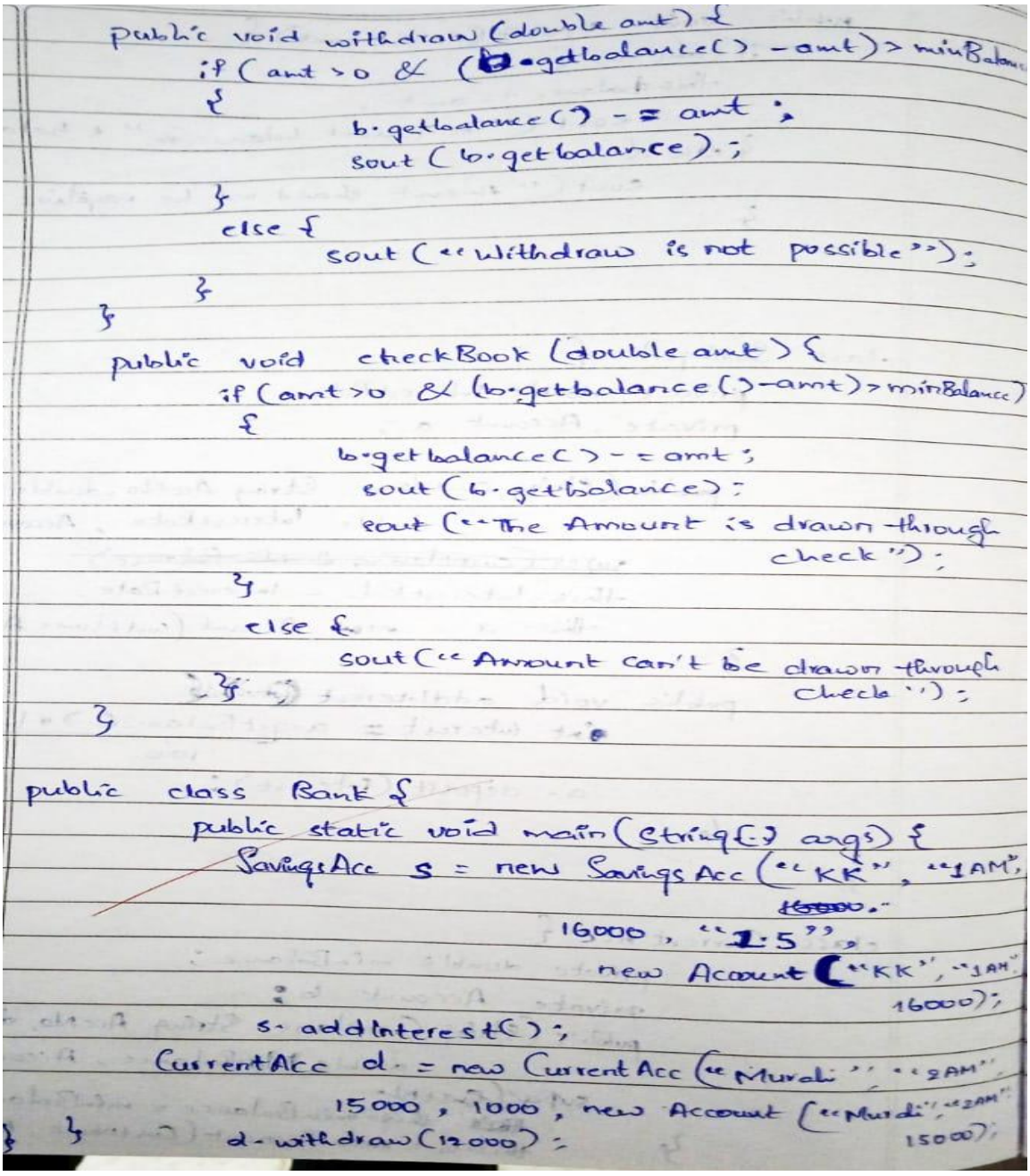
```
Sol:- class Account {
        private String CustName;
        private String AcctNo;
        private double Balance;
        public Account(String CustName, AcctNo, double balance){
            this.CustName = CustName;
            this.AcctNo = AcctNo;
            this.balance = balance;
        }
        public void setB(double balance){
            this.balance = balance;
        public double getBalance (){
            return this.balance;
        }
```



```
public void deposit(double amt) {
    if(amt>0){
        this.balance += amt;
        Sout("The current balance is" + balance)
    }else{
        Sout("Amount should not be negative")
    }
```



```
class SavingsAcc {
    private double InterestRate;
    private Account a;

    public (String custName, String AccNo, double Balance,
                double InterestRate, Account a) {
        super(custName, AccNo, Balance);
        this.InterestRate = InterestRate;
        this.a = new Account(custName AccNo, Balan
                                              -ce);
    }
    public void addInterest
        int interest = a.getBalance() * InterestRate;
                                    100
        a.deposit(interest);
    }
}

class Current Acc {
    private double minBalance;
    private Account b;
    public (String CustName, String AccNo, double Balance,
                double MinBalance, Account b) {
        super(
        this.minBalance = minBalance;
        this.b = new Account(CustName, AccNo, Balance);
    }
```



```
public void withdraw(double amt) {
    if(amt>0 & (b.getbalance() - amt) > minBalance
    {
        b.getbalance() -= amt;
        sout(b.getbalance);
    }
    else {
        sout("withdraw is not possible");
    }
}
public void checkBook(double amt) {
    if(amt>0 & (b.getbalance()-amt)>minBalance)
    {
        b.getbalance() -= amt;
        sout(b.getbalance);
        sout("The Amount is drawn through
                                    check");
    }
    else {
        sout("Amount can't be drawn through
                                    check");
    }
}

public class Bank {
    public static void main(String[] args) {
        SavingsAcc s = new SavingsAcc("KK", "1AM",
                                    16000, "1.5",
                                    new Account("KK", "1AM",
                                    16000);
        s.addInterest();
        CurrentAcc d = new CurrentAcc("Murali", "2AM",
                                    15000, 1000, new Account("Murali", "2AM",
                                    15000);
        d.withdraw(12000);
    }
}
```

