VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT

on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

AFEEFAH (1BM23CS015)

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 **AFEEFAH (1BM23CS015),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

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

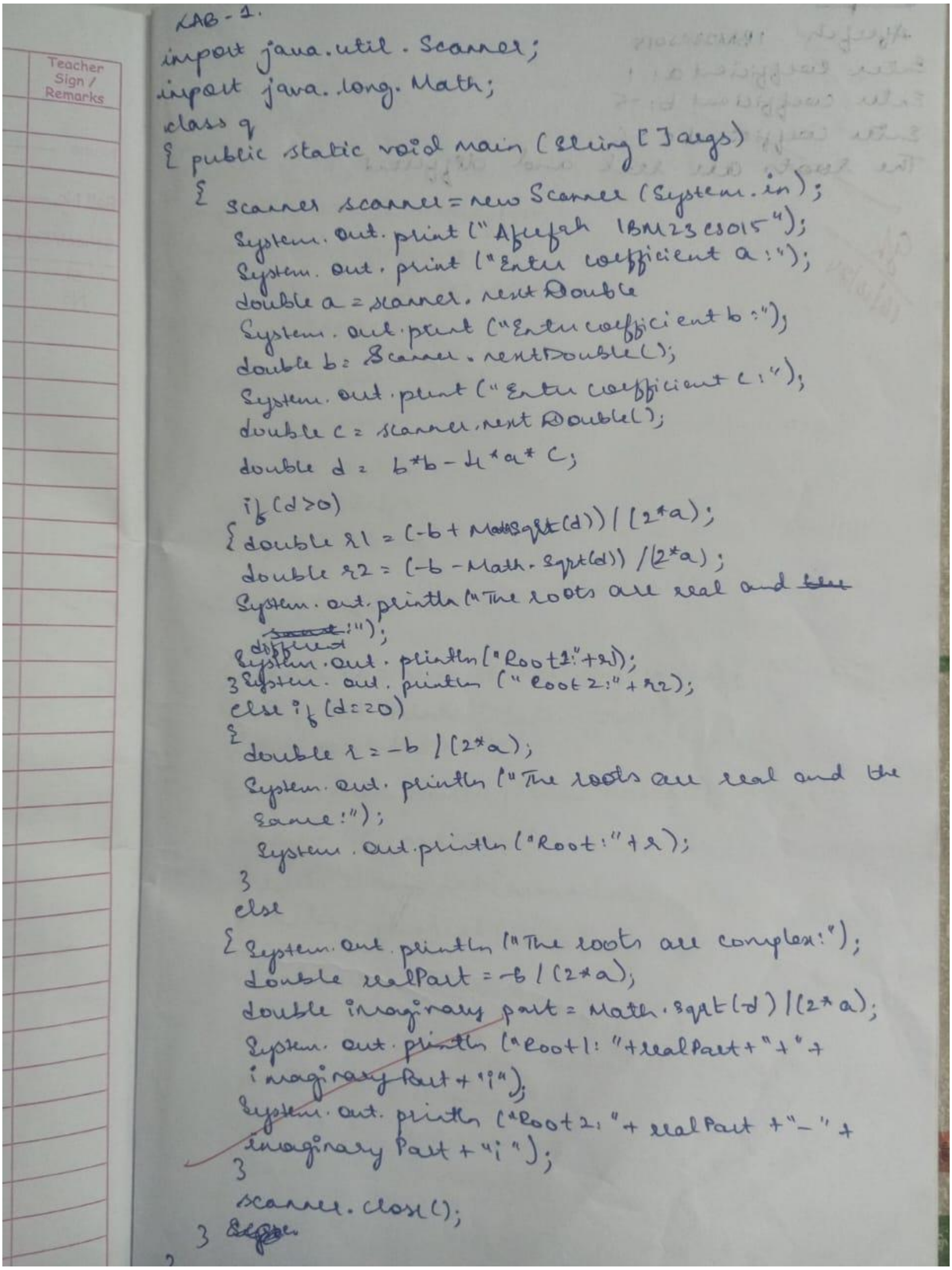
Index

Github Link:

https://github.com/Afeefahafeefah/java-lab

Program 1

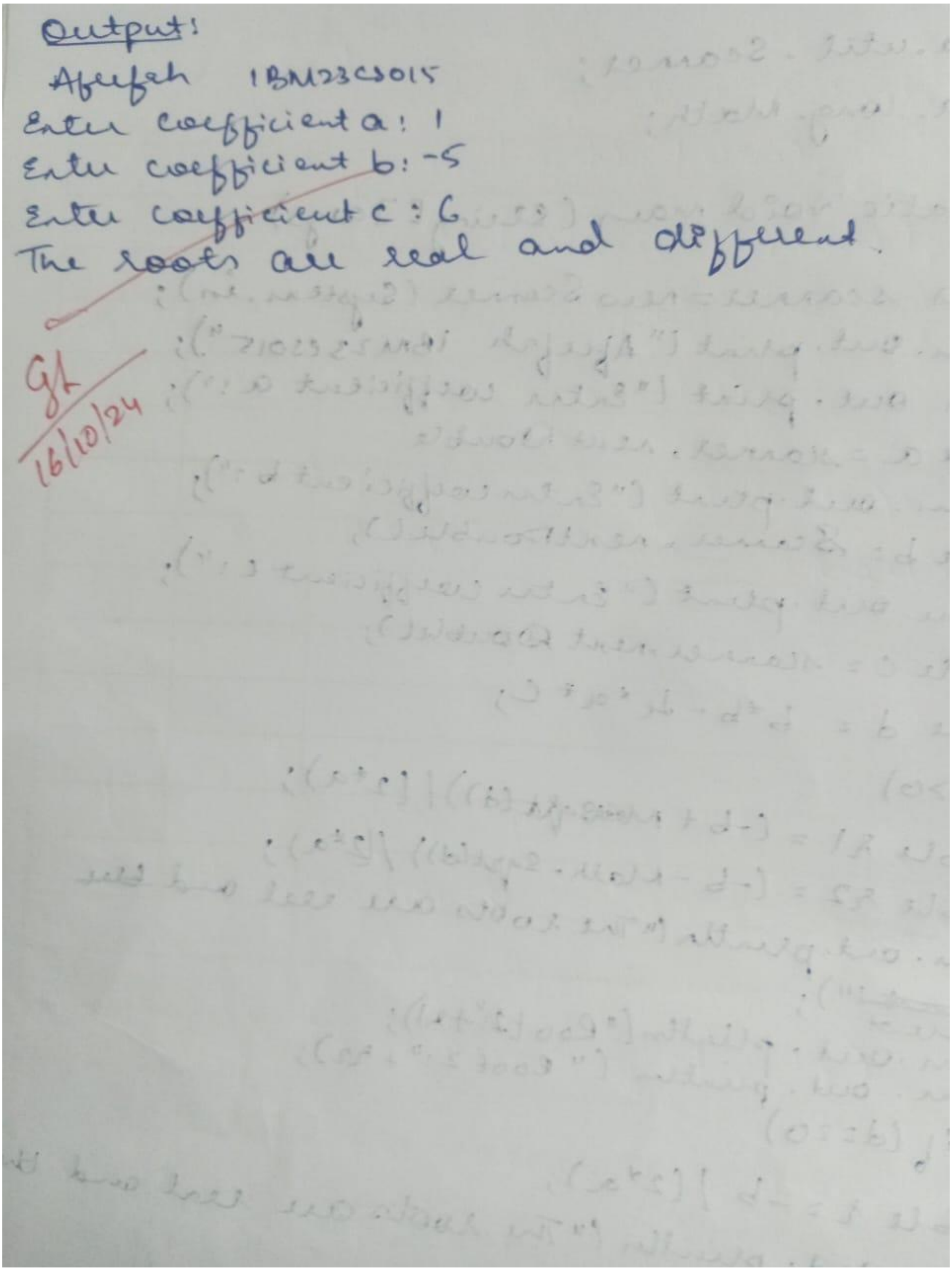
Implement Quadratic Equation



Teacher Sign / Remarks

LAB - 1.

```
import java.util.Scanner;
import java.long.Math;
class q
{ public static void main (String[]args)
  {
   Scanner scanner=new Scanner(System.in);
   System.out.print("Afeefah 1BM23CS015");
   System.out.print("Enter coefficient a:");
   double a = scanner.nextDouble
   System.out.print("Enter coefficient b:");
   double b= Scanner.nextDouble();
   System.out.print("Enter coefficient c:");
   double c= scanner.nextDouble();
   double d = b*b-4*a*c;
   if(d>0)
  {double r1 = (-b+MathSqrt(d))/(2*a);
   double r2 = (-b-Math.Sqrt(d))/(2*a);
   System.out.println("The roots are real and ~~the~~
     ~~same~~ different!");
   System.out.println("Root1:"+r1);
  }System.out.println("Root2:"+r2);
  else if(d==0)
  {
   double r=-b/(2*a);
   System.out.println("The roots are real and the
    same!");
    System.out.println("Root:"+r);
   }
  else
  {System.out.println("The roots are complex:");
   double realPart=-b/(2*a);
   double imaginary part=Math.sqrt(-d)/(2*a);
   System.out.println("Root1: "+realPart+"+"+
   imaginaryPart+"i");
   System.out.println("Root2: "+realPart+"-"+
   imaginaryPart+"i");
  }
  scanner.close();
 }
}
```



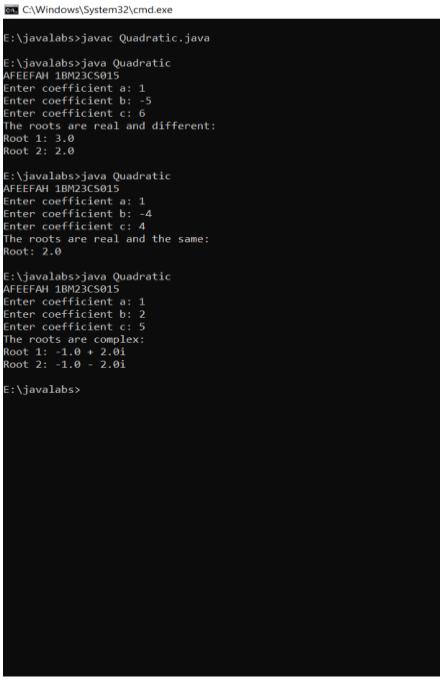
Output:

```
Afeefah    1BM23CS015
Enter coefficient a: 1
Enter coefficient b: -5
Enter coefficient c : 6
The roots are real and different.
```

GL
16/10/24

Code:

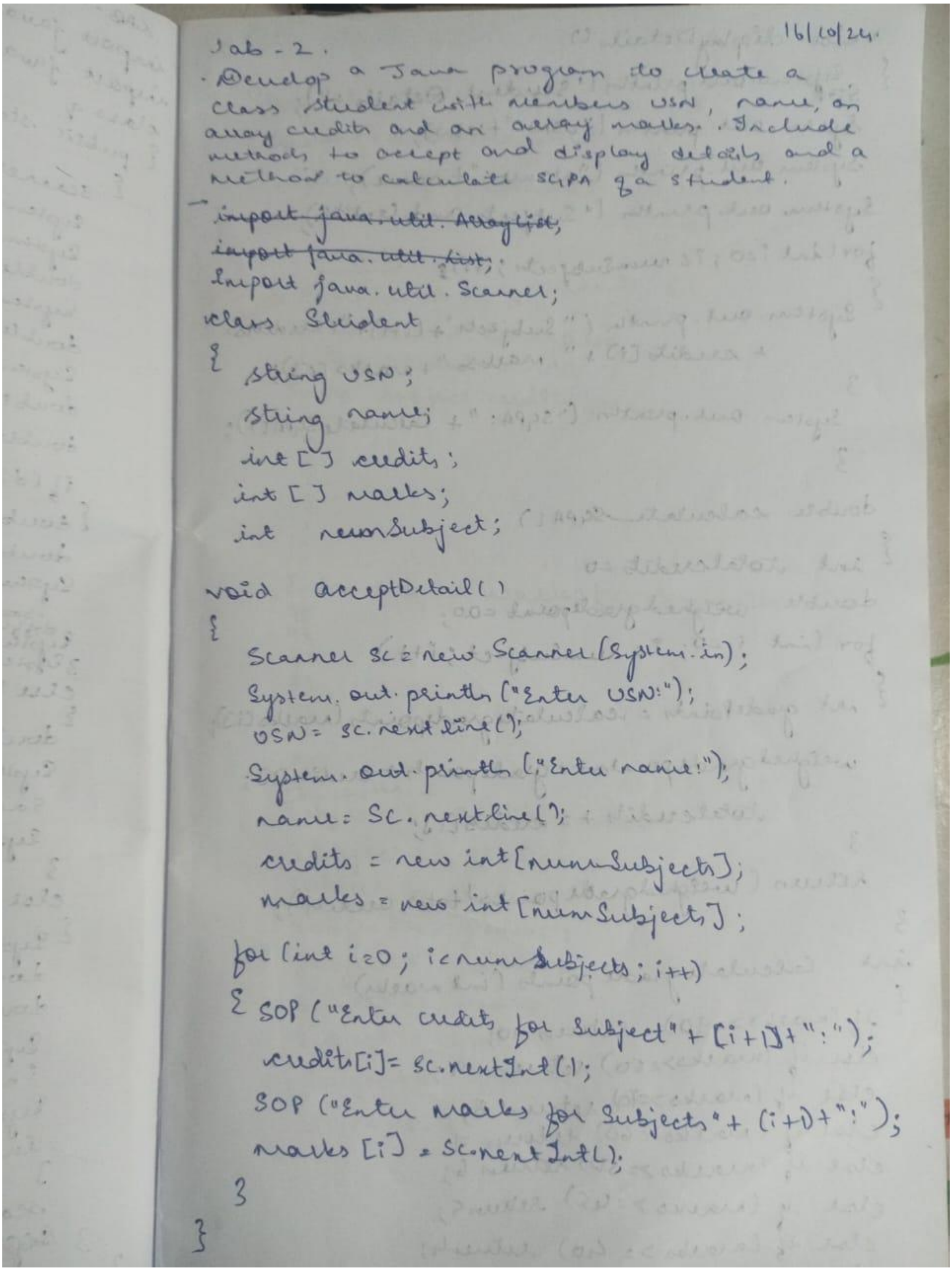
```
import java.util.Scanner;
import java.lang.Math;
class Quadratic
 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("AFEEFAH 1BM23CS015");
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

        double d = b * b - 4 * a * c;

        if (d > 0) {
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots are real and different:");
            System.out.println("Root 1: " + r1);
            System.out.println("Root 2: " + r2);
        } else if (d == 0) {
            double r = -b / (2 * a);
            System.out.println("The roots are real and the same:");
            System.out.println("Root: " + r);
        } else {
            System.out.println("The roots are complex:");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-d) / (2 * a);
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
        }

        scanner.close();

    }
}
```



```
E:\javalabs>javac Quadratic.java

E:\javalabs>java Quadratic
AFEEFAH 1BM23CS015
Enter coefficient a: 1
Enter coefficient b: -5
Enter coefficient c: 6
The roots are real and different:
Root 1: 3.0
Root 2: 2.0

E:\javalabs>java Quadratic
AFEEFAH 1BM23CS015
Enter coefficient a: 1
Enter coefficient b: -4
Enter coefficient c: 4
The roots are real and the same:
Root: 2.0

E:\javalabs>java Quadratic
AFEEFAH 1BM23CS015
Enter coefficient a: 1
Enter coefficient b: 2
Enter coefficient c: 5
The roots are complex:
Root 1: -1.0 + 2.0i
Root 2: -1.0 - 2.0i

E:\javalabs>
```

Program 2

Implement SGPA Calculator

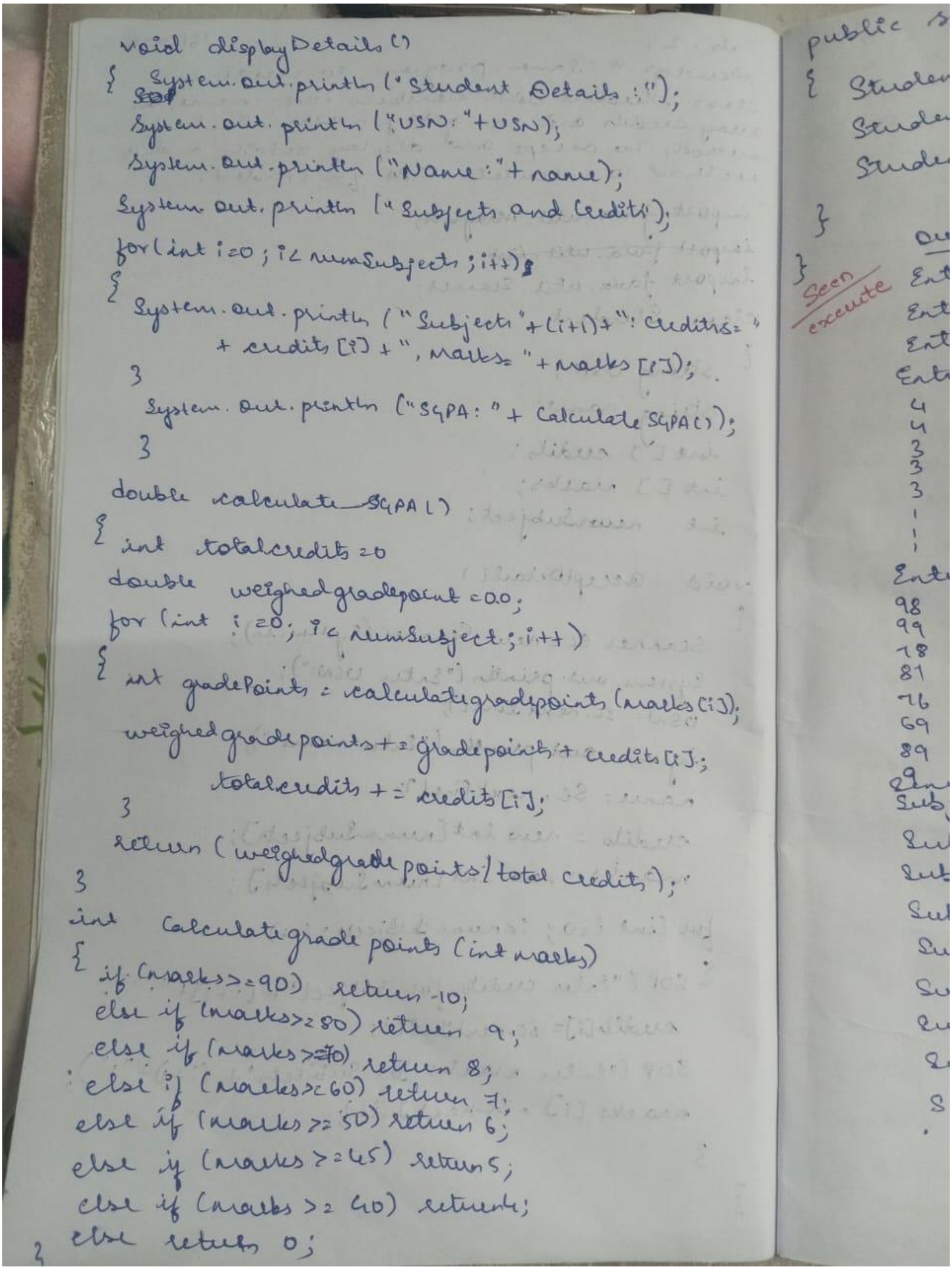


16/10/24.

lab - 2.

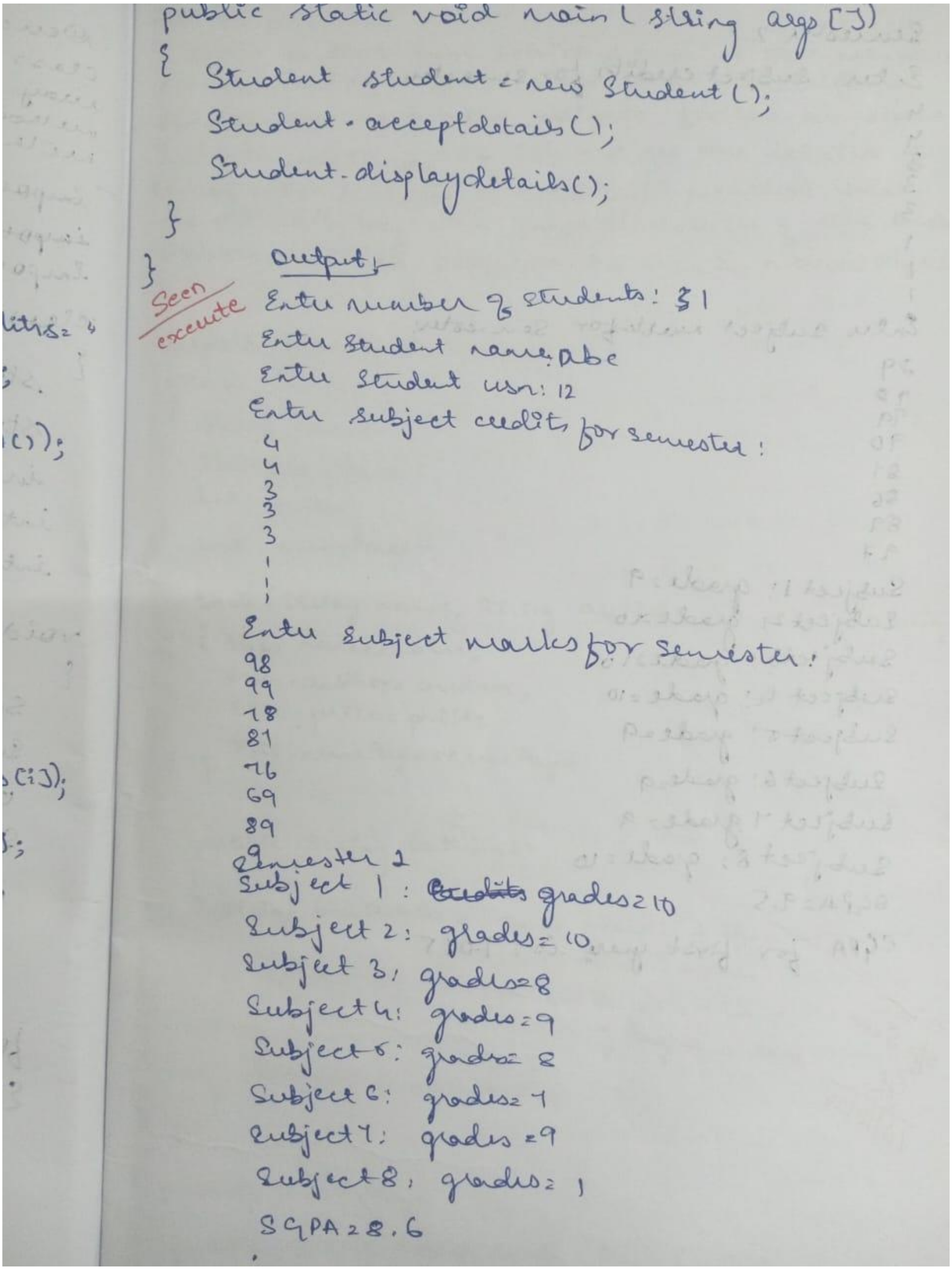
· Develop a Java program to create a Class Student with members USN, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
→ ~~import java.util.ArrayList;~~
~~import java.util.List;~~
Import java.util.Scanner;
class Student
{
  string USN;
  string name;
  int[] credits;
  int[] marks;
  int numSubject;

void acceptDetail()
{
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter USN:");
  USN = sc.nextLine();
  System.out.println("Enter name:");
  name = Sc.nextLine();
  credits = new int[numSubjects];
  marks = new int[numSubjects];
 for (int i=0; i<numSubjects; i++)
 { SOP ("Enter credits for Subject" + [i+1]+":");
   credit[i]= sc.nextInt();
   SOP ("Enter marks for Subjects"+ (i+1)+":");
   marks [i] = sc.nextInt();
 }
}
```



```
void displayDetails ()
{ System.out.println ("Student Details :");
 System.out.println ("USN: "+USN);
 System.out.println ("Name :"+ name);
System.out.println ("Subjects and Credits");
for(int i=0 ; i< numSubjects ; i++)
{
 System.out.println ("Subject "+(i+1)+": Crediths= "
          + credits [i] + ", Marks= "+marks [i]);
 }
 System.out.println ("SGPA: "+ Calculate SGPA());
 }

double calculate_SGPA ()
{ int totalcredits =0
double weighedgradepoint =0.0;
for (int i =0; i< numSubject ; i++)
{ int gradePoints = calculategradepoints (marks [i]);
 weighedgradepoints+= gradepoints + credits [i];
        totalcredits += credits [i];
 }
 return (weighedgradepoints/total credits);
}
int  Calculategradepoints (int marks)
{ if (marks>=90) return 10;
 else if (marks>=80) return 9;
 else if (marks>=70) return 8;
 else if (marks>=60) return 7;
else if (marks >= 50) return 6;
else if (marks >=45) return5;
else if (marks >= 40) return4;
} else return 0;
```

```
public s
{ Studen
  Studen
  Studen

 }
}       Ou
Seen
execute Ent
        Ent
        Ent
        Ent
         4
         4
         3
         3
         3
         1
         1
         1
        Ent
        98
        99
        78
        87
        76
        69
        89
        9
        Sub
        Su
        Sub
        Sub
        Su
        Su
        Su
        S
        S
```

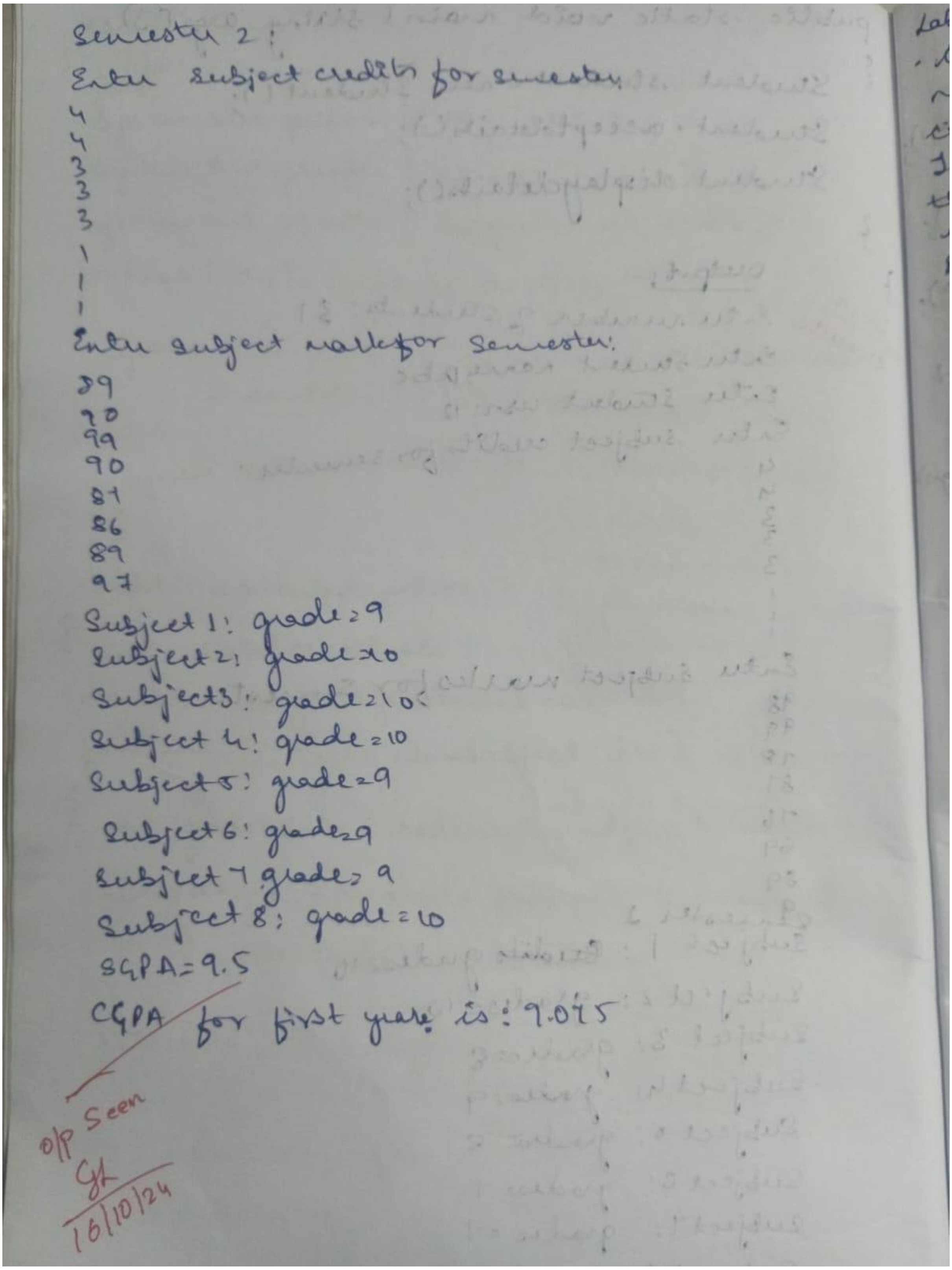


```
public static void main(String args[])
{
   Student student = new Student();
   Student.acceptdetails();
   Student.displaydetails();
 }
}
```

Output:-

Seen execute

```
Enter number of students: ~~3~~1
Enter Student name: abc
Enter Student usn: 12
Enter subject credits for semester:
 4
 4
 3
 3
 3
 1
 1
 1
Enter subject marks for semester:
98
99
78
81
76
69
89
Semester 1
Subject 1: ~~Credits~~ grades=10
Subject 2: grades=10
Subject 3: grades=8
Subject 4: grades=9
Subject 5: grades=8
Subject 6: grades=7
Subject 7: grades=9
Subject 8: grades=1
SGPA=8.6
```



```
Semester 2:
Enter subject credits for semester
4
4
3
3
3
1
1
1
Enter subject marksfor Semester:
89
90
99
90
87
86
89
97
Subject 1: grade=9
Subject2: grade=10
Subject3: grade=10
Subject 4: grade=10
Subject5: grade=9
Subject6: grade=9
Subject 7 grade= 9
Subject 8: grade=10
SGPA=9.5
CGPA for first year is: 9.075
```

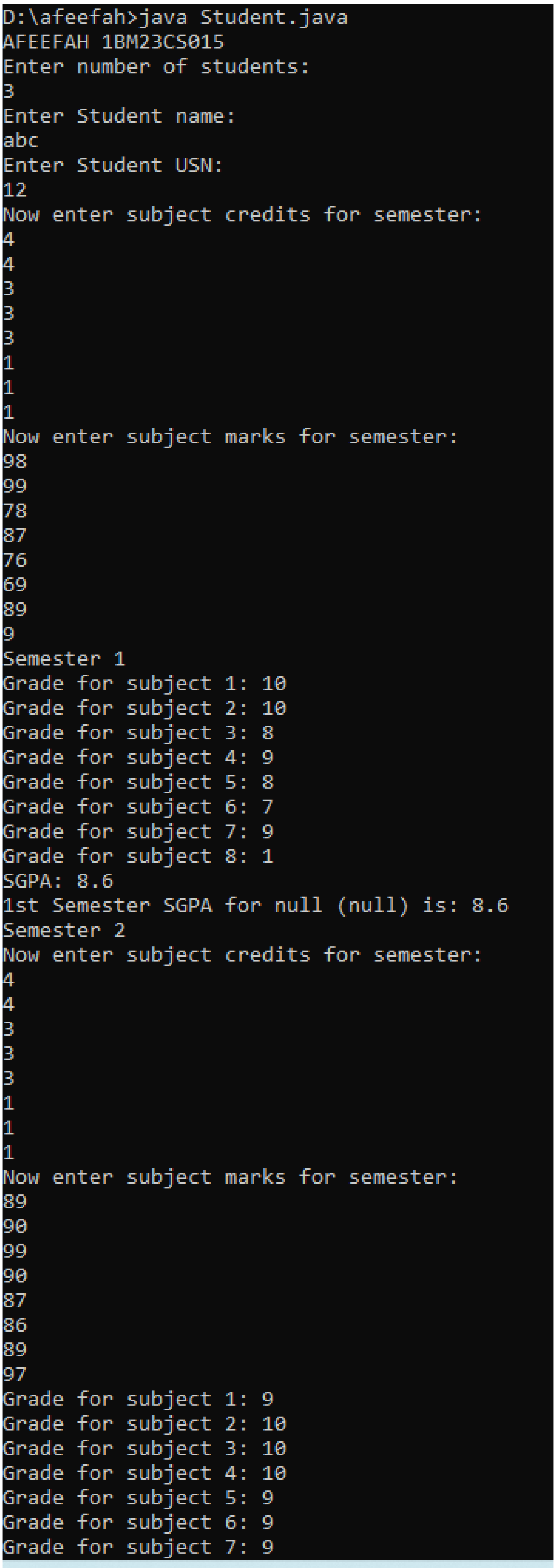
o/p Seen
16/10/24

Code:

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

```
class Student
{
    String name;
    String usn;
    int credits[] = new int[8];
    int marks[] = new int[8];
    double sgpa=0.0;
    double cgpa;
    int grade[] = new int[8];

    double calculate(int m[], int c[])
  {
        int j;
        double sum = 0.0;
        int div = 0;
        for (j = 0; j < 8; j++)
        {
            if (m[j] != 100)
            {
                grade[j] = (m[j] + 10) / 10;
            }
            else
            {
                grade[j] = 10;
            }
            div = credits[j] + div;
            sum = sum + (grade[j] * credits[j]);
            System.out.println("Grade for subject " + (j + 1) + ": " + grade[j]); // error check
        }
        sgpa = sum / div;
        System.out.println("SGPA: " + sgpa);
        return sgpa;
    }
double calcgpa(double sgpa1, double sgpa2)
{
cgpa=(sgpa1+sgpa2)/2;
return cgpa;
}
void input()
{
Scanner sc=new Scanner(System.in);
        System.out.println("Now enter subject credits for semester:");
        int i;
        for (i = 0; i < 8; i++)
        {
            credits[i] = sc.nextInt();
        }
        System.out.println("Now enter subject marks for semester:");
        for (i = 0; i < 8; i++)
        {
            marks[i] = sc.nextInt();
        }
}
```

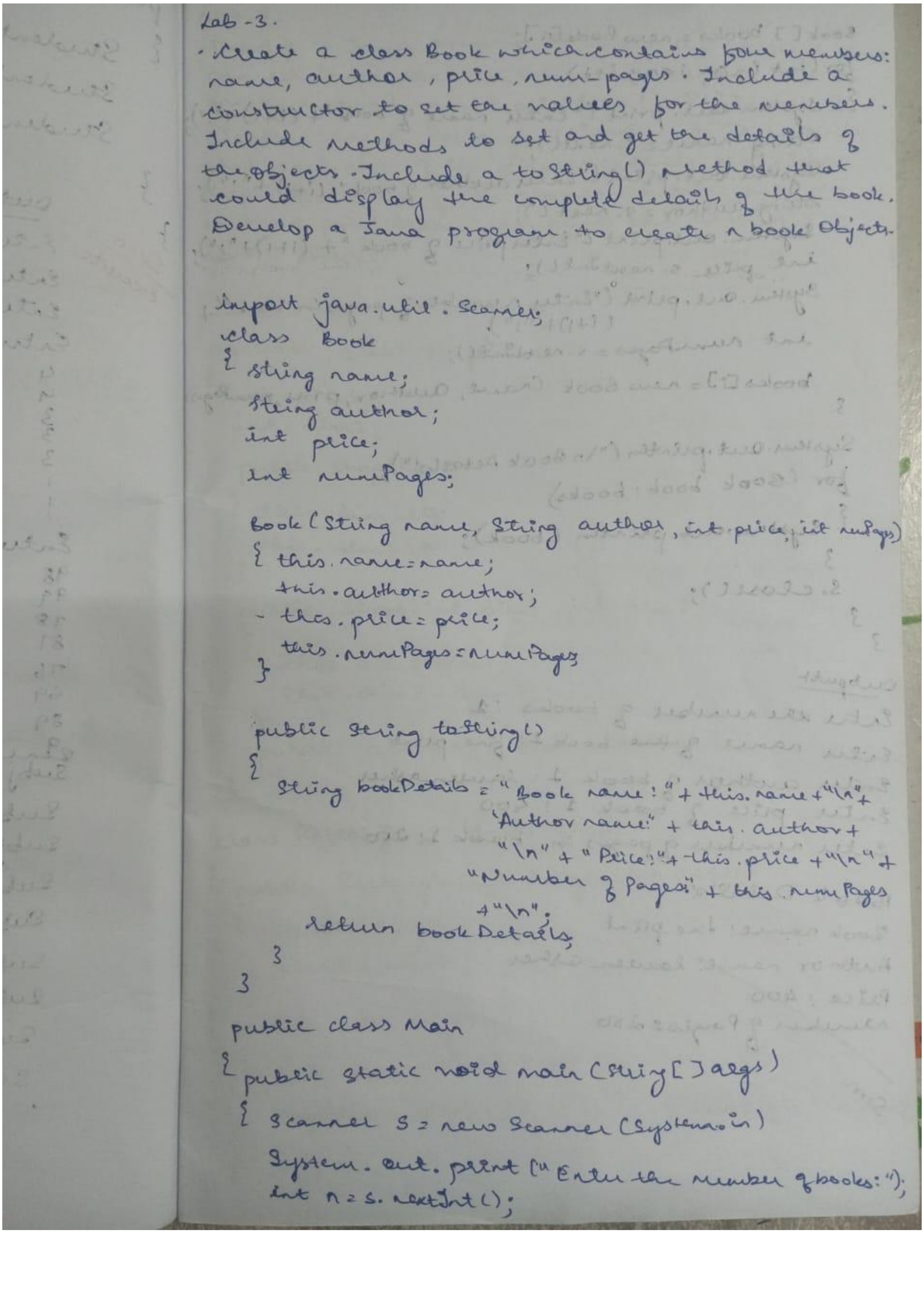
```
    public static void main(String args[]) {
System.out.println("AFEEFAH 1BM23CS015");
Scanner sc1 = new Scanner(System.in);
System.out.println("Enter number of students: ");
int n=sc1.nextInt();
Student obj[]=new Student[n];
int k;
for(k=0;k<n;k++)
{
obj[k]=new Student();
System.out.println("Enter Student name: ");
        String name = sc1.next();
System.out.println("Enter Student USN: ");
        String usn = sc1.next();
obj[k].input();
System.out.println("Semester 1");
        double result = obj[k].calculate(obj[k].marks, obj[k].credits);
        System.out.println("1st Semester SGPA for " + obj[k].name + " (" +
obj[k].usn + ") is: " + result);
System.out.println("Semester 2");
obj[k].input();
        double result2 = obj[k].calculate(obj[k].marks, obj[k].credits);
        System.out.println("2nt Semester SGPA for " + obj[k].name + " (" +
obj[k].usn + ") is: " + result2);
System.out.println("CGPA for 1st year is : "+obj[k].calcgpa(result,result2));
}
    }
}
```



```
D:\afeefah>java Student.java
AFEEFAH 1BM23CS015
Enter number of students:
3
Enter Student name:
abc
Enter Student USN:
12
Now enter subject credits for semester:
4
4
3
3
3
1
1
1
Now enter subject marks for semester:
98
99
78
87
76
69
89
9
Semester 1
Grade for subject 1: 10
Grade for subject 2: 10
Grade for subject 3: 8
Grade for subject 4: 9
Grade for subject 5: 8
Grade for subject 6: 7
Grade for subject 7: 9
Grade for subject 8: 1
SGPA: 8.6
1st Semester SGPA for null (null) is: 8.6
Semester 2
Now enter subject credits for semester:
4
4
3
3
3
1
1
1
Now enter subject marks for semester:
89
90
99
90
87
86
89
97
Grade for subject 1: 9
Grade for subject 2: 10
Grade for subject 3: 10
Grade for subject 4: 10
Grade for subject 5: 9
Grade for subject 6: 9
Grade for subject 7: 9
```

Program 3:

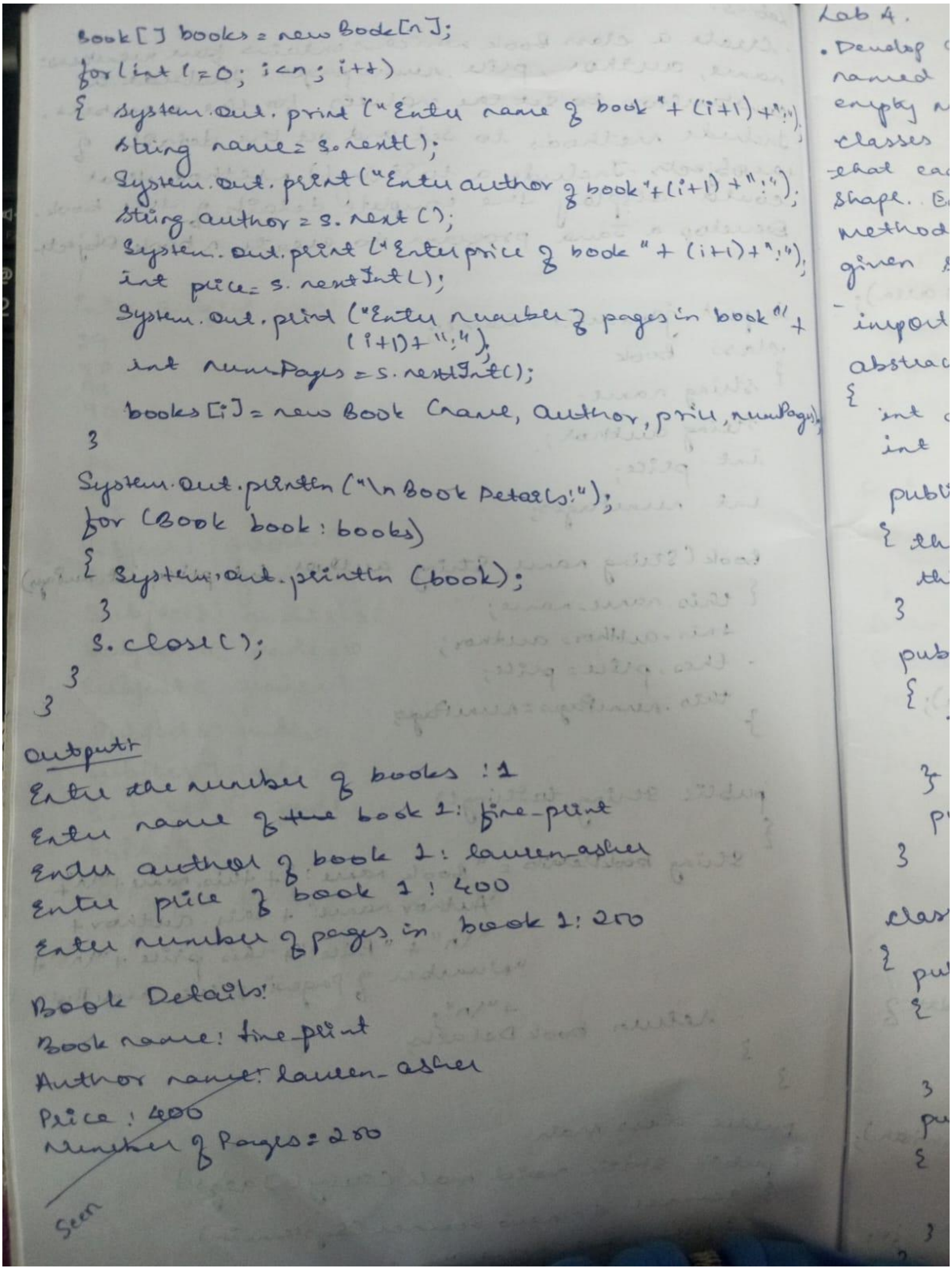
Create Objects for Books



Lab -3.

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