Code:

```
package LAB5;

import java.util.Scanner;

class Account {
    private String custName;
    private String accNo;
    private double balance;

    public Account(String custName, String accNo, double balance) {
```

```
        this.custName = custName;
        this.accNo = accNo;
        this.balance = balance;
    }

    public double getBalance() {
        return this.balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            this.balance += amount;
            System.out.println("The current balance is " + this.balance);
        } else {
            System.out.println("Amount should not be negative");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= 0) {
            this.balance -= amount;
            System.out.println("Withdraw successful. Current balance: " + this.balance);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }
}

class SavingsAccount {
    private double interestRate;
    private Account account;

    public SavingsAccount(String custName, String accNo, double balance, double interestRate) {
        this.interestRate = interestRate;
        this.account = new Account(custName, accNo, balance);
    }

    public void addInterest() {
        double interest = account.getBalance() * this.interestRate;
        account.deposit(interest);
    }

    public Account getAccount() {
        return account;
    }
}

class CurrentAccount {
    private double minBalance;
    private Account account;

    public CurrentAccount(String custName, String accNo, double balance, double minBalance) {
        this.minBalance = minBalance;
```

```
        this.account = new Account(custName, accNo, balance);
    }

    public void withdraw(double amt) {
        if (amt > 0 && (account.getBalance() - amt) >= minBalance) {
            account.withdraw(amt);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }

    public Account getAccount() {
        return account;
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name:");
        String name = sc.nextLine();
        System.out.println("Enter the account number:");
        String accnt = sc.nextLine();

        while (true) {
            System.out.println("Enter your choice:");
            System.out.println("1. Savings Account");
            System.out.println("2. Current Account");
            System.out.println("3. Exit");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Enter initial balance:");
                    double savingsBalance = sc.nextDouble();
                    System.out.println("Enter the interest rate:");
                    double interestRate = sc.nextDouble();
                    SavingsAccount savingsAccount = new SavingsAccount(name, accnt, savingsBalance,
interestRate);
                    savingsAccount.addInterest();
                    break;

                case 2:
                    System.out.println("Enter initial balance:");
                    double currentBalance = sc.nextDouble();
                    System.out.println("Enter minimum balance:");
                    double minBalance = sc.nextDouble();
                    CurrentAccount currentAccount = new CurrentAccount(name, accnt, currentBalance,
minBalance);

                    System.out.println("entr the amount to be withdraw");
                    double q = sc.nextInt();
                    currentAccount.withdraw(q);
```

```
                System.out.println("Account created. Current balance: " +
currentAccount.getAccount().getBalance());
                break;

            case 3:
                System.out.println("Exiting...");
                sc.close();
                return;

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

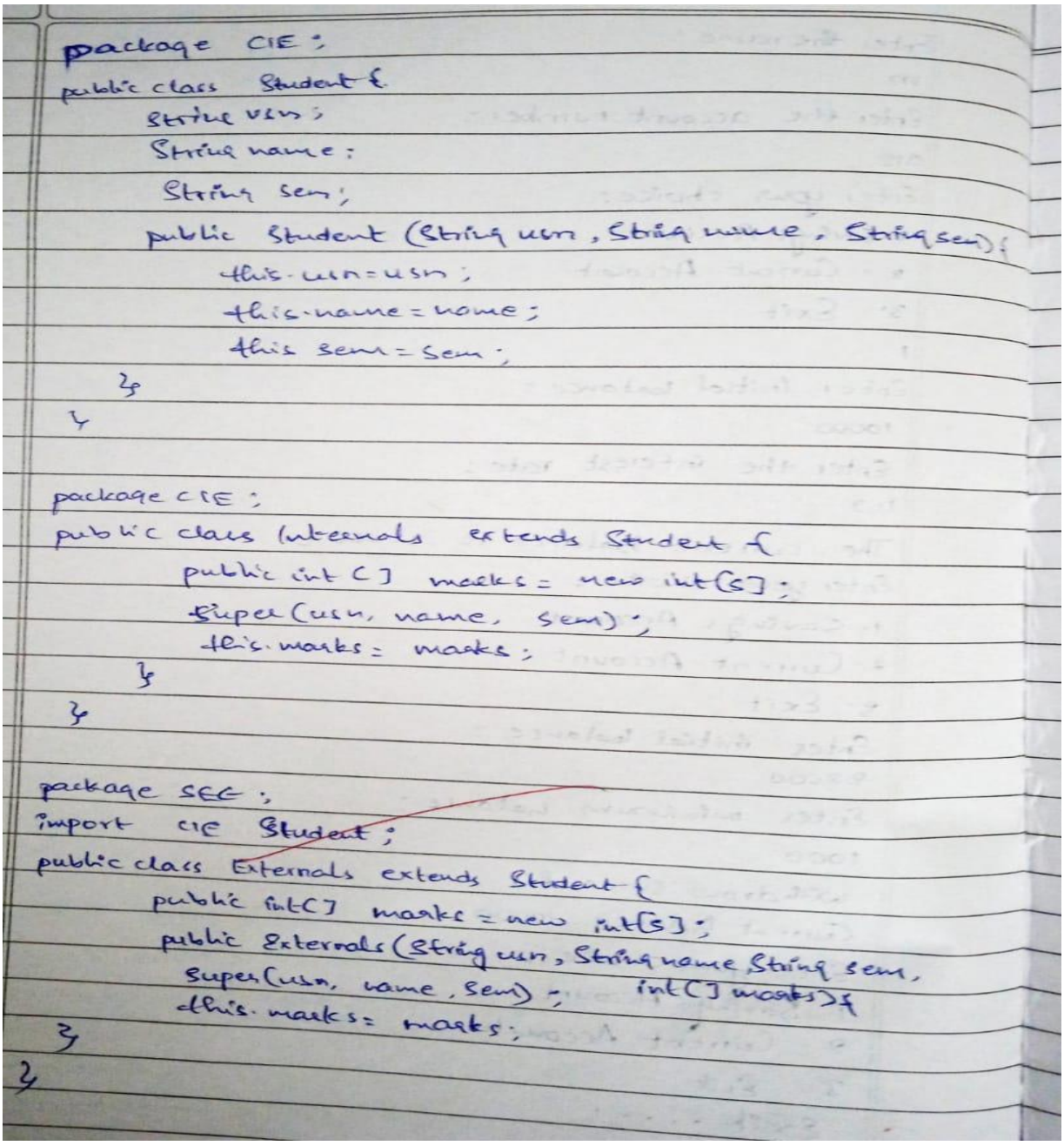
Output:

```
C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB5/Bank.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB5.Bank
Enter the name:
murali
Enter the account number:
123
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
1
Enter initial balance:
1000
Enter the interest rate:
2
The current balance is 3000.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
2
Enter initial balance:
3000
Enter minimum balance:
1000
entr the amount to be withdraw
2000
Withdraw successful. Current balance: 1000.0
Account created. Current balance: 1000.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
3
Exiting...
```

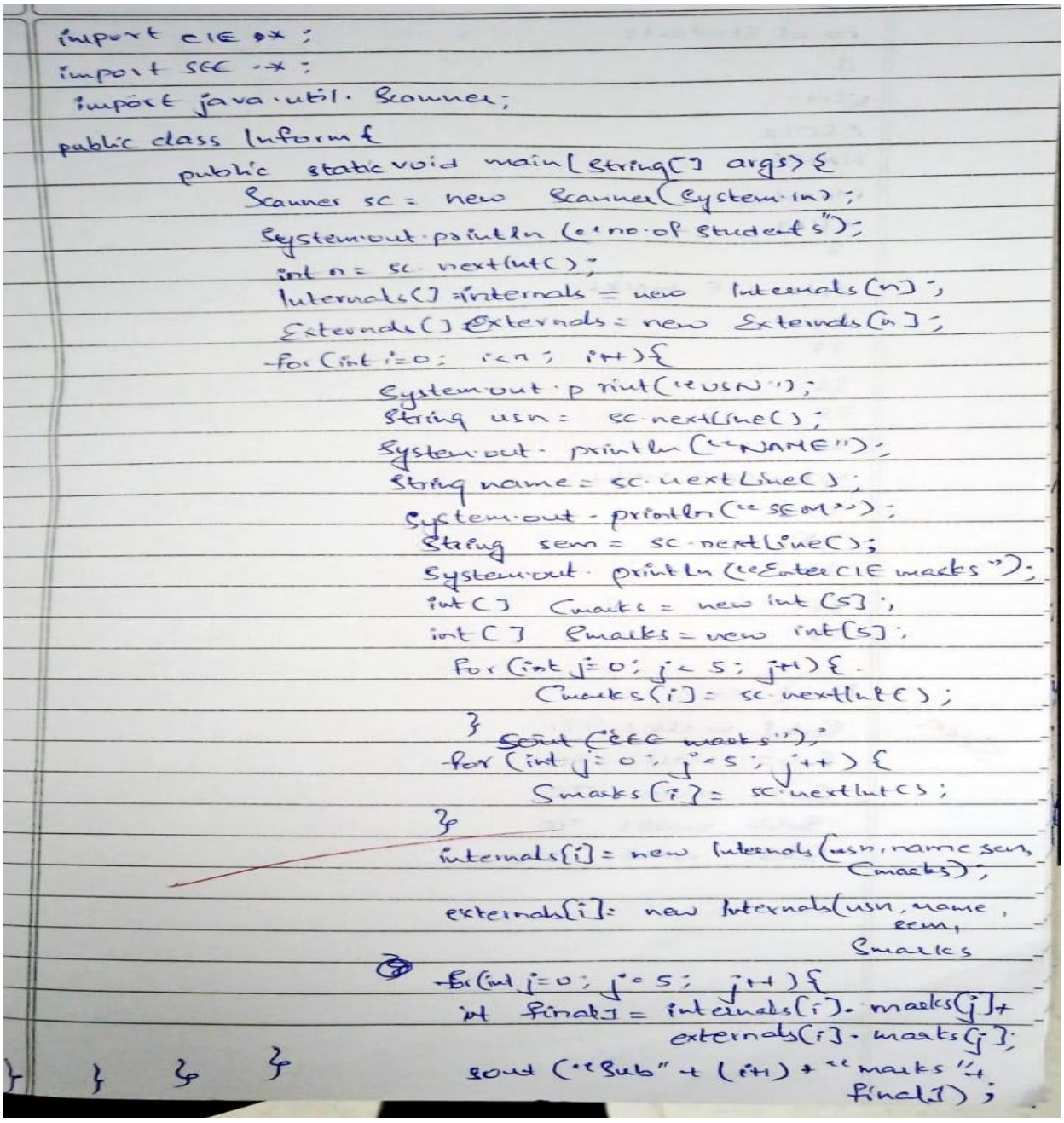
Program 6

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

Algorithm:



```
package CIE;
public class Student {
    String usn;
    String name;
    String sem;
    public Student (String usn, String name, String sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

package CIE;
public class Internals extends Student {
    public int[] marks = new int[5];
    super(usn, name, sem);
    this.marks = marks;
    }
}