```
import java.util.Scanner;
class Book
{ string name;
  String author;
  int  price;
  int  numPages;

  Book(String name, String author, int price, int numPages)
  { this.name=name;
    this.author= author;
  - this.price= price;
    this.numPages=numPages;
  }

  public String toString()
  {
    String bookDetails = "Book name: "+ this.name +"\n"+
                         "Author name:" + this.author +
                          "\n" + "Price:"+ this.price +"\n" +
                         "Number of Pages:" + this.numPages
                          +"\n".
      return bookDetails;
    }
  }
public class Main
{ public static void main (String[] args)
  { Scanner S = new Scanner (System.in)
    System.out.print ("Enter the number of books:");
    int n = s.nextInt();
```



```
Book[] books = new Book[n];
for(int i=0; i<n; i++)
{ System.out.print("Enter name of book "+ (i+1)+":");
  String name= s.next();
  System.out.print("Enter author of book "+(i+1) +":");
  String author = s.next();
  System.out.print("Enter price of book "+ (i+1)+":");
  int price= s.nextInt();
  System.out.print("Enter number of pages in book "+
                   (i+1)+":");
  int numPages = s.nextInt();
  books[i]= new Book (name, author, price, numPages);
}
System.out.println("\nBook Details:");
for (Book book: books)
{ System.out.println (book);
 }
 s.close();
}
}
```

Output:

```
Enter the number of books :1
Enter name of ~~the~~ book 1: fine-print
Enter author of book 1: lauren_asher
Enter price of book 1 ! 400
Enter number of pages in book 1: 250

Book Details!
Book name! fine-print
Author name: lauren_asher
Price : 400
Number of Pages: 250
```

Seen

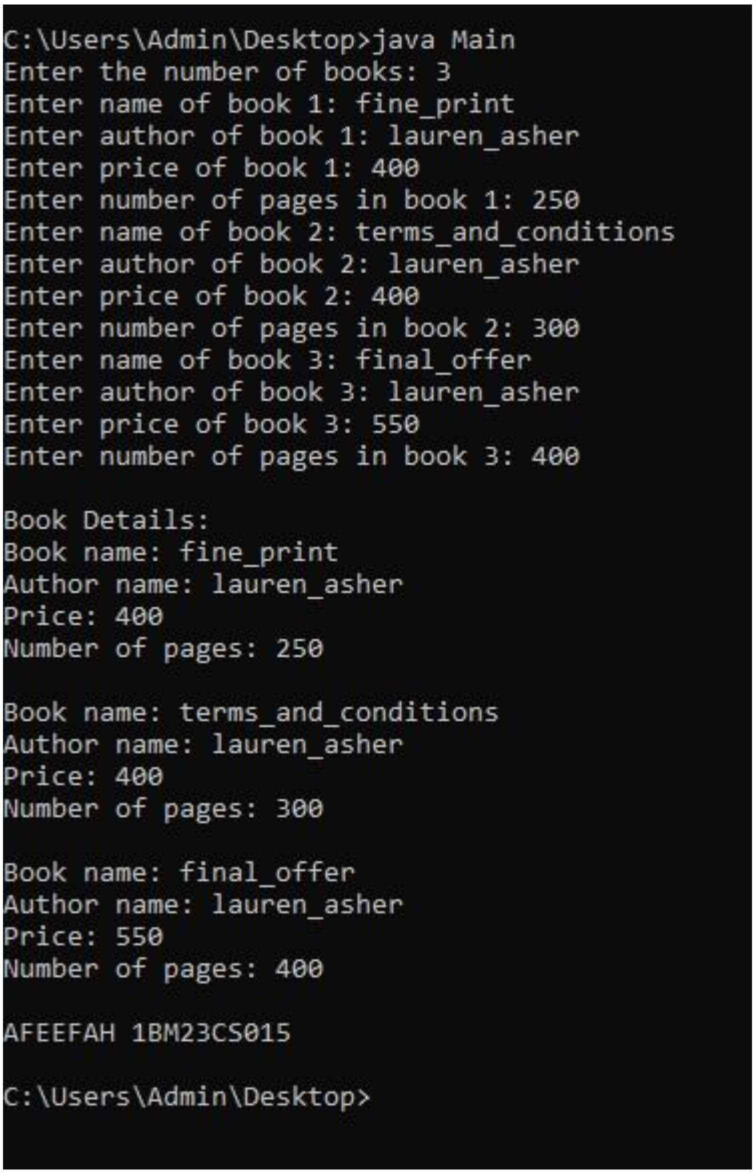
Lab 4.
• Develop
named
employ
classes
that ea
Shape. E
method
given
- import
abstrac
{
int
int
publ
{ th
th
}
pub
{
}
p
}
clas
{ pu
{
}
pu
{
}

Code:

import java.util.Scanner;

```
class Book
{

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

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


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

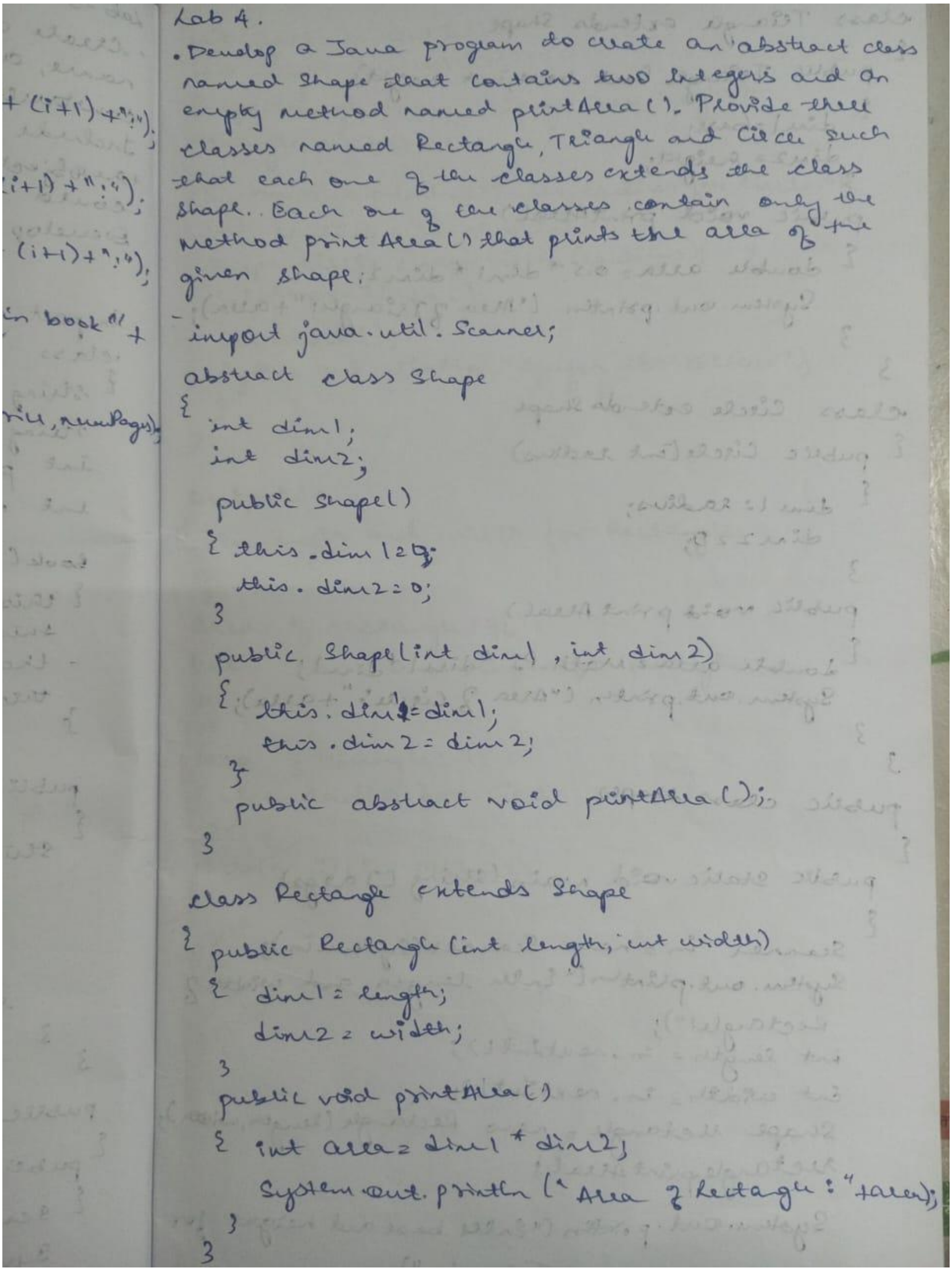
```
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            // Prompt user to enter book details
            System.out.print("Enter name of book " + (i + 1) + ": ");
            String name = s.next();
            System.out.print("Enter author of book " + (i + 1) + ": ");
            String author = s.next();
            System.out.print("Enter price of book " + (i + 1) + ": ");
            int price = s.nextInt();
            System.out.print("Enter number of pages in book " + (i + 1) + ": ");
            int numPages = s.nextInt();

            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (Book book : books) {
            System.out.println(book);
        }


        s.close();
            System.out.println("AFEEFAH 1BM23CS015");
    }
}
```



```
C:\Users\Admin\Desktop>java Main
Enter the number of books: 3
Enter name of book 1: fine_print
Enter author of book 1: lauren_asher
Enter price of book 1: 400
Enter number of pages in book 1: 250
Enter name of book 2: terms_and_conditions
Enter author of book 2: lauren_asher
Enter price of book 2: 400
Enter number of pages in book 2: 300
Enter name of book 3: final_offer
Enter author of book 3: lauren_asher
Enter price of book 3: 550
Enter number of pages in book 3: 400

Book Details:
Book name: fine_print
Author name: lauren_asher
Price: 400
Number of pages: 250

Book name: terms_and_conditions
Author name: lauren_asher
Price: 400
Number of pages: 300

Book name: final_offer
Author name: lauren_asher
Price: 550
Number of pages: 400

AFEEFAH 1BM23CS015

C:\Users\Admin\Desktop>
```

Program 4:

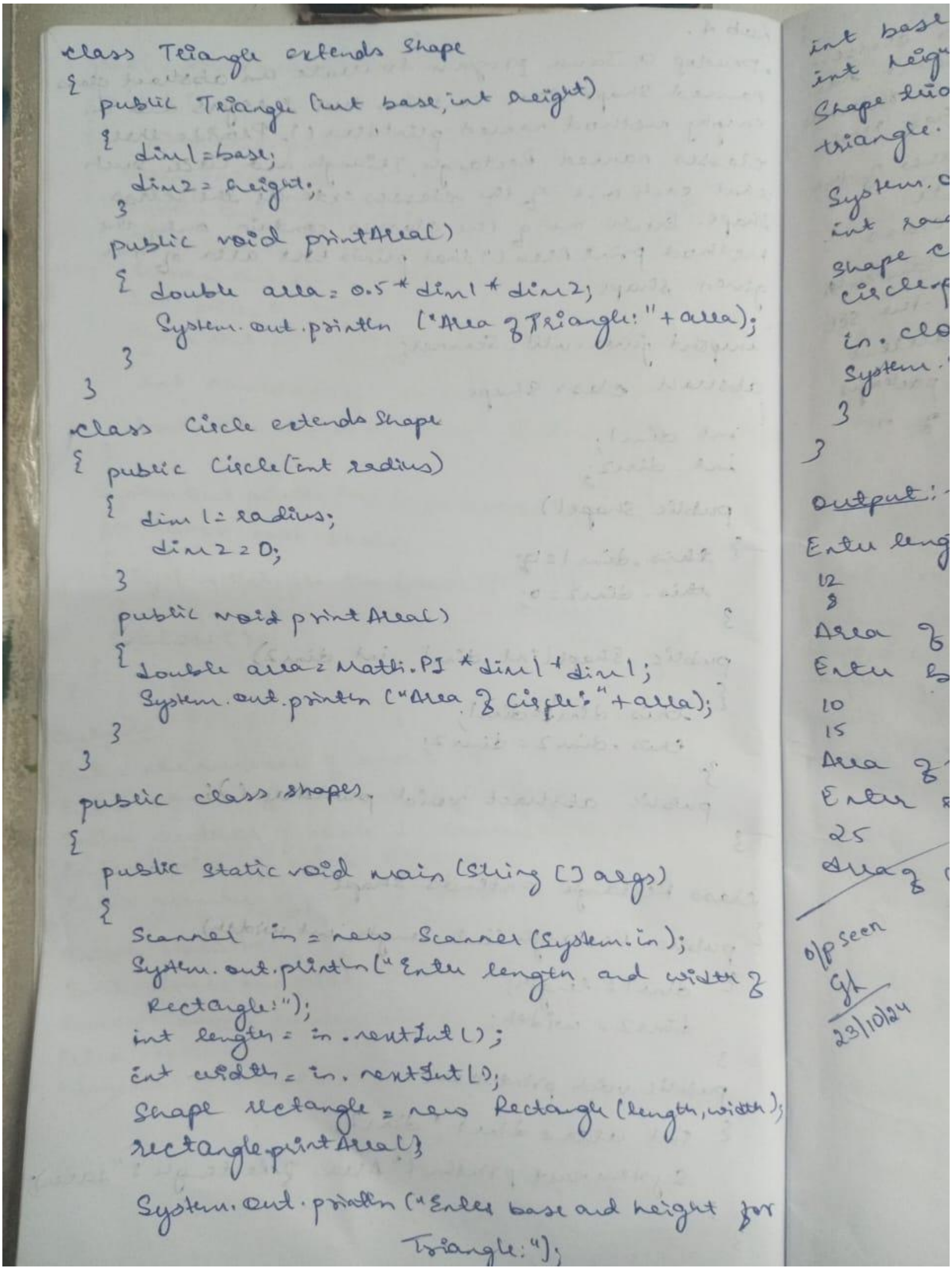
Implement Abstract Class



Lab 4.

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

```
import java.util.Scanner;
abstract class Shape
{
  int dim1;
  int dim2;
  public Shape()
  { this.dim1=0;
    this.dim2=0;
  }
  public Shape(int dim1 , int dim2)
  {
     this.dim1=dim1;
     this.dim2=dim2;
  }
   public abstract void printArea();
}
class Rectangle extends Shape
{
  public Rectangle(int length, int width)
  {  dim1 = length;
     dim2 = width;
  }
  public void printArea()
  {  int area= dim1 * dim2;
     System.out.println ("Area of Rectangle : "+area);
  }
}
```



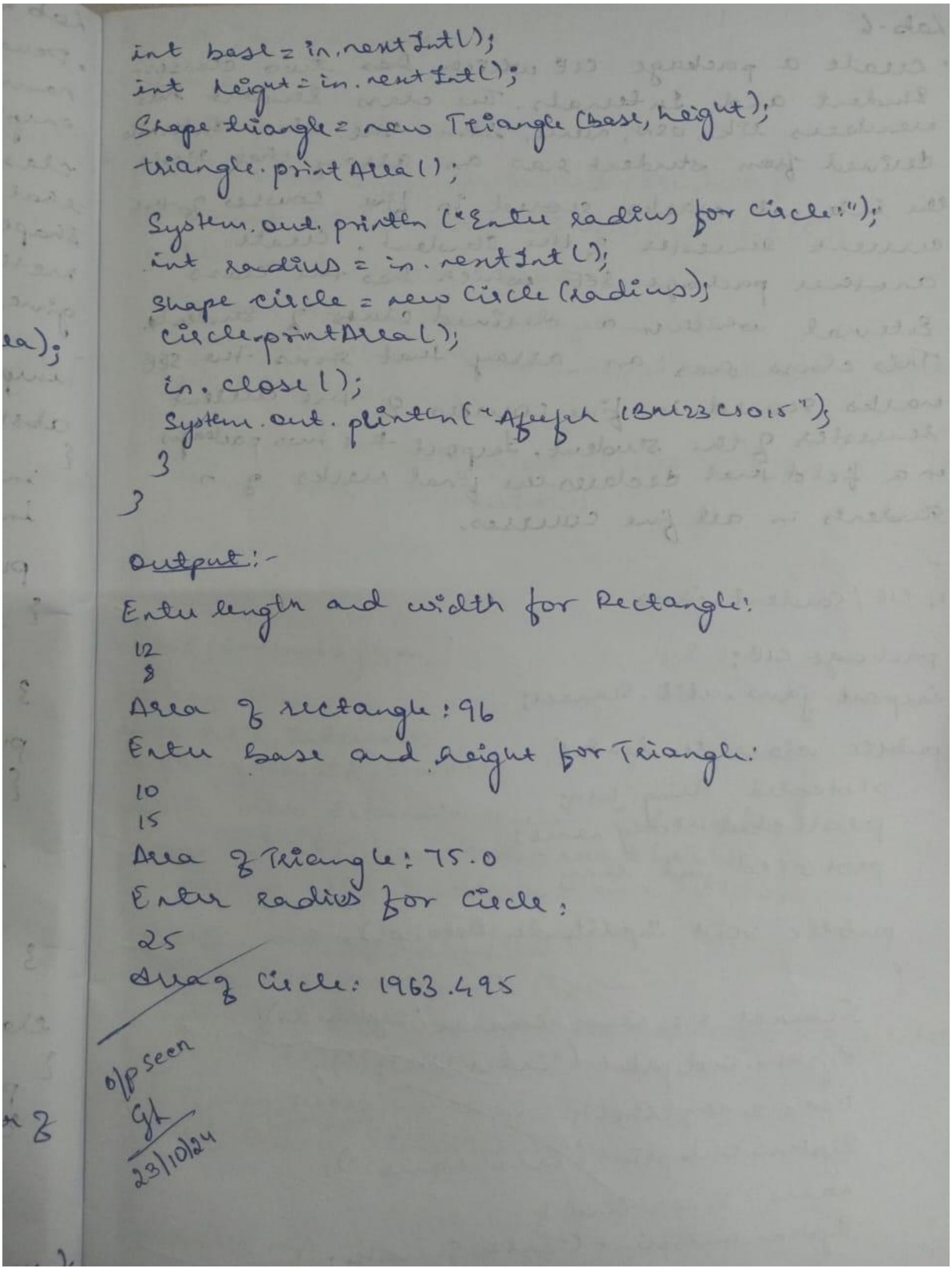
```
class Triangle extends Shape
{
  public Triangle (int base,int height)
  {
    dim1=base;
    dim2= height;
  }
  public void printArea()
  {
    double area= 0.5*dim1*dim2;
    System.out.println ("Area of Triangle:"+area);
  }
}
class Circle extends Shape
{ public Circle(int radius)
  {
    dim1= radius;
    dim2=0;
  }
  public void printArea()
  {
    double area= Math.PI *dim1*dim1;
    System.out.println ("Area of Circle:"+area);
  }
}
public class shapes
{
  public static void main (String [] args)
  {
    Scanner in = new Scanner(System.in);
    System.out.println("Enter length and width of
     Rectangle:");
    int length = in.nextInt();
    int width = in.nextInt();
    Shape rectangle = new Rectangle(length,width);
    rectangle.printArea()

    System.out.println ("Enter base and height for
                                  Triangle:");
```

```
int base
int heig
Shape tria
triangle.

System.o
int rad
Shape c
circle.p
in.clo
System.
  }
}
```

Output:

```
Enter leng
12
8
Area of
Enter b
10
15
Area of
Enter r
25
Area of
```

o/p seen
GL
23/10/24



```
int base = in.nextInt();
int height = in.nextInt();
Shape triangle = new Triangle(base, height);
triangle.printArea();
 System.out.println("Enter radius for circle:");
 int radius = in.nextInt();
 Shape circle = new Circle(radius);
 circle.printArea();
 in.close();
 System.out.println("Afeefah 1BM23CS015");
 }
}
```

Output:-

```
Enter length and width for Rectangle:
 12
 8
Area of rectangle: 96
Enter base and height for Triangle:
10
15
Area of Triangle: 75.0
Enter radius for Circle:
25
Area of Circle: 1963.495
```

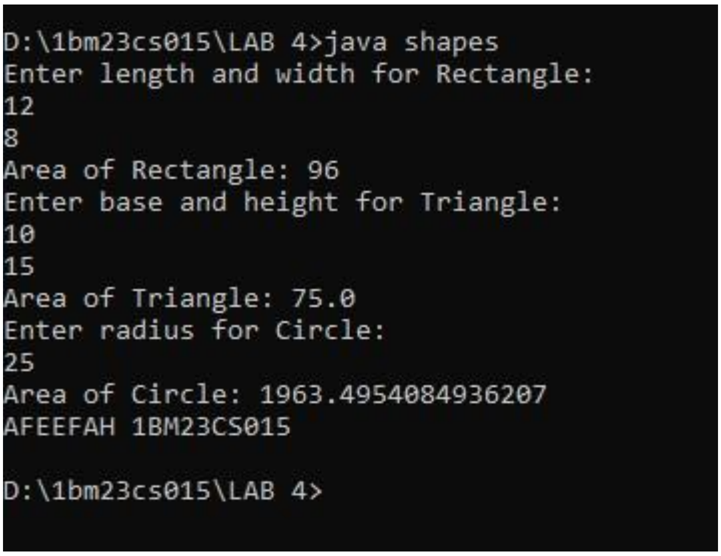
o/p seen

Gh
23/10/24

Code:

import java.util.Scanner;