package SEE;
import CIE.Student;
public class Externals extends Student {
    public int[] marks = new int[5];
    public Externals(String usn, String name, String sem,
    super(usn, name, sem);    int[] marks){
    this.marks = marks;
}
}
```



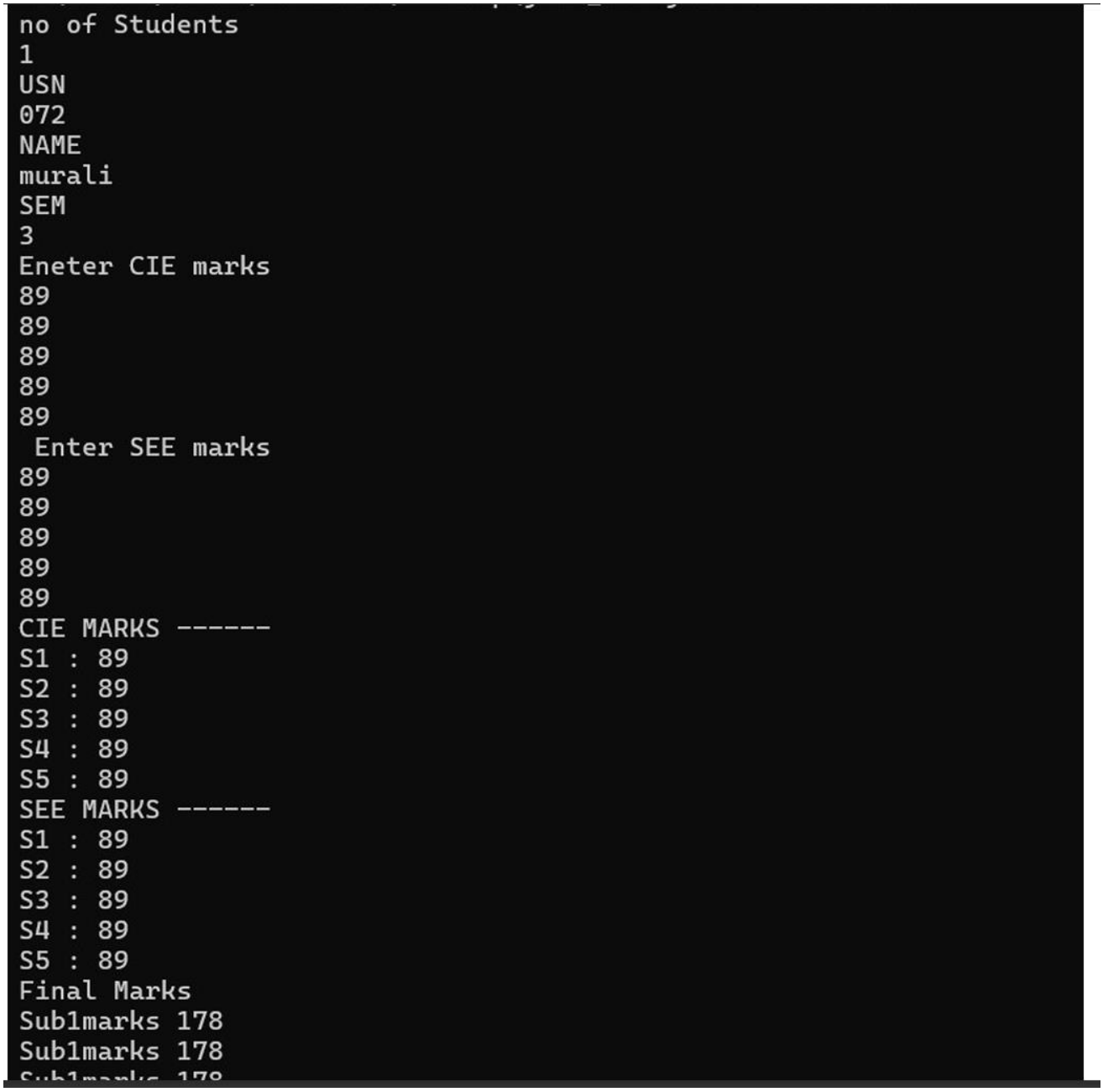
Code:

```
package LAB6.CIE;

public class Student1 {
    String usn;
    String name;
    String sem;

    public Student1(String usn, String name, String sem) {
        this.usn = usn;
        this.name = name;
```

```
        this.sem = sem;

    }
}

package LAB6.CIE;

public class Internals extends Student1 {
    public int[] marks = new int[5];

    public Internals(String usn, String name, String sem, int[] marks) {
        super(usn, name, sem);
        this.marks=marks;

        System.out.println("CIE MARKS ------" +
            "\nS1 : " + this.marks[0] +
            "\nS2 : " + this.marks[1] +
            "\nS3 : " + this.marks[2] +
            "\nS4 : " + this.marks[3] +
            "\nS5 : " + this.marks[4]);
    }
}

package LAB6.SEE;
import LAB6.CIE.Student1;
public class Externals extends Student1{
  public int[] marks=new int[5];
    public Externals(String usn,String name,String sem,int[] marks){
        super(usn,name,sem);
        this.marks =marks;
        System.out.println("SEE MARKS ------"+
        "\nS1 : "+this.marks[0]+
        "\nS2 : "+this.marks[1]+
        "\nS3 : "+this.marks[2]+
        "\nS4 : "+this.marks[3]+
        "\nS5 : "+this.marks[4]);
    }
}