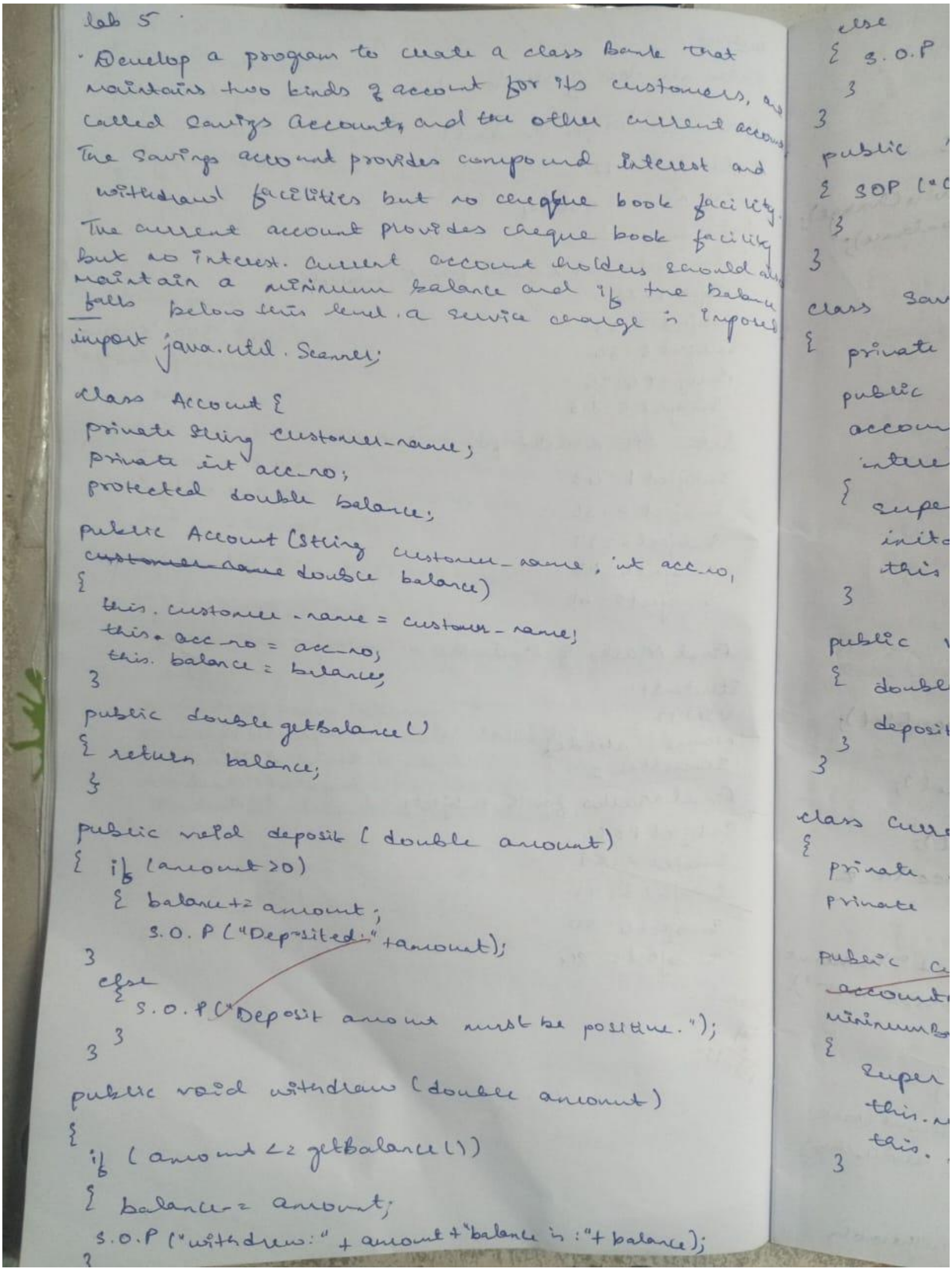
```
}


abstract class Shape {
    int dim1;
    int dim2;



    public Shape() {
        this.dim1 = 0;
        this.dim2 = 0;

    }

    public Shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    public abstract void printArea();

}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dim1 = length;
        dim2 = width;
    }

    public void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);

    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dim1 = base;
        dim2 = height;
    }

    public void printArea() {

        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);

    }
}

class Circle extends Shape {
    public Circle(int radius) {

        dim1 = radius;
        dim2 = 0;
    }
    public void printArea() {

        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class shapes
{

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

        System.out.println("Enter length and width for Rectangle:");

        int length = in.nextInt();
        int width = in.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");

        int base = in.nextInt();
        int height = in.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");

        int radius = in.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();

        in.close();
            System.out.println("AFEEFAH 1BM23CS015");
    }
}
```



```
D:\1bm23cs015\LAB 4>java shapes
Enter length and width for Rectangle:
12
8
Area of Rectangle: 96
Enter base and height for Triangle:
10
15
Area of Triangle: 75.0
Enter radius for Circle:
25
Area of Circle: 1963.4954084936207
AFEEFAH 1BM23CS015

D:\1bm23cs015\LAB 4>
```

Program 5:

Bank Account Management

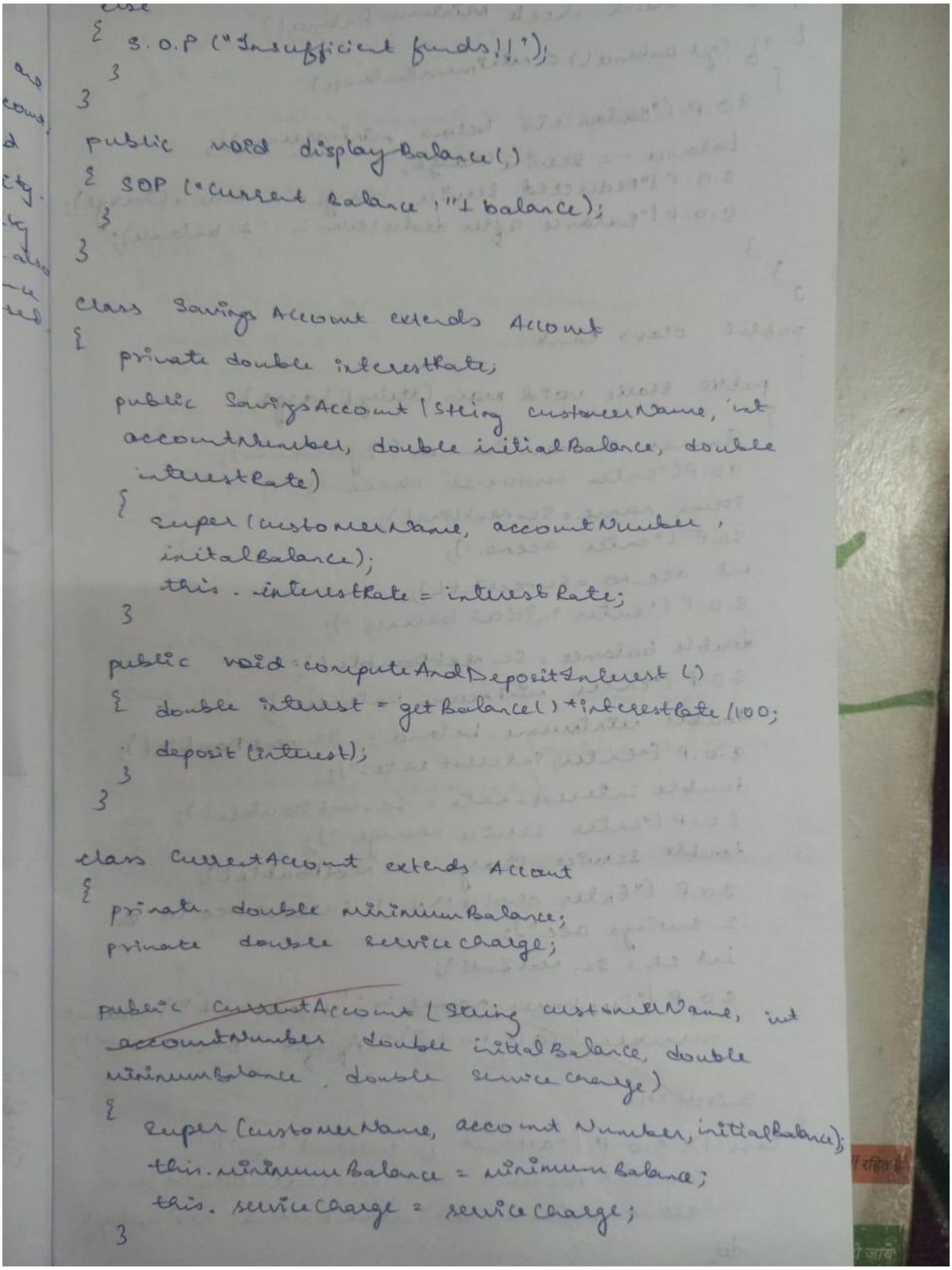


lab 5 ·

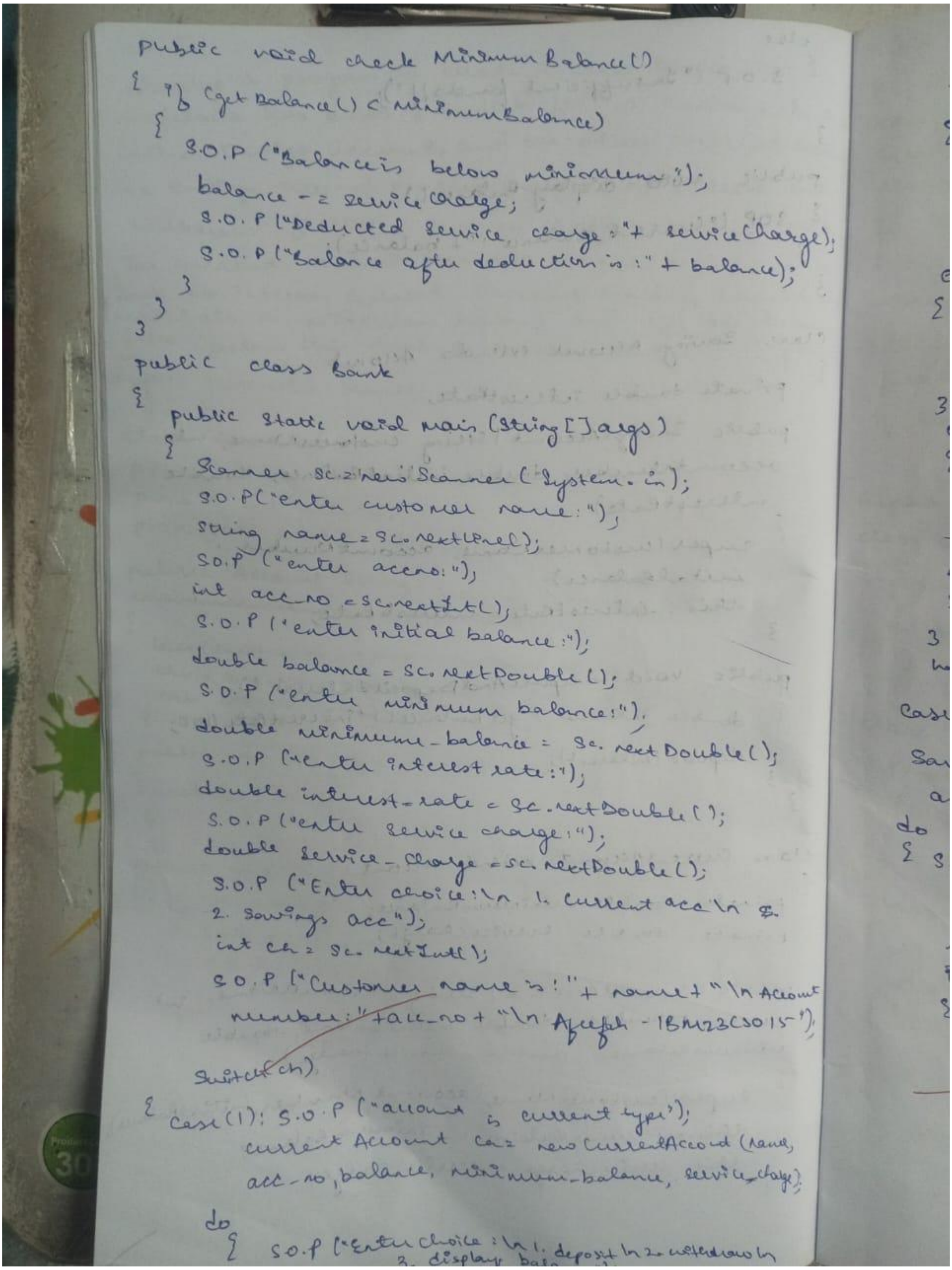
· Develop a program to create a class Bank that maintains two kinds of account for its customers, one called Savings Accounts and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility 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.

```
import java.util.Scanner;

class Account {
 private String customer-name;
 private int acc_no;
 protected double balance;
public Account (String customer_name, int acc_no,
~~customer_name~~ double balance)
{
  this.customer_name = customer_name;
  this. acc_no = acc_no;
  this. balance = balance;
 }
public double getBalance()
{ return balance;
 }

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

public void withdraw (double amount)
{
 if (amount <= getBalance())
 { balance-= amount;
 S.O.P ("withdrew:" + amount +"balance is :"+ balance);
 }
```

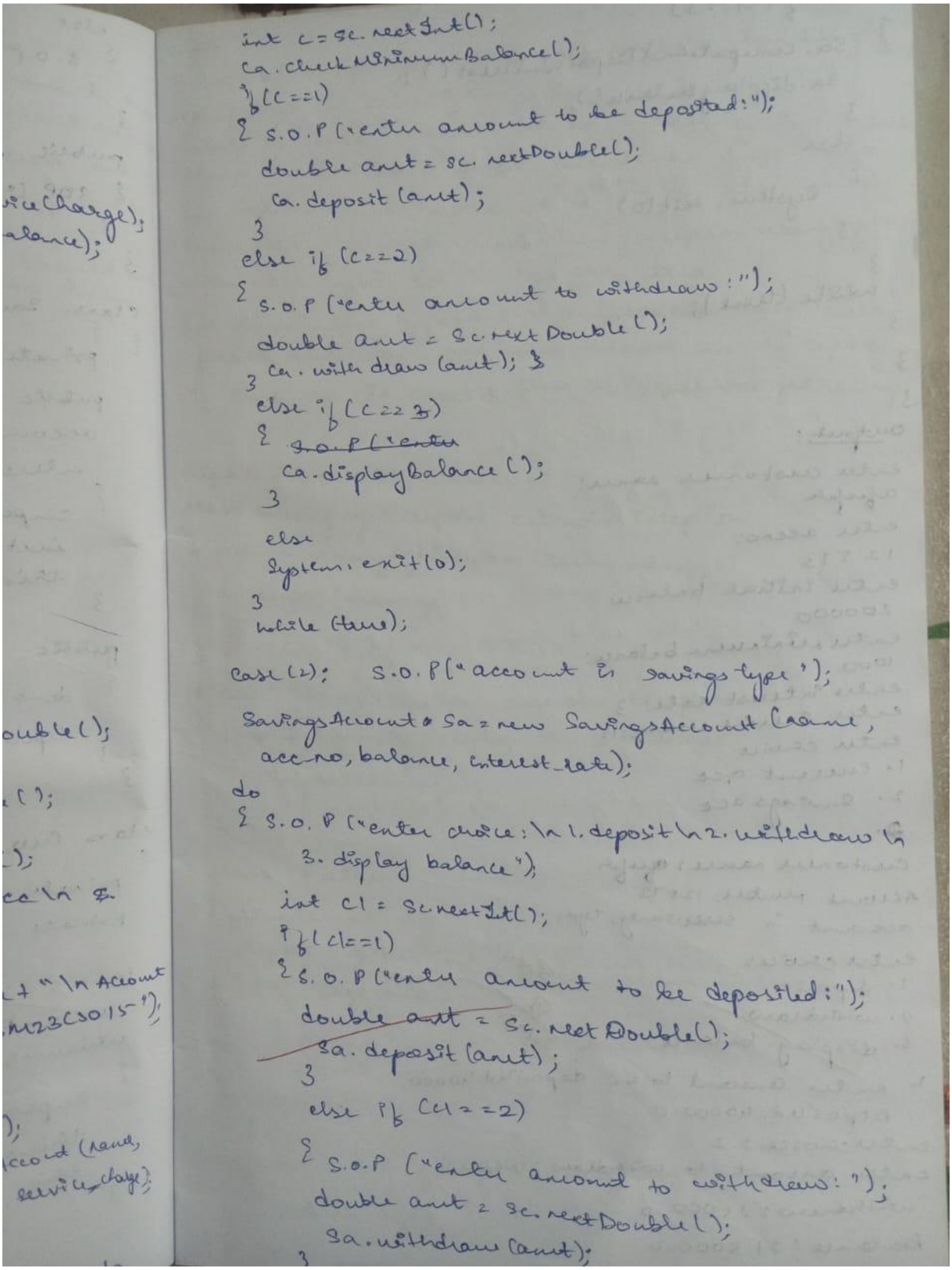
```
else
{ S.O.P
 }
}
public
{ SOP ("C
 }
}
class Sav
{ private
  public
  accoun
  intere
  { supe
   init
   this
  }
 public
 { doubl
   deposi
 }
}
class Curre
{
 private
 Private
 public C
 ~~account~~
 minimumBa
 {
  super
  this.
  this.
 }
```



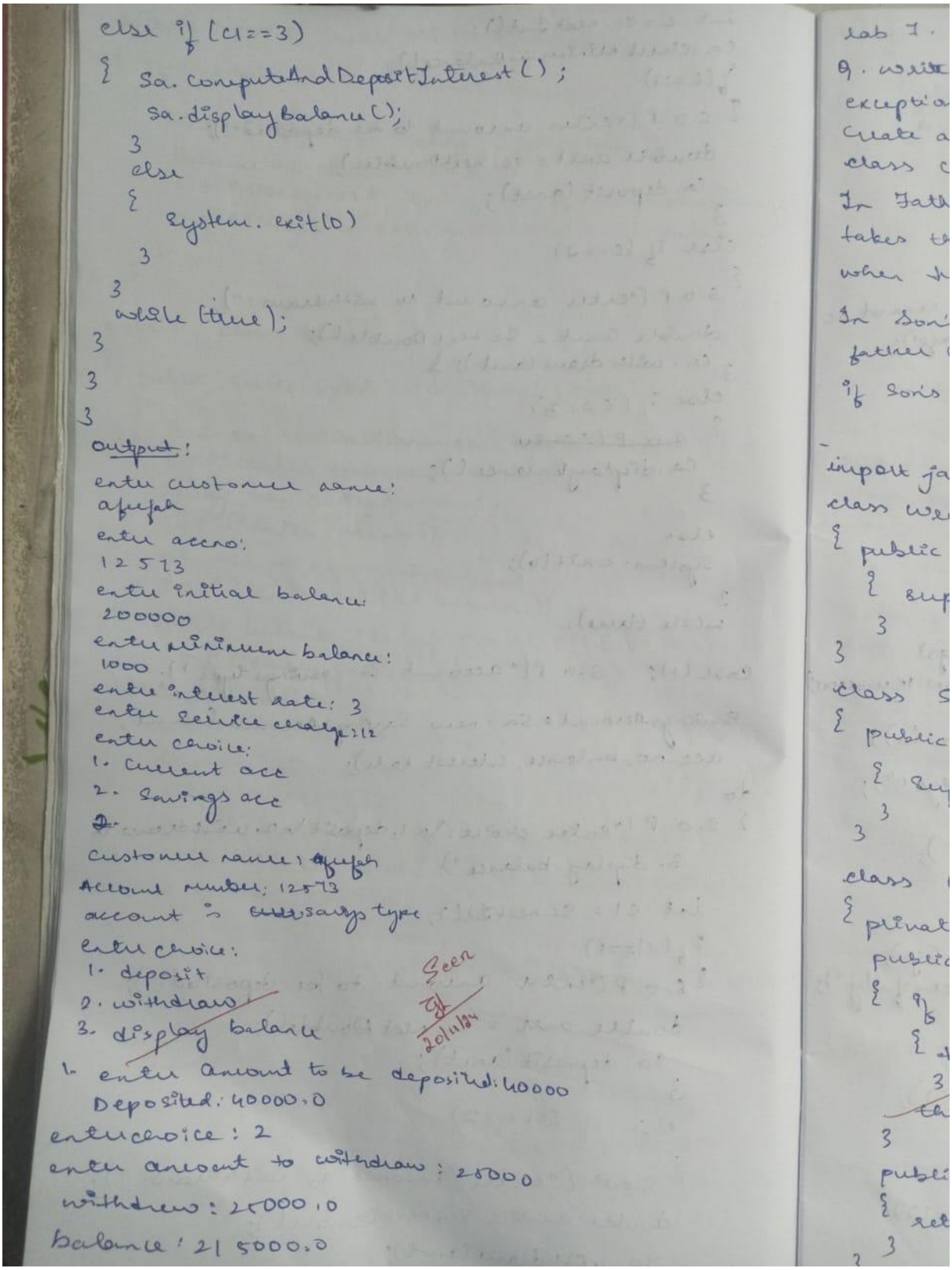
```
else
 { S.O.P ("Insufficient funds!!");
 }
}
public void displayBalance()
 { SOP ("Current Balance :"+ balance);
 }
}

class SavingsAccount extends Account
{
   private double interestRate;
   public SavingsAccount (String customerName, int
   accountNumber, double initialBalance, double
   interestRate)
   {
     super (customerName, accountNumber,
     initalBalance);
     this . interestRate = interestRate;
   }
   public void computeAndDepositInterest ()
   { double interest = getBalance() *interestRate /100;
     deposit (interest);
   }
 }

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

  public CurrentAccount (String customerName, int
  ~~accountNumber~~ double initialBalance, double
  minimumBalance , double serviceCharge)
  {
     super (customerName, accountNumber, initialBalance);
     this.minimumBalance = minimumBalance;
     this. serviceCharge = serviceCharge;
  }
```



```
public void check MinimumBalance()
{ if (getBalance() < minimumBalance)
  {
    S.O.P ("Balance is below minimum");
    balance -= serviceCharge;
    S.O.P ("Deducted service charge:"+ serviceCharge);
    S.O.P ("Balance after deduction is :" + balance);
  }
 }
}
public class Bank
{
  public static void main (String[] args)
  {
    Scanner sc=new Scanner (System.in);
    S.O.P("enter customer name:");
    String name=sc.nextLine();
    S.O.P ("enter accno:");
    int acc_no =sc.nextInt();
    S.O.P ("enter initial balance:");
    double balance = sc.nextDouble();
    S.O.P ("Enter minimum balance:");
    double minimum_balance = sc.nextDouble();
    S.O.P ("enter interest rate:");
    double interest_rate = sc.nextDouble();
    S.O.P ("enter service charge:");
    double service_charge =sc.nextDouble();
    S.O.P ("Enter choice:\n 1. current acc\n 2.
    2. Savings acc");
    int ch = sc.nextInt();
    S.O.P ("Customer name is:"+ name+ "\n Account
     number:"+acc_no + "\n Afeefah - 1BM23CS015");

  Switch(ch)
 { case(1): S.O.P ("account is current type");
      current Account ca= new CurrentAccount (name,
      acc_no, balance, minimum_balance, service_charge);

  do
   { S.O.P ("Enter choice :\n 1. deposit \n 2. withdraw \n
              3. display bal
```



```
int c = sc.nextInt();
ca.checkMinimumBalance();
if(c==1)
{ s.o.p("enter amount to be deposited:");
 double amt = sc.nextDouble();
  ca.deposit (amt);
 }
else if (c==2)
{ s.o.p ("enter amount to withdraw:");
 double amt = sc.nextDouble();
} ca.withdraw (amt); }
 else if (c==3)
 { ~~s.o.p("enter~~
  ca.displayBalance();
  }
  else
  System.exit(0);
 }
 while (true);
case(2):   s.o.p("account is savings type");
SavingsAccount sa = new SavingsAccount (name,
  acc_no, balance, interest_rate);
do
{ s.o.p ("enter choice:\n 1.deposit\n 2.withdraw \n
     3. display balance");
   int c1 = sc.nextInt();
   if(c1==1)
   {s.o.p("enter amount to be deposited:");
   double amt = sc.nextDouble();
   sa.deposit (amt);
   }
   else if (c1==2)
   { s.o.p ("enter amount to withdraw:");
   double amt = sc.nextDouble();
    sa.withdraw (amt);
   }
```



```
else if (c1==3)
{  Sa.computeAndDepositInterest ( ) ;
    Sa.displayBalance ( );
  }
  else
  {
     System. exit(0)
   }
 }
 while (true);
}
}
}
```

```
output:
enter customer name:
afeefah
enter accno:
12573
enter initial balance:
200000
enter minimum balance:
1000
enter interest rate: 3
enter service charge:12
enter choice:
1. Current acc
2. Savings acc
~~2.~~
Customer name: afeefah
Account number: 12573
account is ~~curr~~savings type
enter choice:
1. deposit
2. withdraw
3. display balance
1. enter amount to be deposited: 40000
   Deposited: 40000.0
enter choice : 2
enter amount to withdraw : 25000
 withdrew : 25000.0
balance : 215000.0
```

Seen
20/11/24

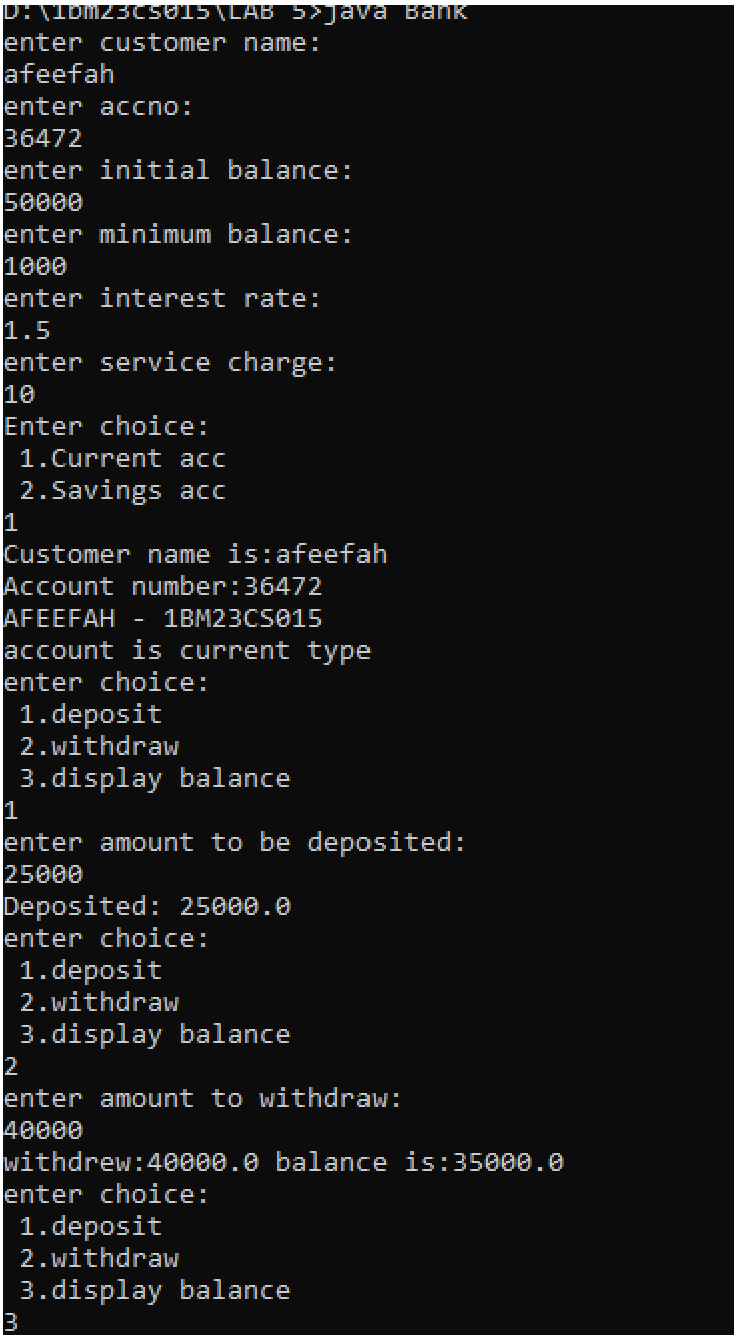
Code:

import java.util.Scanner;

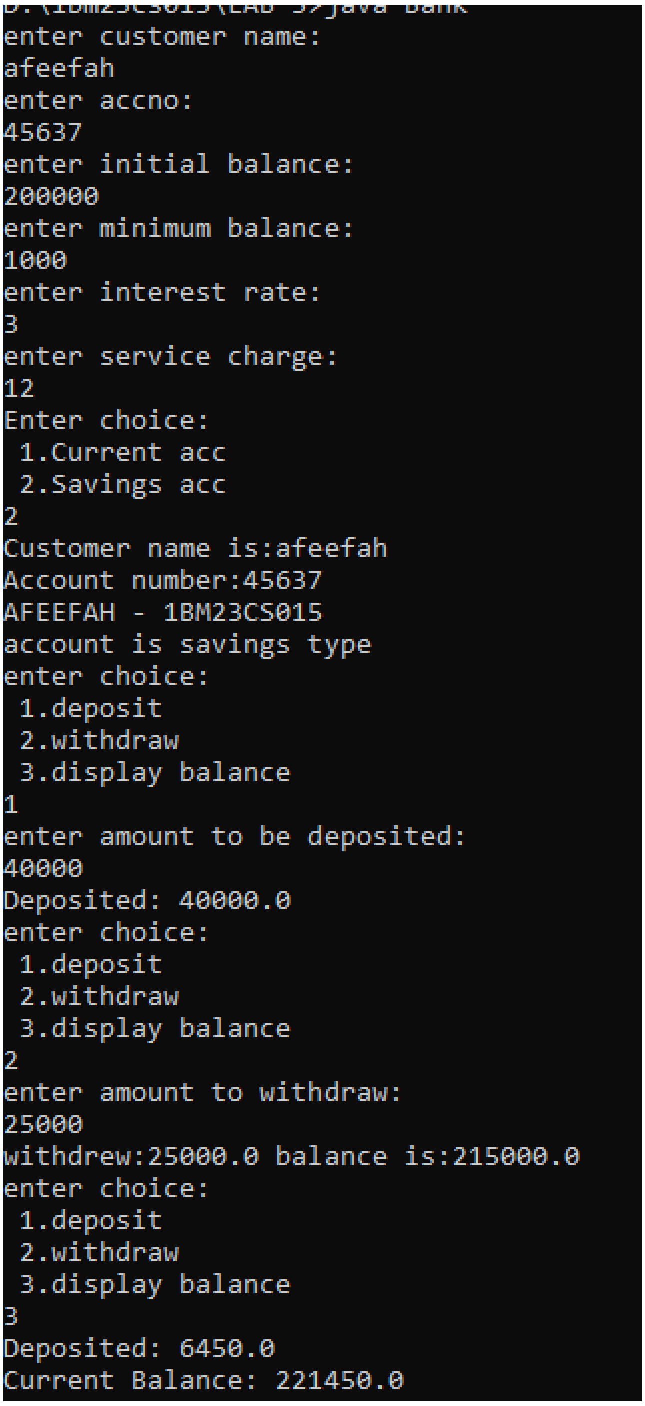
```
class Account {
    private String customer_name;
    private int acc_no;
    protected double balance;

    public Account(String customer_name, int acc_no, double balance) {
        this.customer_name = customer_name;
        this.acc_no = acc_no;
        this.balance = balance;
    }

    public double getBalance() {
        return balance;
    }

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

```
          System.out.println("withdrew:"+amount + " balance is:"+ balance);
          }
        else
         System.out.println("Insufficient funds!!");
     }
    public void displayBalance(){
        System.out.println("Current Balance: " + balance);
    }
}

class SavingsAccount extends Account {
    private double interestRate;

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

    public void computeAndDepositInterest() {
        double interest = getBalance() * interestRate / 100;
        deposit(interest);
    }
}
class CurrentAccount extends Account {
    private double minimumBalance;
    private double serviceCharge;
    public CurrentAccount(String customerName, int accountNumber, double
initialBalance, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
    public void checkMinimumBalance() {
        if (getBalance() < minimumBalance) {
            System.out.println("Balance is below minimum");
            balance-=serviceCharge;
            System.out.println("Deducted service charge:" +serviceCharge);
            System.out.println("Balance after deduction is:"+balance);
        }
    }
}
public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter customer name:");
        String name=sc.nextLine();
        System.out.println("enter accno:");
        int acc_no=sc.nextInt();
        System.out.println("enter initial balance:");
        double balance=sc.nextDouble();
        System.out.println("enter minimum balance:");
        double minimum_balance=sc.nextDouble();
        System.out.println("enter interest rate:");
        double interest_rate=sc.nextDouble();
        System.out.println("enter service charge:");
        double service_charge=sc.nextDouble();
        System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
        int ch=sc.nextInt();
        System.out.println("Customer name is:"+ name+"\nAccount
number:"+acc_no+"\nAFEEFAH - 1BM23CS015");


        switch(ch){
            case(1):
                System.out.println("account is current type");
                CurrentAccount ca = new
CurrentAccount(name,acc_no,balance,minimum_balance,service_charge);
               do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n
3.display balance");
                int c=sc.nextInt();
                ca.checkMinimumBalance();
                if(c==1){
                   System.out.println("enter amount to be deposited:");
                   double amt=sc.nextDouble();
                     ca.deposit(amt);}
                else if(c==2){
                   System.out.println("enter amount to withdraw:");
                   double amt=sc.nextDouble();
                   ca.withdraw(amt);}
                else if(c==3){
                   ca.displayBalance();}
                else
                  System.exit(0);
```

```
         }while(true);


     case(2):
         System.out.println("account is savings type");
        SavingsAccount sa=new SavingsAccount(name,acc_no,balance,interest_rate);
        do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
        int c1=sc.nextInt();
        if(c1==1){
          System.out.println("enter amount to be deposited:");
          double amt=sc.nextDouble();
           sa.deposit(amt);}
        else if(c1==2){
          System.out.println("enter amount to withdraw:");
          double amt=sc.nextDouble();
          sa.withdraw(amt);}
        else if(c1==3){
         sa.computeAndDepositInterest();
          sa.displayBalance();}
        else{
         System.exit(0);
            }
        }while(true);
  }
}
}
```



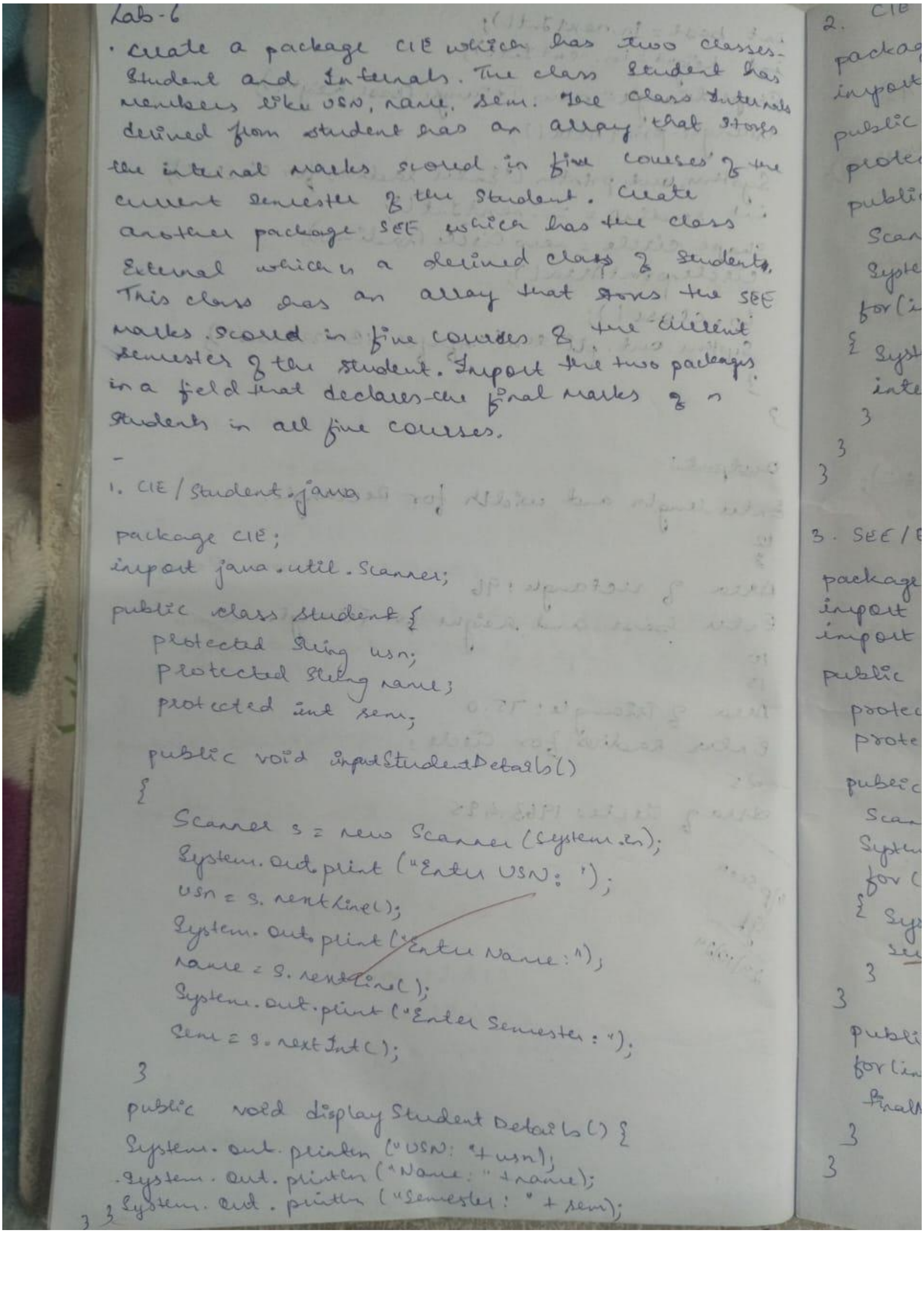
```
D:\1bm23cs015\LAB 5>java Bank
enter customer name:
afeefah
enter accno:
36472
enter initial balance:
50000
enter minimum balance:
1000
enter interest rate:
1.5
enter service charge:
10
Enter choice:
 1.Current acc
 2.Savings acc
1
Customer name is:afeefah
Account number:36472
AFEEFAH - 1BM23CS015
account is current type
enter choice:
 1.deposit
 2.withdraw
 3.display balance
1
enter amount to be deposited:
25000
Deposited: 25000.0
enter choice:
 1.deposit
 2.withdraw
 3.display balance
2
enter amount to withdraw:
40000
withdrew:40000.0 balance is:35000.0
enter choice:
 1.deposit
 2.withdraw
 3.display balance
3
```



```
D:\1BM23CS015\LAB 5>java Bank
enter customer name:
afeefah
enter accno:
45637
enter initial balance:
200000
enter minimum balance:
1000
enter interest rate:
3
enter service charge:
12
Enter choice:
 1.Current acc
 2.Savings acc
2
Customer name is:afeefah
Account number:45637
AFEEFAH - 1BM23CS015
account is savings type
enter choice:
 1.deposit
 2.withdraw
 3.display balance
1
enter amount to be deposited:
40000
Deposited: 40000.0
enter choice:
 1.deposit
 2.withdraw
 3.display balance
2
enter amount to withdraw:
25000
withdrew:25000.0 balance is:215000.0
enter choice:
 1.deposit
 2.withdraw
 3.display balance
3
Deposited: 6450.0
Current Balance: 221450.0
```

Program 6:

Implement Packages



Lab-6

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

-

1. CIE / Student.java

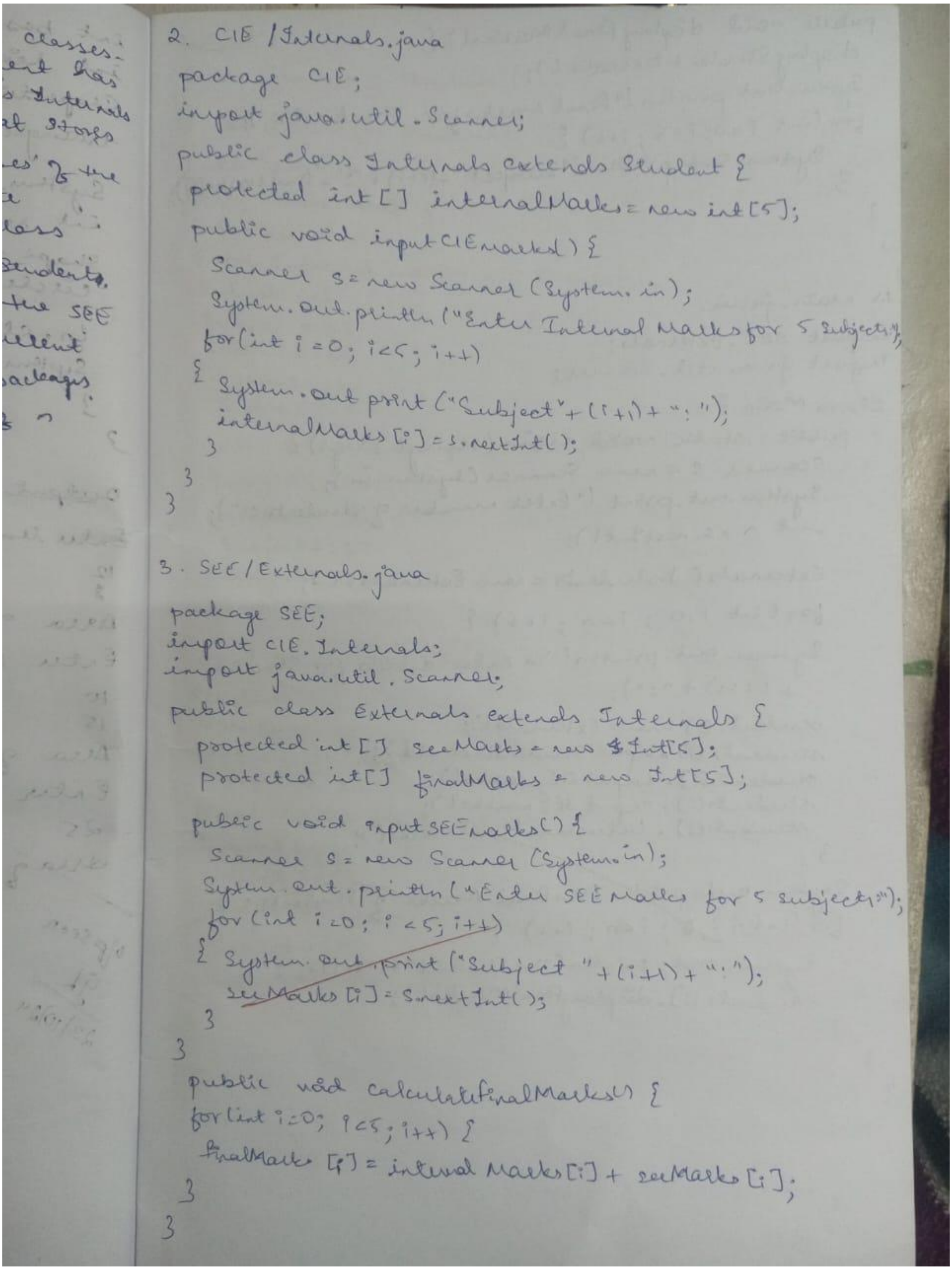
```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;

  public void inputStudentDetails()
  {
     Scanner s = new Scanner(System.in);
     System.out.print("Enter USN: ");
     usn = s.nextLine();
     System.out.print("Enter Name: ");
     name = s.nextLine();
     System.out.print("Enter Semester: ");
     sem = s.nextInt();
  }
 public void displayStudentDetails() {
 System.out.println("USN: "+ usn);
 System.out.println("Name: " + name);
 System.out.println("Semester: " + sem);
} }
```

2. CIE