package LAB6;
import LAB6.CIE.*;
import LAB6
.SEE.*;
import java.util.Scanner;
public class Inform{
    public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      System.out.println("no of Students");
```

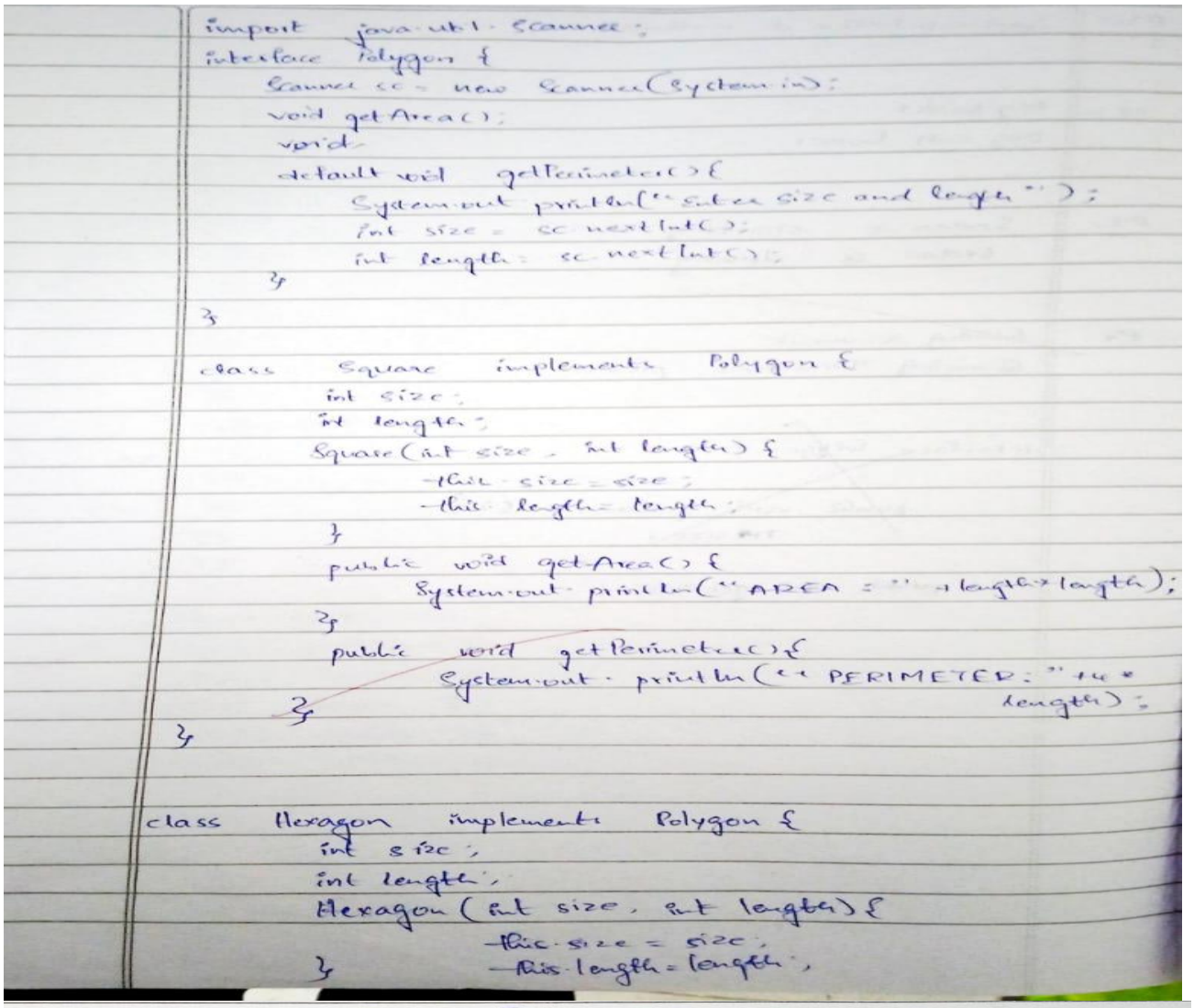
```
      int n=sc.nextInt();
      Internals[] internals=new Internals[n];
      Externals[] externals =new Externals[n];
      for(int i=0;i<n;i++){
        System.out.println("USN");
        String usn = sc.nextLine();
sc.next();

        System.out.println("NAME");
        String name=sc.nextLine();
sc.next();
        System.out.println("SEM");

        String sem =sc.nextLine();
        sc.next();
        System.out.println("Eneter CIE marks");
        int[] Cmarks=new int[5];
        int[] Smarks=new int[5];
        for(int j=0;j<5;j ++){
          Cmarks[j]=sc.nextInt();
        }
        System.out.println(" Enter SEE marks");
        for(int j=0;j<5;j++){
           Smarks[j]=sc.nextInt();
        }

        internals[i]=new Internals(usn,name,sem,Cmarks);
        externals[i]=new Externals(usn,name,sem,Smarks);
        System.out.println("Final Marks");
        for(int j=0;j<5;j++){
           int final1=internals[i].marks[j]+externals[i].marks[j];
           System.out.println("Sub"+(i+1)+"marks  "+final1);
        }
      }
    }
  }
```

Output:

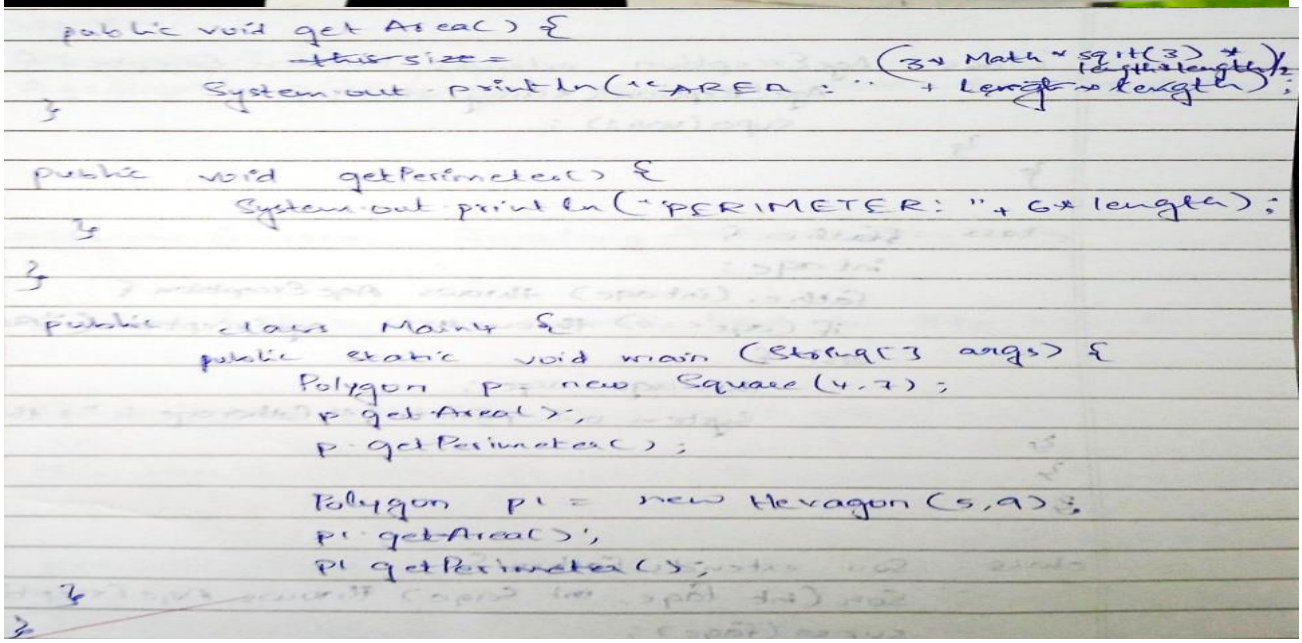


Program 7

We have created an interface named Polygon. It includes a default method `getPerimeter()` and an abstract method `getArea()`. We can calculate the perimeter of all polygons in the same manner so we implemented the body of `getPerimeter()` in Polygon. Now, all polygons that implement Polygon can use `getPerimeter()` to calculate the perimeter. However, the rule for calculating the area is different for different polygons. Hence, `getArea()` is included without implementation. Any class that implements Polygon must provide an implementation of `getArea()`.