```
packag
import
public
protec
publi
 Scan
 Syste
 for(i
 { Syst
  inte
  }
 }
}
```

3. SEE/E

```
package
import
import
public
 protec
 prote
 public
  Scan
  Syste
  for(
  { Sy
   se
  }
 }
 publi
 for(in
  finalM
 }
}
```

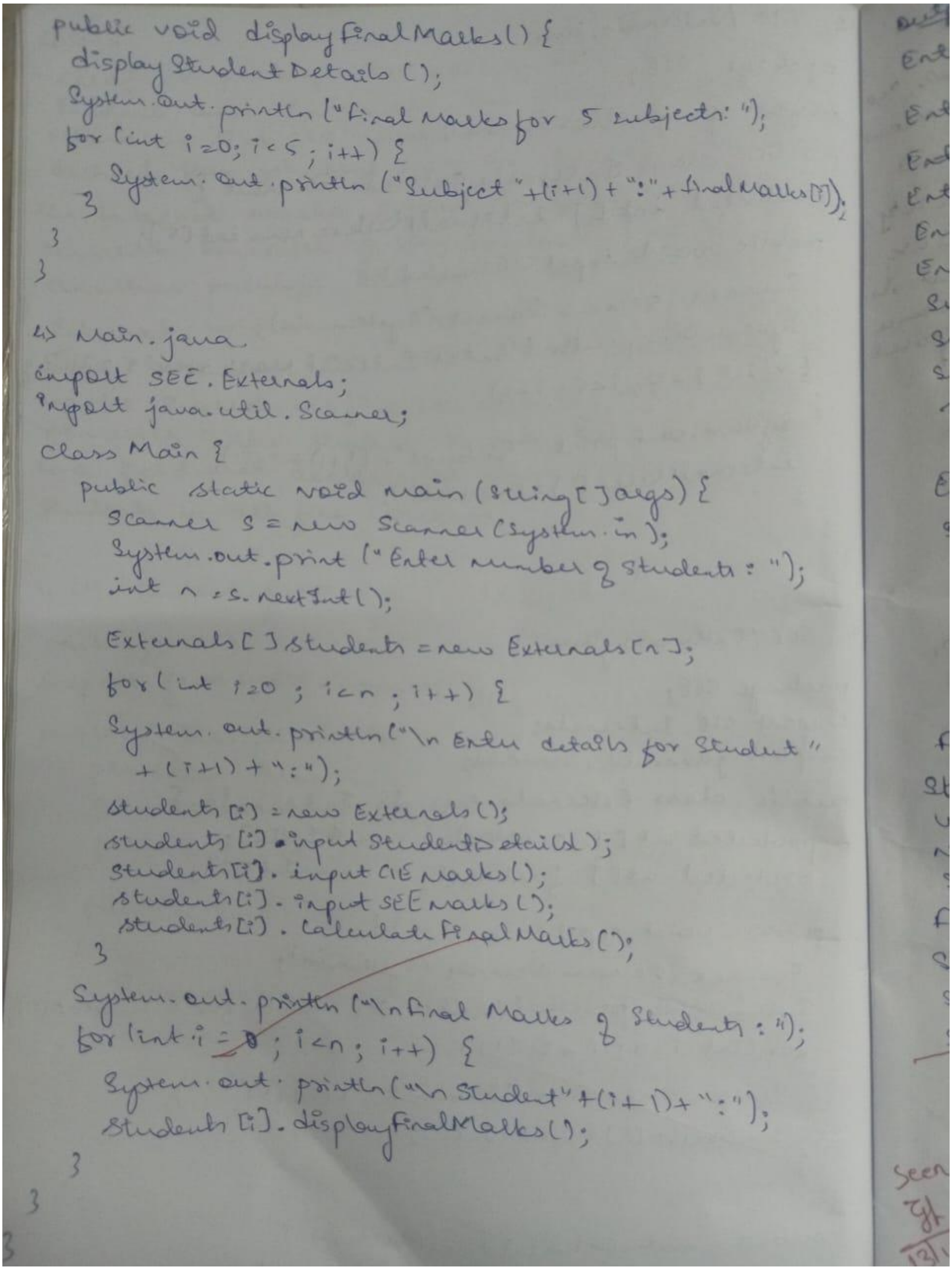


2. CIE /Internals.java

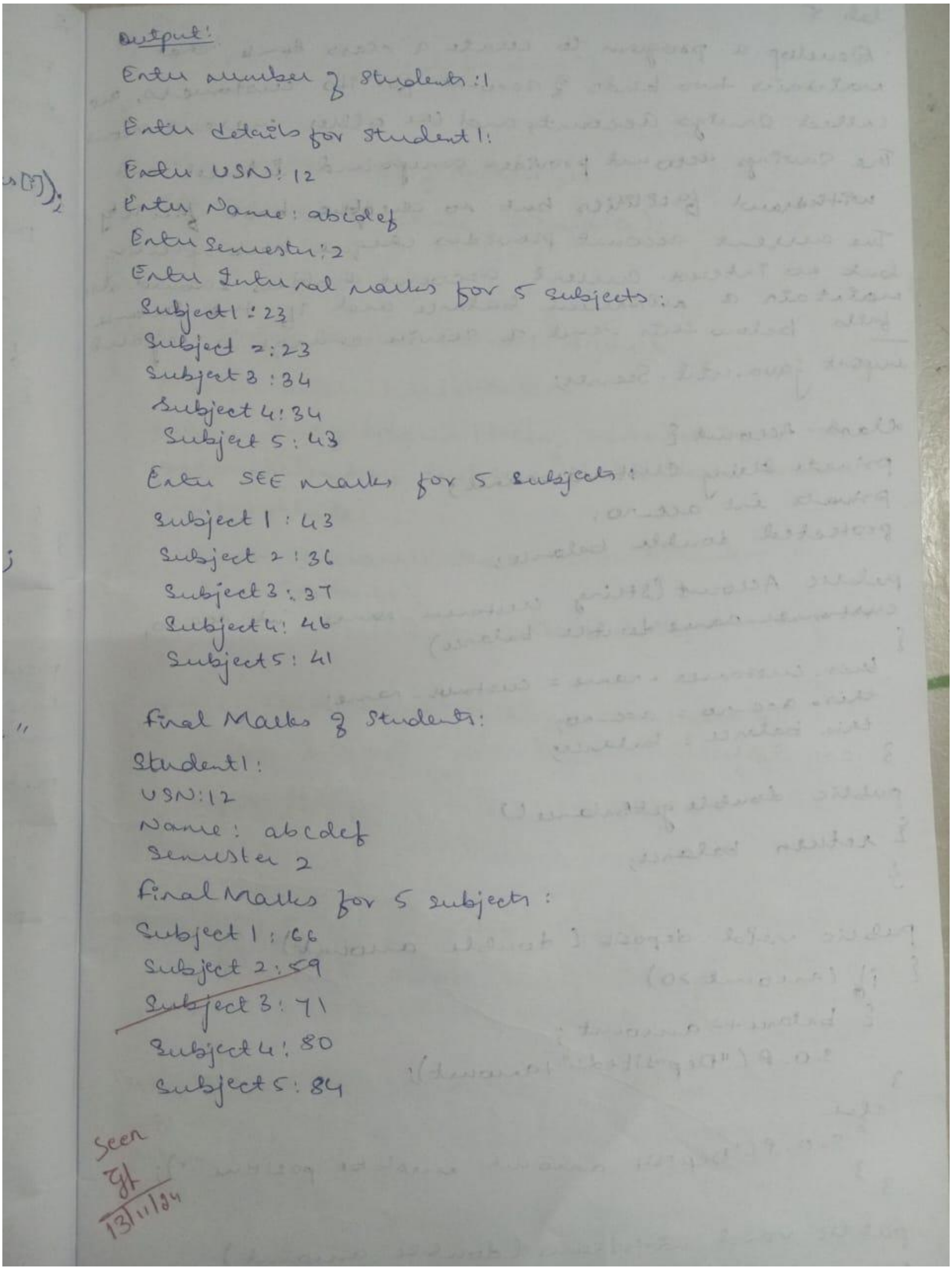
```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
protected int [] internalMarks= new int[5];
public void inputCIEmarks() {
  Scanner s= new Scanner (System.in);
  System.out.println ("Enter Internal Marks for 5 subjects:");
  for(int i =0; i<5; i++)
  {
   System.out.print ("Subject"+ (i+1)+ ": ");
   internalMarks [i] = s.nextInt();
  }
 }
}
```

3. SEE/Externals.java

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

  public void inputSEEmarks() {
   Scanner s= new Scanner (System.in);
   System.out.println ("Enter SEE Marks for 5 subjects:");
   for (int i=0; i<5; i++)
   { System.out.print ("Subject "+(i+1)+ ": ");
    seeMarks [i] = s.nextInt();
   }
 }
 public void calculateFinalMarks() {
 for(int i=0; i<5; i++) {
  finalMarks [i] = internal Marks[i] + seeMarks [i];
 }
}
```



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

4) Main.java
import SEE.Externals;
import java.util.Scanner;
class Main {
  public static void main(String[]args) {
    Scanner s = new Scanner(System.in);
    System.out.print("Enter number of students: ");
    int n = s.nextInt();

    Externals[] students = new Externals[n];
    for(int i=0 ; i<n ; i++) {
    System.out.println("\n Enter details for Student "
      + (i+1) + ":");
    students[i] = new Externals();
    students[i].inputStudentDetails();
    students[i].inputCIEMarks();
    students[i].inputSEEMarks();
    students[i].calculateFinalMarks();
    }
  System.out.println("\nFinal Marks of Students: ");
  for (int i = 0; i<n; i++) {
    System.out.println("\n Student"+(i+1)+":");
    Students[i].displayFinalMarks();
  }
}
}
```



Output:

```
Enter number of students:1
Enter details for student1:
Enter USN: 12
Enter Name: abcdef
Enter Semester: 2
Enter Internal marks for 5 subjects:
Subject1: 23
Subject 2: 23
Subject 3: 34
Subject 4: 34
Subject 5: 43
Enter SEE marks for 5 subjects:
Subject 1: 43
Subject 2: 36
Subject3: 37
Subject4: 46
Subject5: 41
Final Marks of Students:
Student1:
USN:12
Name: abcdef
Semester 2
Final Marks for 5 subjects:
Subject 1: 66
Subject 2: 59
Subject 3: 71
Subject4: 80
Subject5: 84
```

Seen
13/11/24

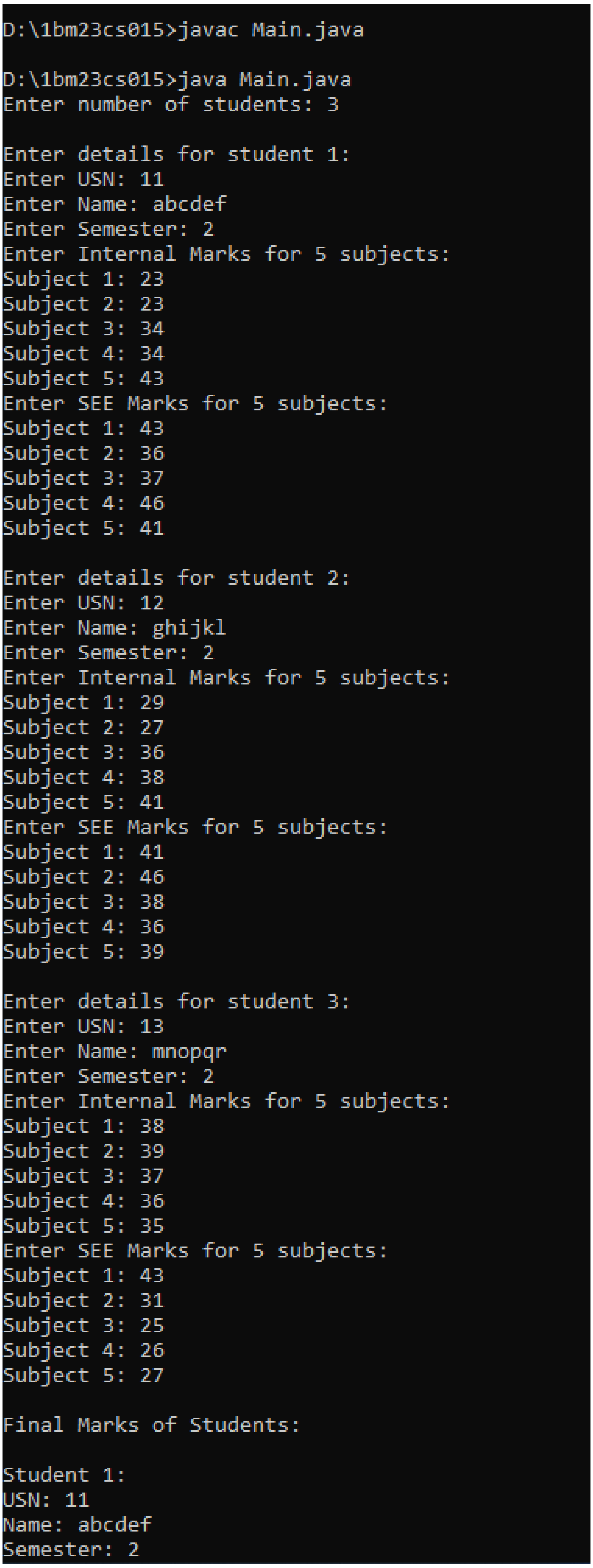
Code:

//CIE/Student.java

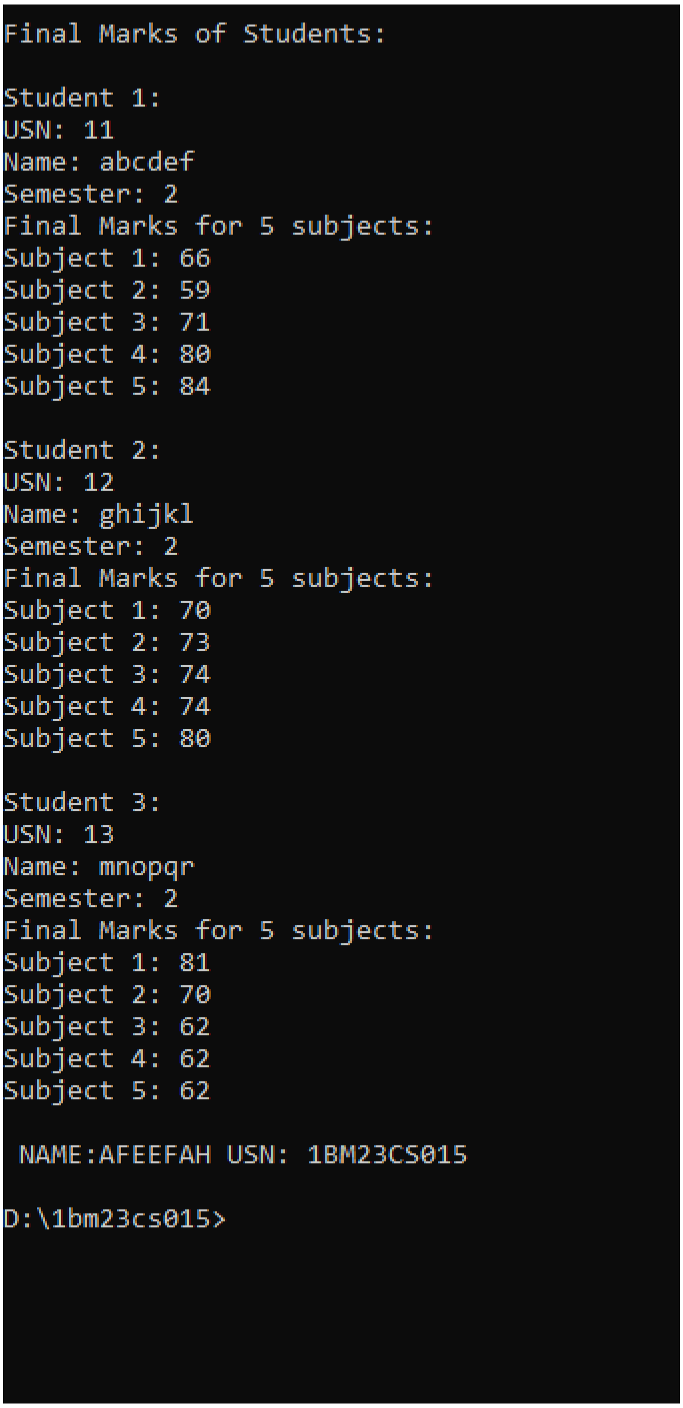
package CIE;

```
import java.util.Scanner;

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

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

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

//CIE/Internals.java
package CIE;
import java.util.Scanner;
```

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

    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            internalMarks[i] = s.nextInt();
        }
    }
}

//SEE/Externals.java
package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    private int[] seeMarks = new int[5];
    private int[] finalMarks = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + seeMarks[i];
        }
    }

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

//Main.java
import SEE.Externals;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = s.nextInt();

        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            students[i] = new Externals();
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayFinalMarks();

        }
System.out.println("\n NAME:AFEEFAH USN: 1BM23CS015");
    }
}
```



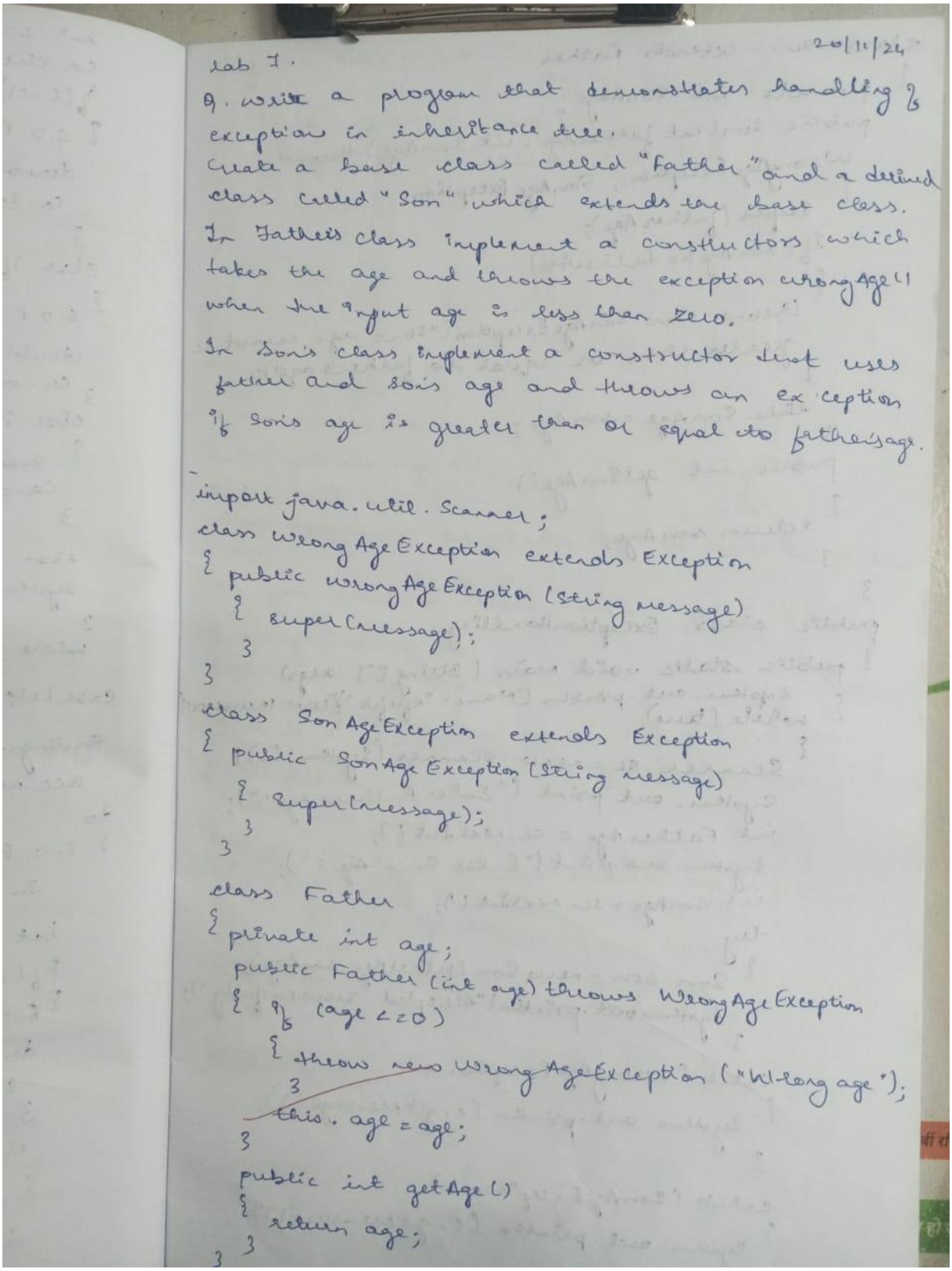
```
D:\1bm23cs015>javac Main.java

D:\1bm23cs015>java Main.java
Enter number of students: 3

Enter details for student 1:
Enter USN: 11
Enter Name: abcdef
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 23
Subject 2: 23
Subject 3: 34
Subject 4: 34
Subject 5: 43
Enter SEE Marks for 5 subjects:
Subject 1: 43
Subject 2: 36
Subject 3: 37
Subject 4: 46
Subject 5: 41

Enter details for student 2:
Enter USN: 12
Enter Name: ghijkl
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 29
Subject 2: 27
Subject 3: 36
Subject 4: 38
Subject 5: 41
Enter SEE Marks for 5 subjects:
Subject 1: 41
Subject 2: 46
Subject 3: 38
Subject 4: 36
Subject 5: 39

Enter details for student 3:
Enter USN: 13
Enter Name: mnopqr
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 38
Subject 2: 39
Subject 3: 37
Subject 4: 36
Subject 5: 35
Enter SEE Marks for 5 subjects:
Subject 1: 43
Subject 2: 31
Subject 3: 25
Subject 4: 26
Subject 5: 27

Final Marks of Students:

Student 1:
USN: 11
Name: abcdef
Semester: 2
```



```
Final Marks of Students:

Student 1:
USN: 11
Name: abcdef
Semester: 2
Final Marks for 5 subjects:
Subject 1: 66
Subject 2: 59
Subject 3: 71
Subject 4: 80
Subject 5: 84

Student 2:
USN: 12
Name: ghijkl
Semester: 2
Final Marks for 5 subjects:
Subject 1: 70
Subject 2: 73
Subject 3: 74
Subject 4: 74
Subject 5: 80

Student 3:
USN: 13
Name: mnopqr
Semester: 2
Final Marks for 5 subjects:
Subject 1: 81
Subject 2: 70
Subject 3: 62
Subject 4: 62
Subject 5: 62

 NAME:AFEEFAH USN: 1BM23CS015

D:\1bm23cs015>
```