```
import java.util.Scanner;
interface Polygon {
    Scanner sc = new Scanner(System.in);
    void getArea();
    void
    default void getPerimeter(){
        System.out.println("Enter size and length");
        int size = sc.nextInt();
        int length = sc.nextInt();
    }
}

class Square implements Polygon {
        int size;
        int length;
        Square(int size, int length) {
            this.size = size;
            this.length = length;
        }
        public void getArea() {
            System.out.println("AREA : " + length*length);
        }
        public void getPerimeter(){
            System.out.println("PERIMETER: " + 4*
                                length);
        }
}


class Hexagon implements Polygon {
        int size;
        int length;
        Hexagon(int size, int length){
            this.size = size;
            this.length = length;
        }
```



```
public void getArea() {
    this.size =
    System.out.println("AREA : " + (3* Math.sqrt(3) * length*length)/2
                                   + length*length);
}

public void getPerimeter() {
    System.out.println("PERIMETER: " + 6* length);
}
}

public class Main {
    public static void main(String[] args) {
        Polygon p = new Square(4,7);
        p.getArea();
        p.getPerimeter();

        Polygon p1 = new Hexagon(5,9);
        p1.getArea();
        p1.getPerimeter();
    }
}
```

Code:

```
package LAB7;

import java.util.Scanner;

interface Polygon {
    Scanner sc = new Scanner(System.in);

    void getArea();

    default void getPerimeter() {
        System.out.println("Enter the size and enter the length");
        int size = sc.nextInt();
        int length = sc.nextInt();
        System.out.println("Perimeter:  " + size * length);
    }
}

class Square implements Polygon {
    int size;
    int length;

    Square(int size, int length) {
        this.size = size;
        this.length = length;
    }


    public void getArea() {
        System.out.println("AREA: " + length * length);
    }


    public void getPerimeter() {
        System.out.println("Perimeter:  " + 4 * length);
    }
}

class Hexagon implements Polygon {
    int size;
    int length;

    Hexagon(int size, int length) {
        this.size = size;
        this.length = length;
    }


    public void getArea() {
        System.out.println("AREA: " + (3 * Math.sqrt(3) * length * length) / 2);
```

```
    }


    public void getPerimeter() {
        System.out.println("Perimeter:  " + 6 * length);
    }
}

public class Main4 {
    public static void main(String[] args) {
        Polygon p = new Square(4, 7);
        p.getArea();
        p.getPerimeter();

        Polygon p1 = new Hexagon(5, 9);
        p1.getArea();
        p1.getPerimeter();
    }
}
```

Output:

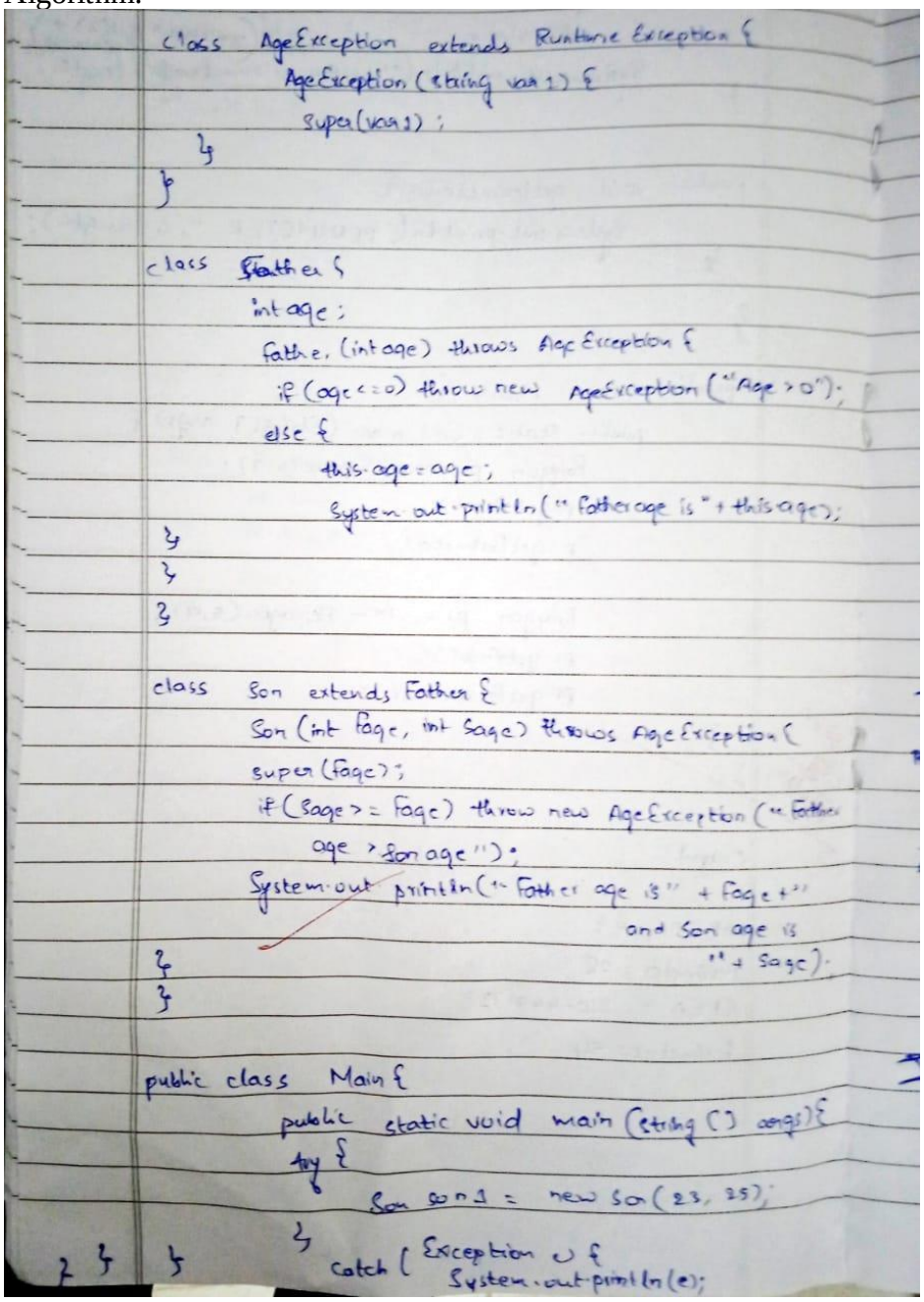
```
C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB7/Main4.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB7.Main4
AREA: 49
Perimeter: 28
AREA: 210.4441731196186
Perimeter: 54