Program 7:

Implement Exception Handling



lab 7. 20/11/24

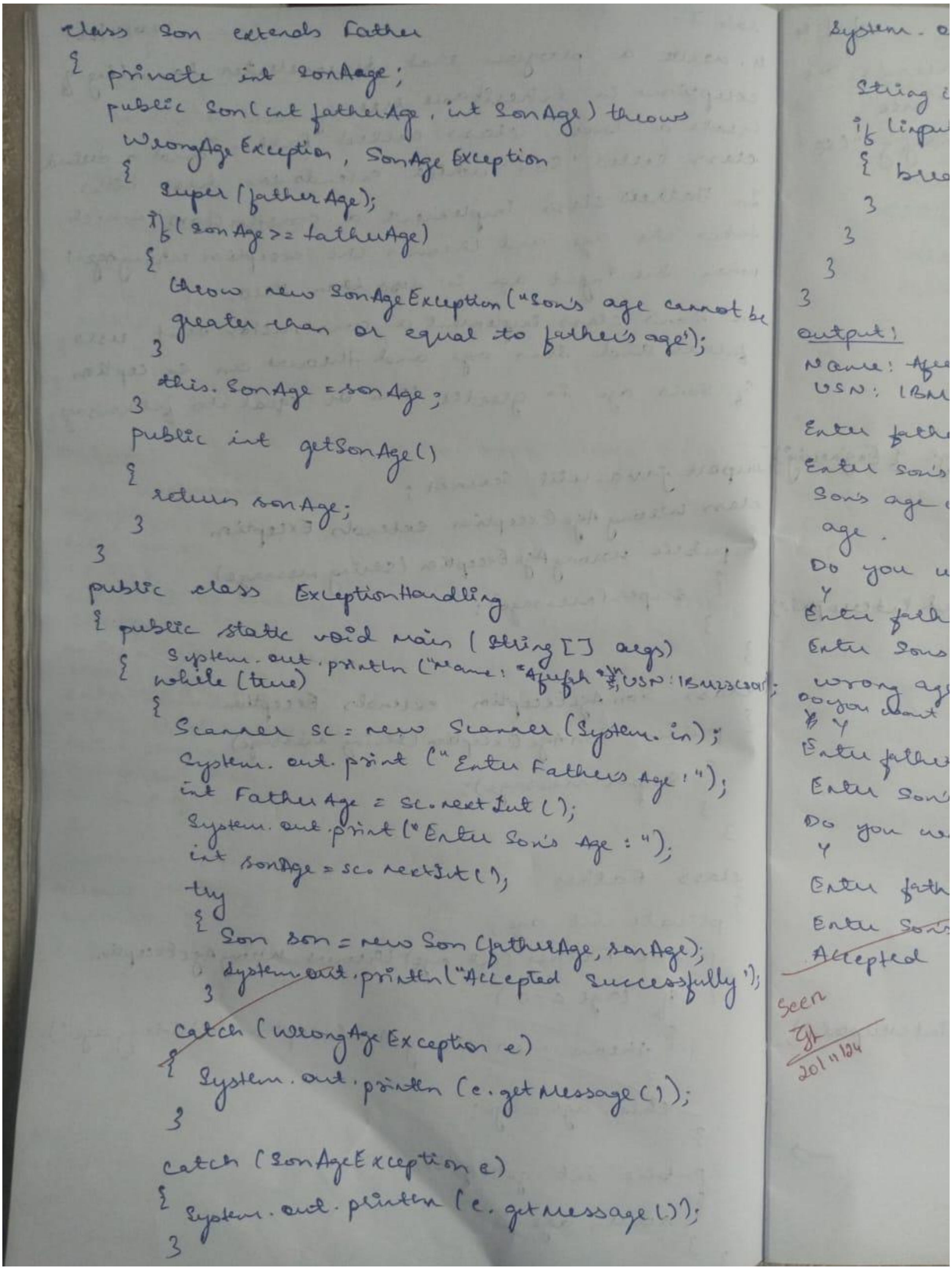
Q. write a program that demonstrates handling of exception in inheritance tree.

Create a base class called "Father" and a derived class called "Son" which extends the base class.

In Father's class implement a constructors which takes the age and throws the exception wrongAge() when the input age is less than zero.

In Son's class implement a constructor that uses father and son's age and throws an exception if son's age is greater than or equal to father's age.

```
import java.util.Scanner;
class WrongAgeException extends Exception
{ public wrongAgeException (String message)
  { super(message);
  }
}
class SonAgeException extends Exception
{ public SonAgeException (String message)
  { super(message);
  }
}

class Father
{ private int age;
  public Father (int age) throws WrongAgeException
  { if (age <=0)
    { throw new WrongAgeException ("Wrong age");
    }
    this.age = age;
  }
  public int getAge()
  {
    return age;
  }
}
```



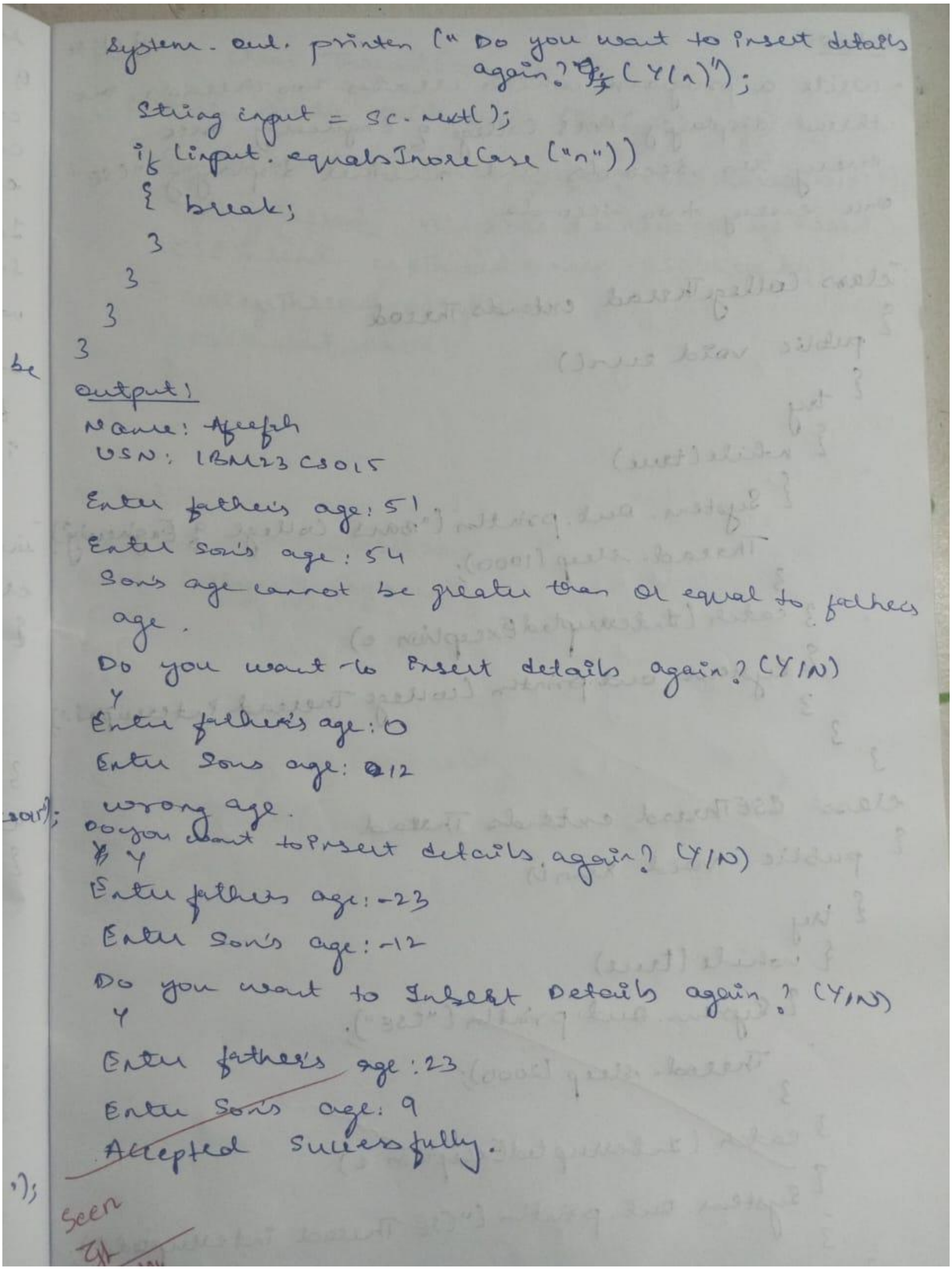
```
class Son extends Father
{ private int sonAage;
  public Son(int fatherAge, int SonAge) throws
   WrongAge Exception, SonAge Exception
   {
    Super (fatherAge);
    if ( sonAge >= fatherAge)
    {
     throw new SonAgeException ("Son's age cannot be
     greater than or equal to father's age");
    }
    this. SonAge = sonAge;
   }
   public int getSonAge()
   {
    return sonAge;
   }
}
public class ExceptionHandling
{ public static void main ( String [] args)
  {  System.out.println ("Name: Afeefah \n USN: 1BM23CS0[illegible]");
     while (true)
     {
      Scanner sc = new Scanner (System. in);
      System. out. print (" Enter Fathers Age: ");
      int FatherAge = sc. nextInt ();
      System. out. print (" Enter Son's Age : ");
      int sonAge = sc. nextInt ();
      try
      {
       Son son = new Son (fatherAge, sonAge);
       System.out.println("Accepted Successfully");
      }
     catch (WrongAge Exception e)
     {
      System. out. println (c. getMessage ());
     }
     catch (SonAgeException e)
     {
      System. out. println (c. getMessage ());
     }
```

```
System. o
  String i
  if (inpu
  { bre
  }
 }
}
}
```

output:

```
Name: Afee
USN: 1BM
Enter fath
Enter Son's
Son's age c
age.
Do you w
Y
Enter fath
Enter Sons
wrong ag
Do you want
Y
Enter father
Enter Son's
Do you w
Y
Enter fath
Enter Son's
Accepted
```

Seen
20/11/24



```
System.out.printen ("Do you want to insert details
                            again? (Y/n)");
  String input = sc.next();
  if (input.equalsInoreCase("n"))
  { break;
   }
  }
 }
}
```

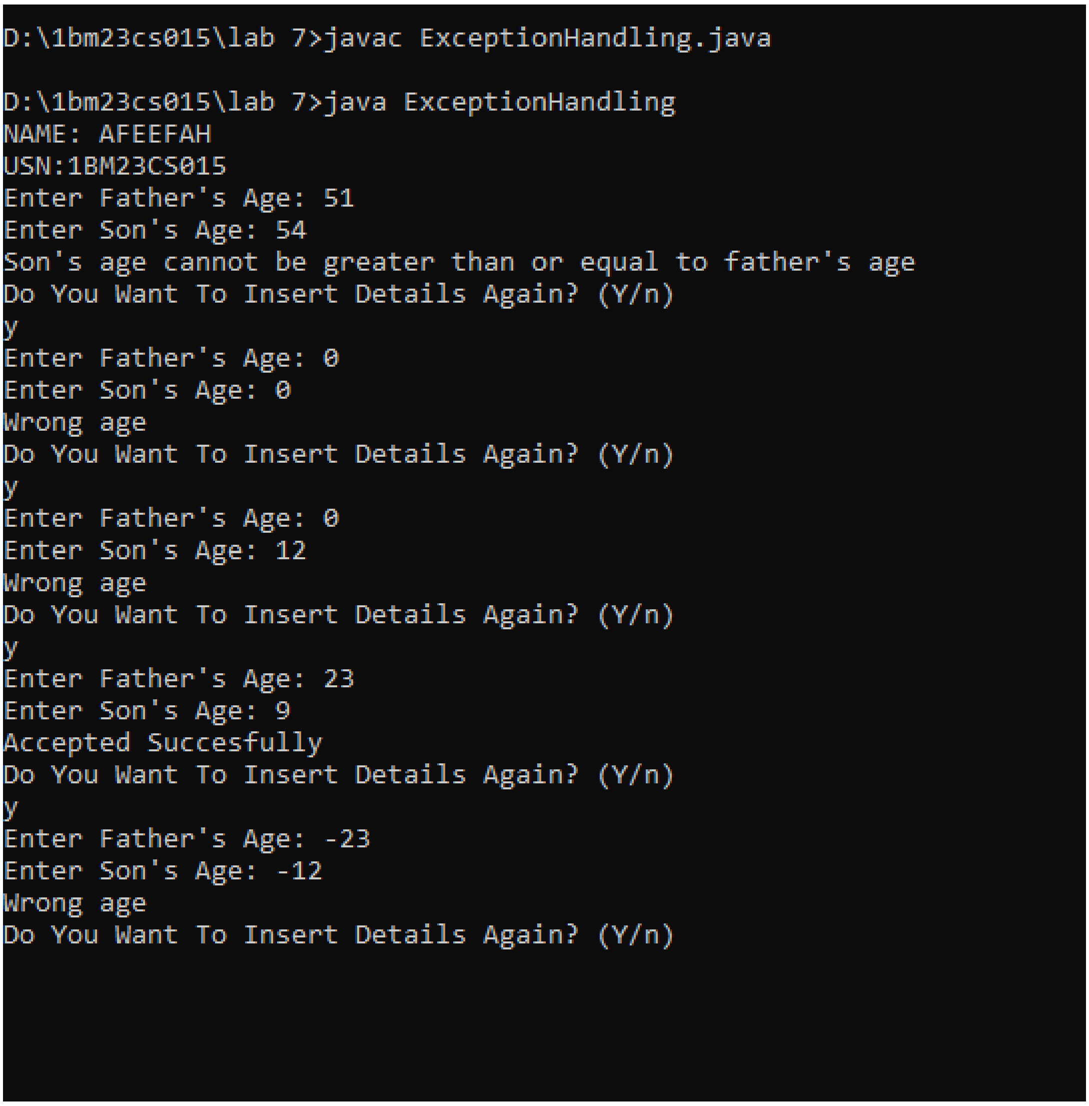
output:

```
Name: Afeefah
USN: 1BM23CS015

Enter father's age: 51
Enter son's age: 54
Son's age cannot be greater than or equal to fathers
age.
Do you want to insert details again? (Y/N)
Y
Enter father's age: 0
Enter Sons age: 12
wrong age.
Do you want to insert details again? (Y/N)
Y
Enter fathers age: -23
Enter Son's age: -12
Do you want to Insert Details again? (Y/N)
Y
Enter father's age: 23
Enter Son's age: 9
Accepted Successfully.
```

Seen

Code:

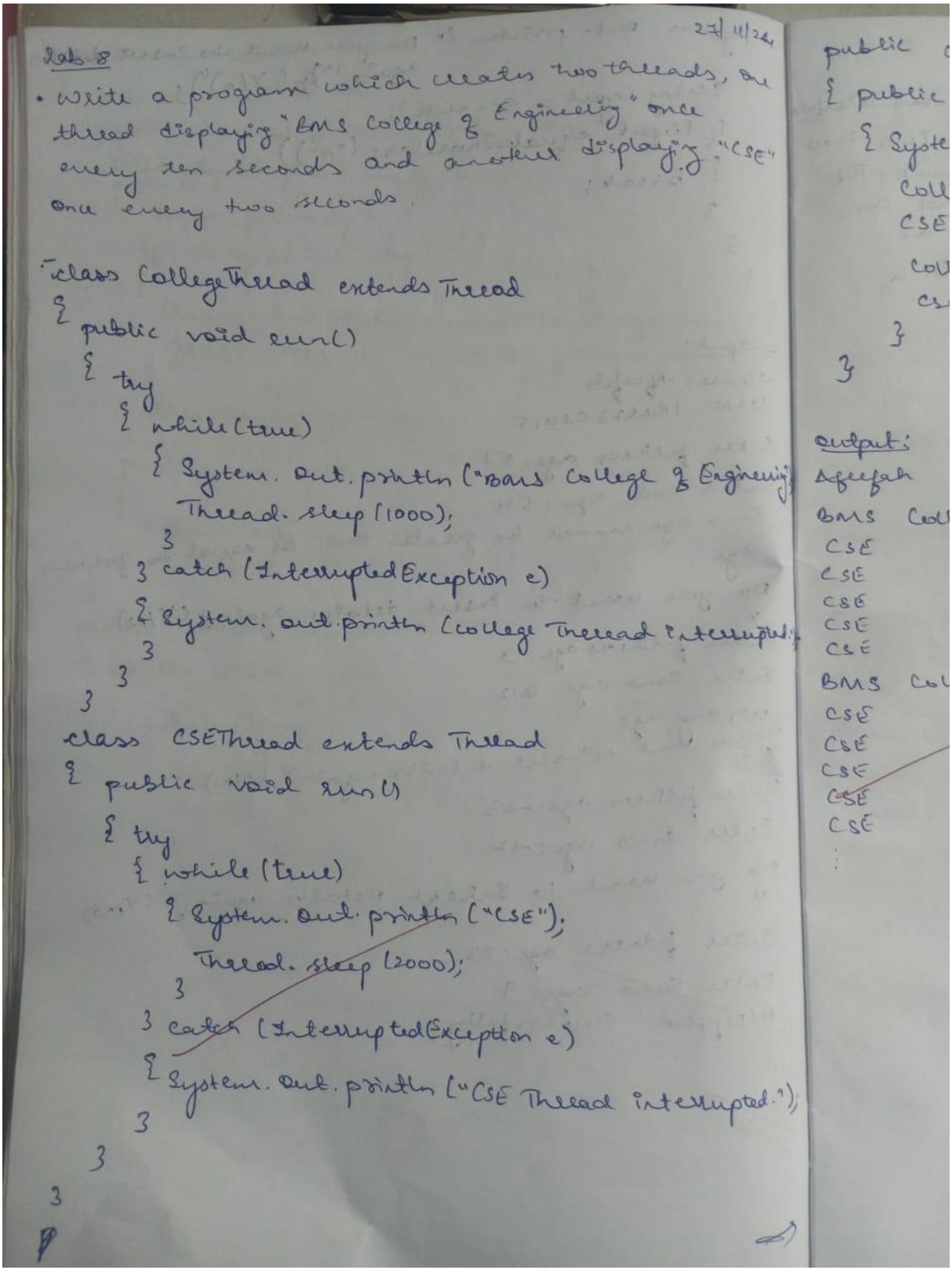
```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age <= 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal
to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class ExceptionHandling{
    public static void main(String[] args)
{
System.out.println("NAME: AFEEFAH\nUSN:1BM23CS015");

        while(true)
{
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Do You Want To Insert Details Again? (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```



```
D:\1bm23cs015\lab 7>javac ExceptionHandling.java

D:\1bm23cs015\lab 7>java ExceptionHandling
NAME: AFEEFAH
USN:1BM23CS015
Enter Father's Age: 51
Enter Son's Age: 54
Son's age cannot be greater than or equal to father's age
Do You Want To Insert Details Again? (Y/n)
y
Enter Father's Age: 0
Enter Son's Age: 0
Wrong age
Do You Want To Insert Details Again? (Y/n)
y
Enter Father's Age: 0
Enter Son's Age: 12
Wrong age
Do You Want To Insert Details Again? (Y/n)
y
Enter Father's Age: 23
Enter Son's Age: 9
Accepted Succesfully
Do You Want To Insert Details Again? (Y/n)
y
Enter Father's Age: -23
Enter Son's Age: -12
Wrong age
Do You Want To Insert Details Again? (Y/n)
```

Program 8:

Multithreading, Creating Threads in Java



27/11/24

Lab 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.

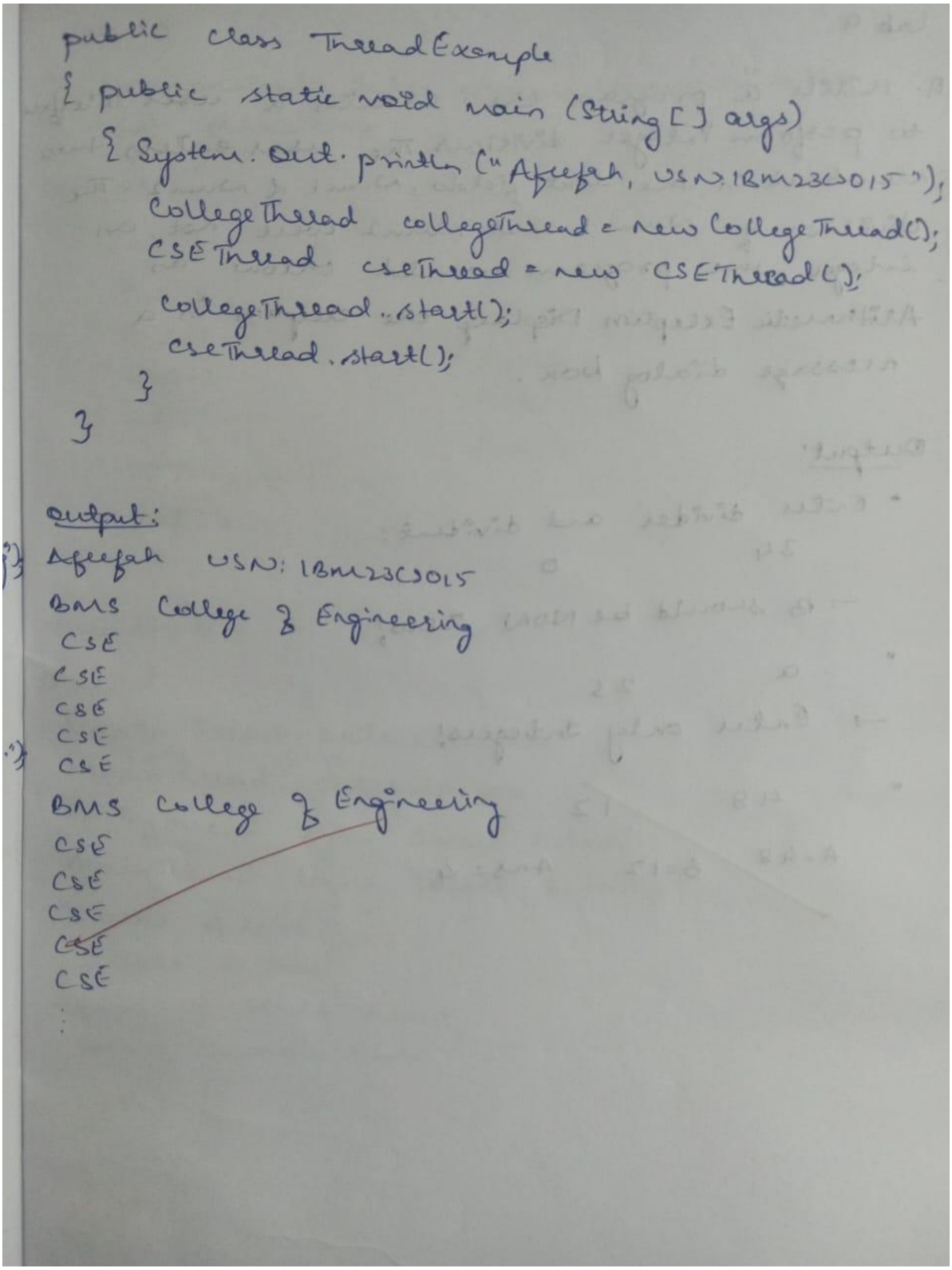
```
class CollegeThread extends Thread
{
  public void run()
  {  try
     {  while(true)
        {  System.out.println("BMS College of Engineering
           Thread.sleep(1000);
        }
     } catch (InterruptedException e)
     { System.out.println(College Thread interrupted.
     }
  }
}
class CSEThread 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 c
{ public
  { Syste
     Coll
     CSE

     Col
     Cs
  }
}
```

output:

Afeefah

```
BMS  Coll
CSE
CSE
CSE
CSE
CSE
BMS  Col
CSE
CSE
CSE
CSE
CSE
 ⋮
```



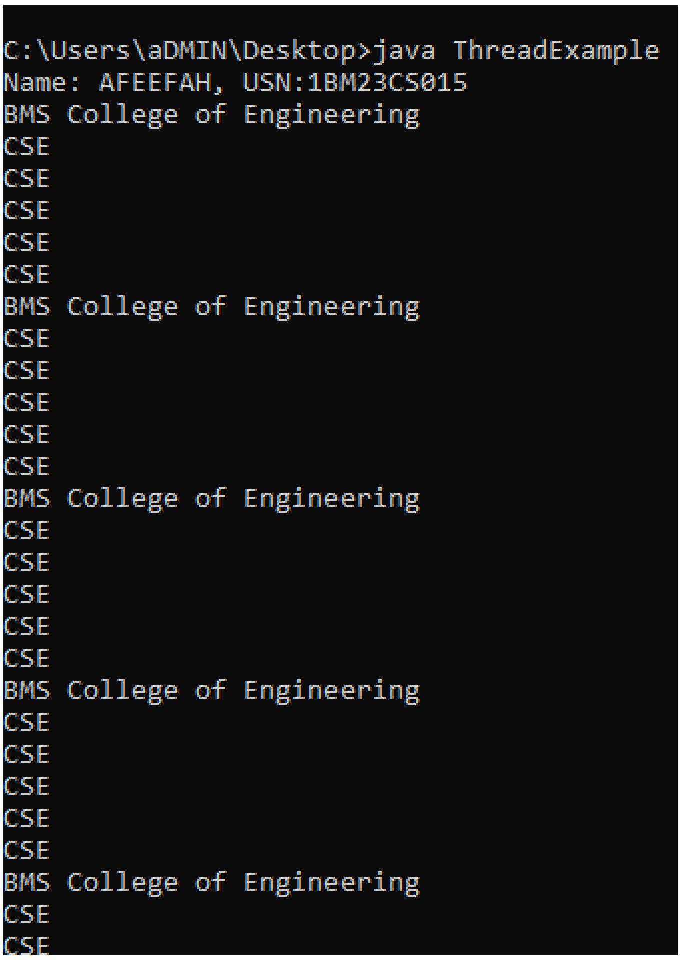
```
public class ThreadExample
{ public static void main (String[] args)
  { System.out.println("Afeefah, USN:1BM23CS015");
    CollegeThread collegeThread = new CollegeThread();
    CSEThread cseThread = new CSEThread();
    collegeThread.start();
    cseThread.start();
  }
}
```

output:

```
Afeefah USN: 1BM23CS015
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
:
```

Code:

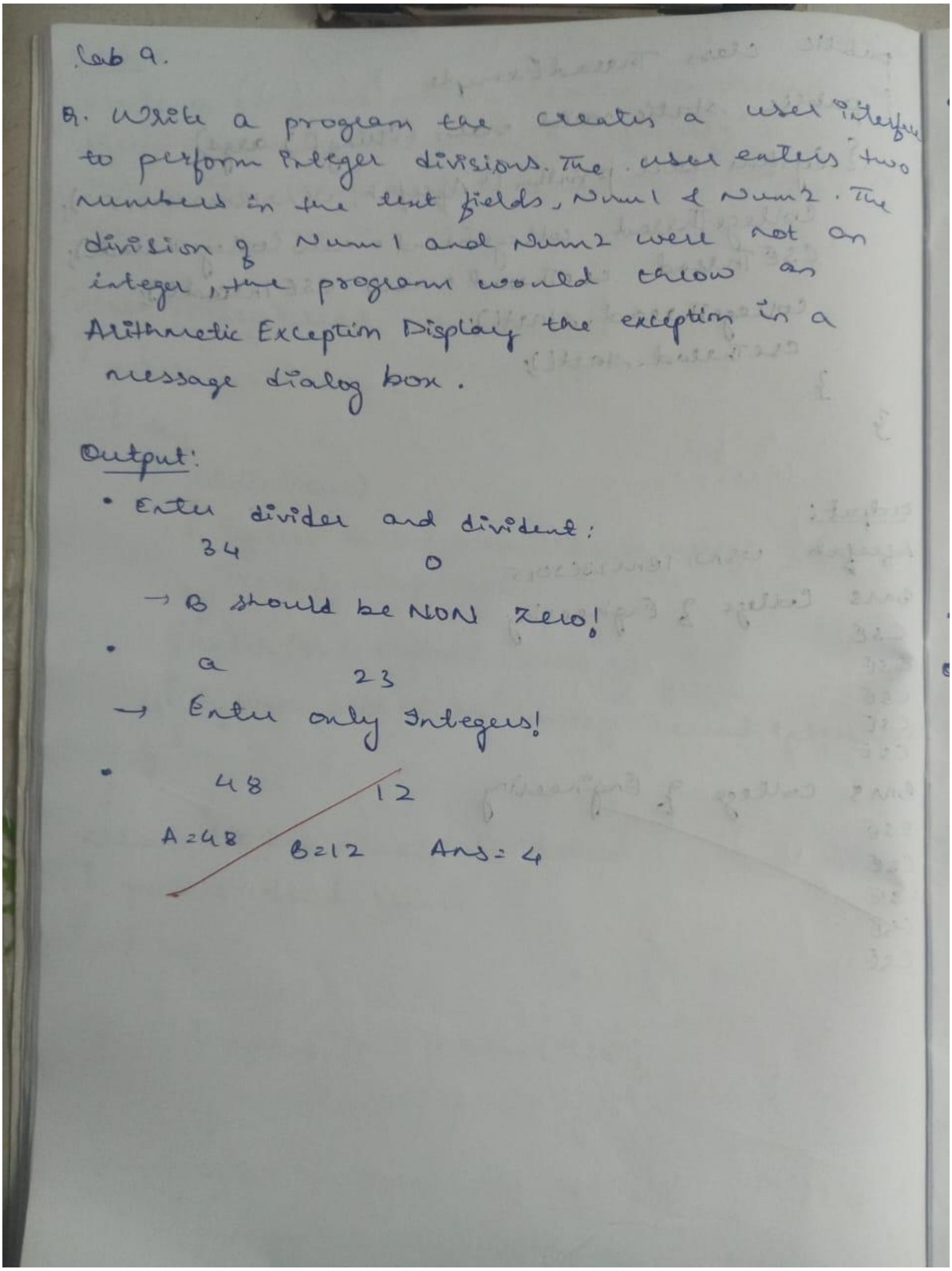
```
class CollegeThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("CollegeThread interrupted.");
        }
    }
}
class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted.");
        }
    }
}

public class ThreadExample {
    public static void main(String[] args) {
System.out.println("Name: AFEEFAH, USN:1BM23CS015");
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();


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



```
C:\Users\aDMIN\Desktop>java ThreadExample
Name: AFEEFAH, USN:1BM23CS015
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

Program 9:

Interface to Perform Integer Division



Lab 9.

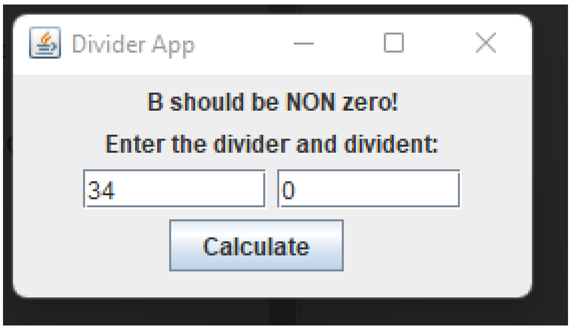
Q. Write a program the creates a user interface to perform Integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 and Num2 were not an integer, the program would throw as Arithmetic Exception Display the exception in a message dialog box.

Output:

- Enter divider and divident:
  34 0
  → B should be NON Zero!
- a 23
  → Enter only Integers!
- 48 12
  A=48 B=12 Ans= 4

Code:

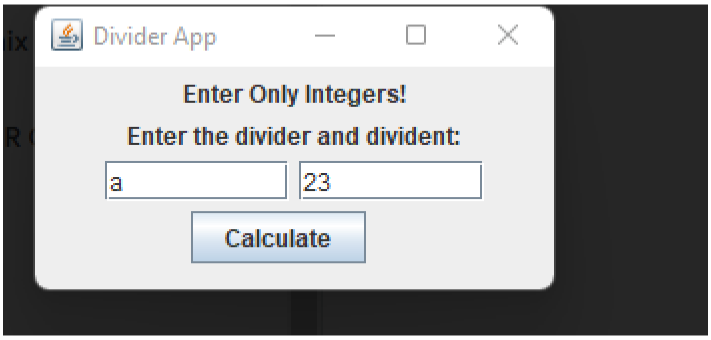
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

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



B should be NON zero!

Enter the divider and divident:

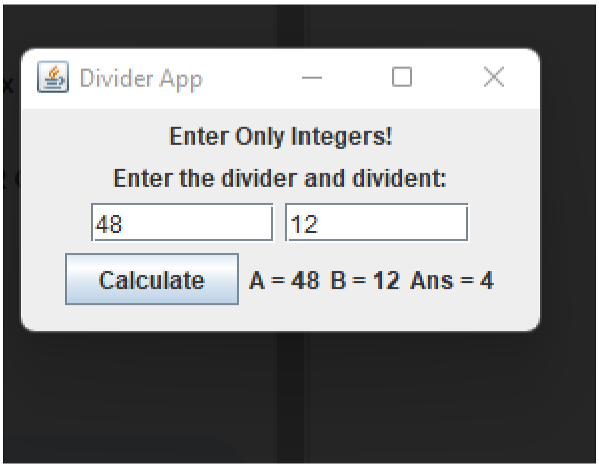
34 | 0



Enter Only Integers!

Enter the divider and divident:

a | 23



Enter Only Integers!

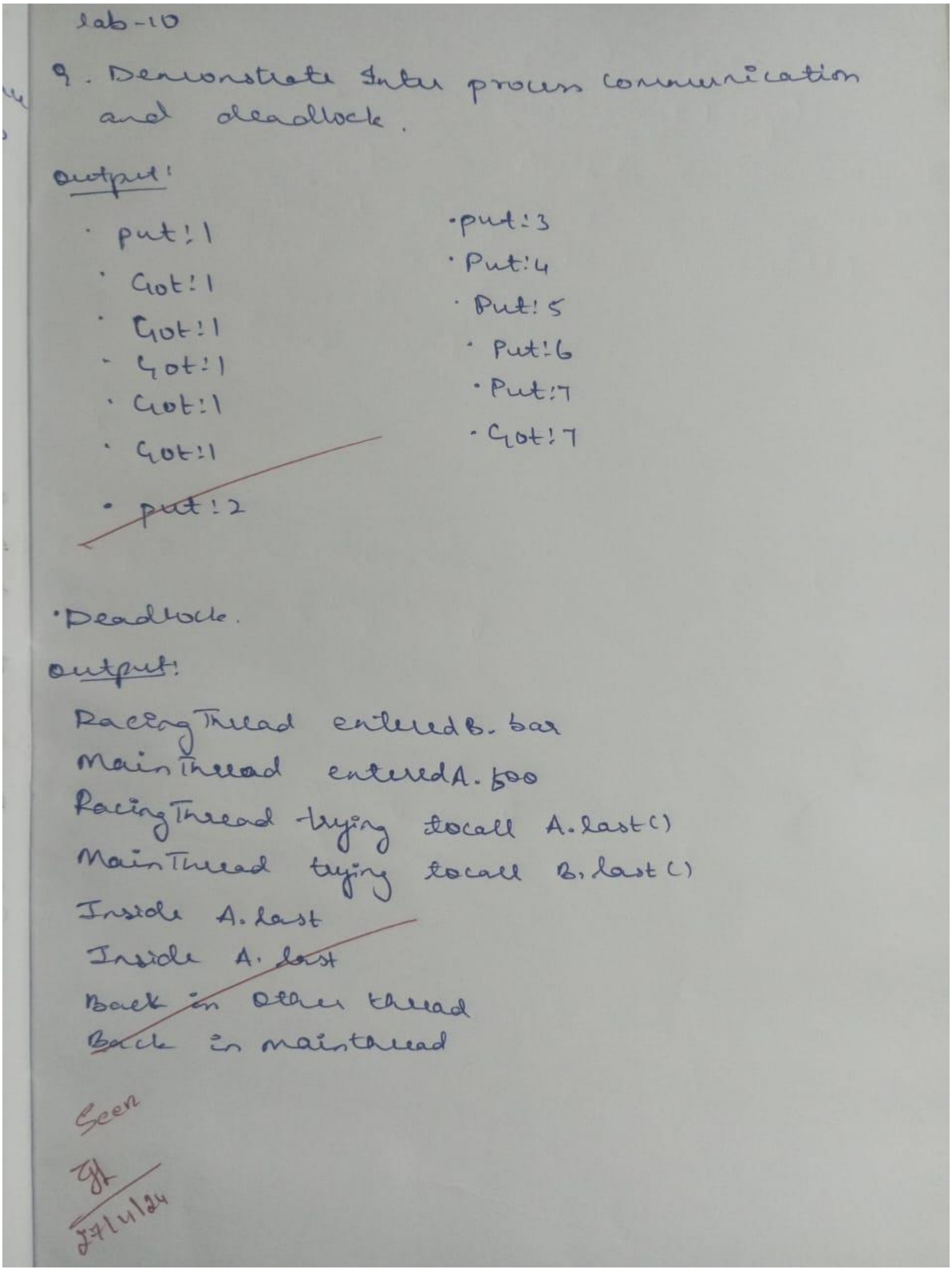
Enter the divider and divident:

48 | 12

A = 48 B = 12 Ans = 4

Program 10:

Implement Deadlock and Inter-process Communication



lab-10

9. Demonstrate Inter process communication and deadlock.

Output:

```
· put: 1
· Got: 1
· Got: 1
· Got: 1
· Got: 1
· Got: 1
· put: 2
·put: 3
· Put: 4
· Put: 5
· Put: 6
· Put: 7
· Got: 7
```

·Deadlock.

Output:

```
RacingThread  enteredB. bar
MainThread  enteredA. foo
RacingThread  trying  tocall  A.last()
MainThread  trying  tocall  B, last ()
Inside  A. last
Inside  A. last
Back  in  Other  thread
Back  in  mainthread
```

Seen

27/11/24

Code:

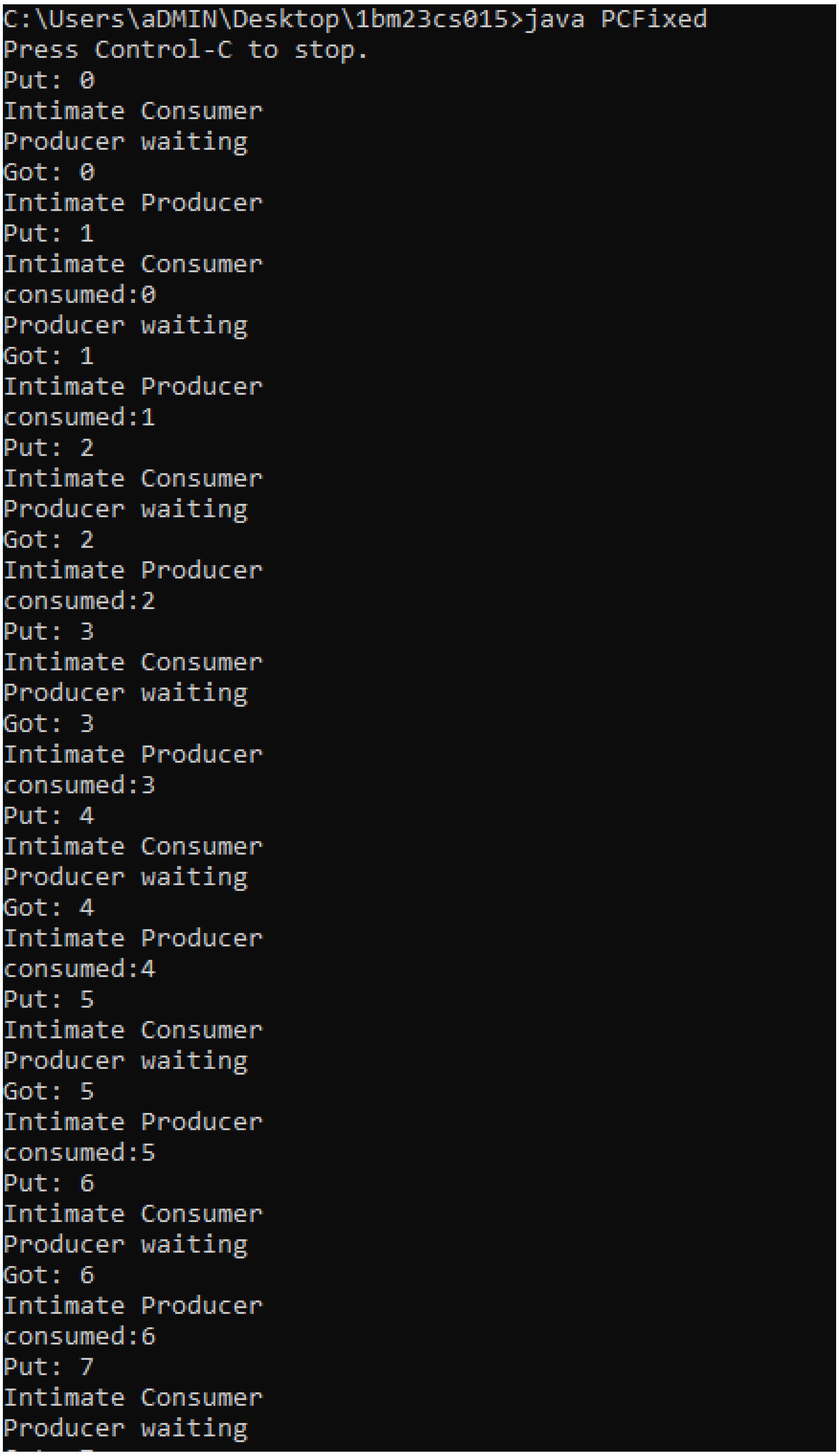
```
//PCFixed.java
class Q {
int n;
boolean valueSet = false;
synchronized int get() {
while(!valueSet)
try {
System.out.println("Consumer waiting");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedExceptioncaught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("Intimate Producer");
notify();
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
System.out.println("Producer waiting");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
```

```
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("Intimate Consumer");
notify();
}
}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}
class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
```