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB7/Main.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB7.Main
Dog barks
Dog eats bones

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB7/Main1.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB7.Main1
Sedan is starting
Sedan is driving

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB7/Main2.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB7.Main2
Dog barks
Dog eats bones

C:\Users\mural\OneDrive\Desktop\java_lab>
```

Program 8

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class

called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that uses both father and son’s age and throws an exception if son’s age is >=father’s age.

Algorithm:



Code:

```
package LAB8;
```

```
class AgeException extends RuntimeException{
  AgeException(String msg){
  super(msg);
  }
}
class Father{
  int age;
  Father(int age) throws AgeException{
      if (age<=0) throw new AgeException("Age should be greater than zero");
      else{
          this.age=age;
          System.out.println("Father age is "+this.age);
}
}
}
class Son extends Father{

  Son(int Fage,int Sage)throws AgeException{
  super(Fage);
  if(Sage>=Fage) throw new AgeException("Father age should be greater than Son age");
  System.out.println("Father age is "+ Fage+ " and Son age is "+Sage);
}
}
public class Main{
  public static void main(String[] args){
      try{

          Son son1 =new Son(23,25);

      }
      catch(Exception e){
          System.out.println(e);

}
}
}
```

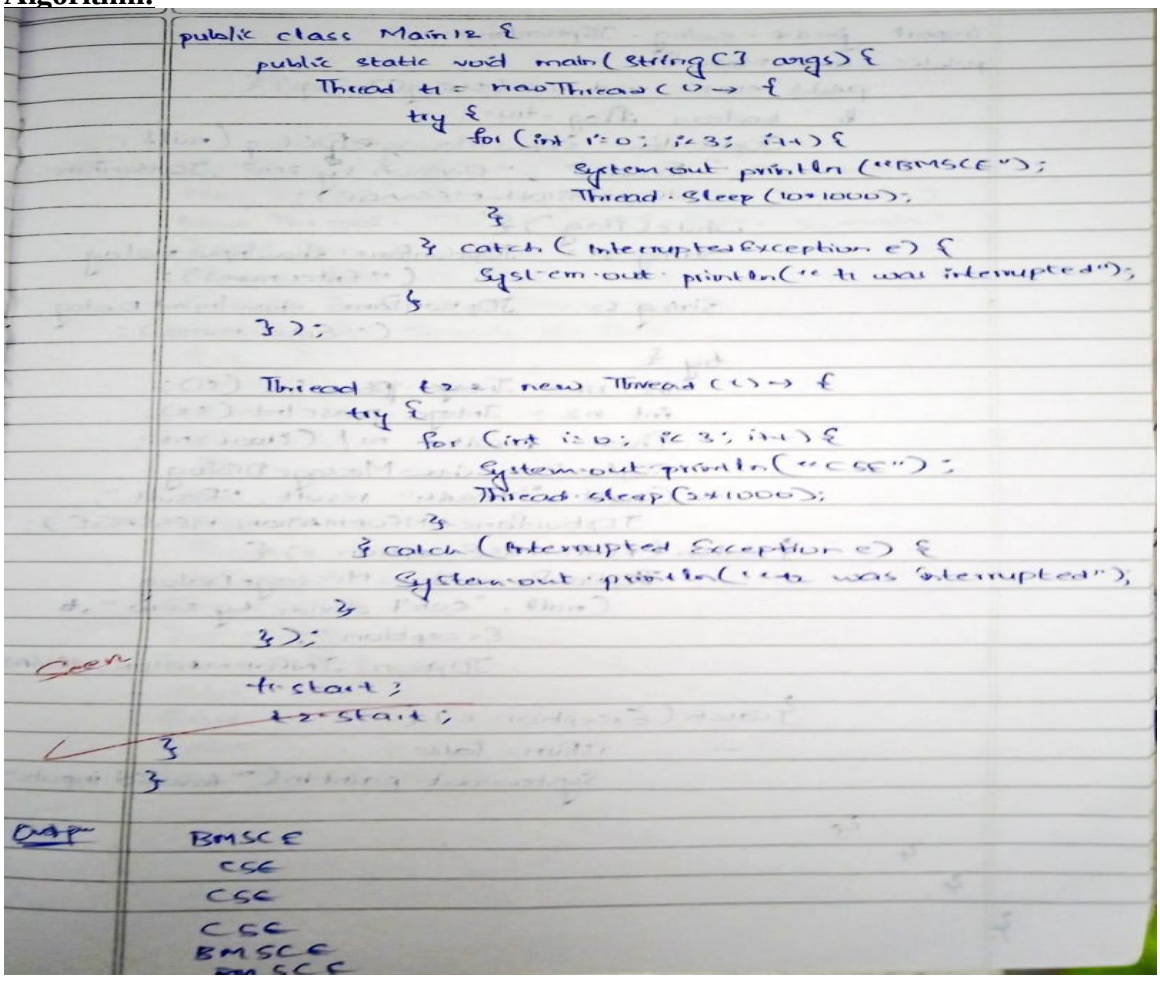
Output:

```
C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB8/E1.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB8.E1
File test.txt is missing

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB8/E3.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB8.E3
java.lang.ArithmeticException: / by zero

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB8/E4.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB8.E4
/ by zero

C:\Users\mural\OneDrive\Desktop\java_lab>javac LAB8/Main.java

C:\Users\mural\OneDrive\Desktop\java_lab>java LAB8.Main
Father age is 23
LAB8.AgeException: Father age should be greater than Son age

C:\Users\mural\OneDrive\Desktop\java_lab>
```

Program 7

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:



Code:

```
package LAB9;

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

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

public class TwoThreads {
  public static void main(String[] args) {
     BMSCollegeThread bmsThread = new BMSCollegeThread();
     CSEThread cseThread = new CSEThread();

     bmsThread.start();
     cseThread.start();  }
}
```

Output:

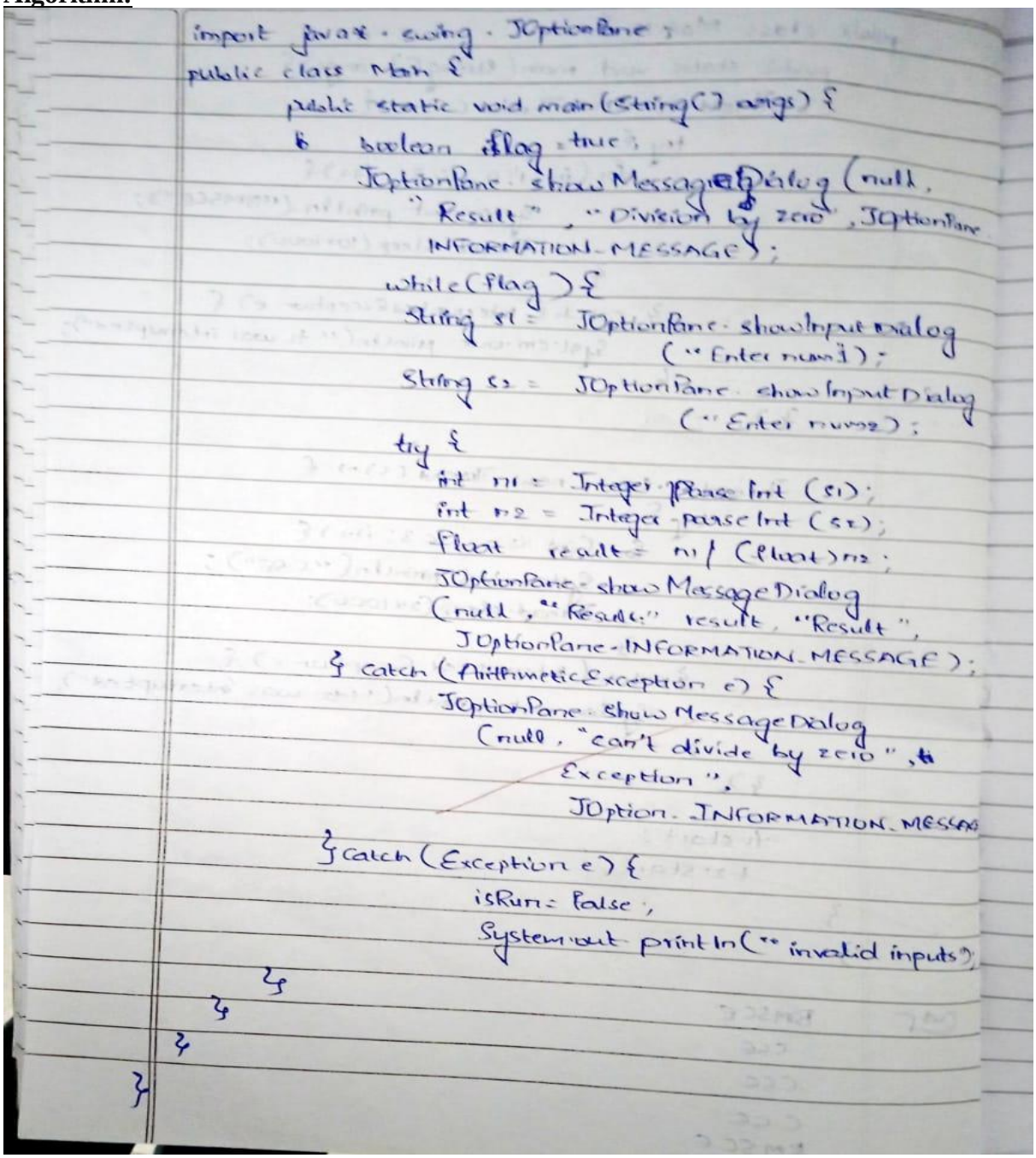
```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

Program 8

Write a program that creates a user interface to perform integer divisions. The user enters two

numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Algorithm:



Code:

```
package LAB10;

import javax.swing.JOptionPane;
```

```
public class add {
    public static void main(String[] args) {
        boolean isRun = true;
        JOptionPane.showMessageDialog(null, "Caluclator", "Division by zero",
JOptionPane.INFORMATION_MESSAGE);
        while (isRun) {
            String s1 = JOptionPane.showInputDialog("enter a no ");
            String s2 = JOptionPane.showInputDialog("enter second no");
            try {
                int n1 = Integer.parseInt(s1);
                int n2 = Integer.parseInt(s2);
                float result = n1 / (float) n2;
                JOptionPane.showMessageDialog(null, "result :" + result, "Result",
JOptionPane.INFORMATION_MESSAGE);
            } catch (ArithmeticException e) {
                JOptionPane.showMessageDialog(null, "cant divide by zero  :", "  Exception ",
                        JOptionPane.INFORMATION_MESSAGE);

            } catch (Exception e) {
                isRun = false;
                System.out.println("invalid inputs ");
            }
        }

    }
}
```

Output: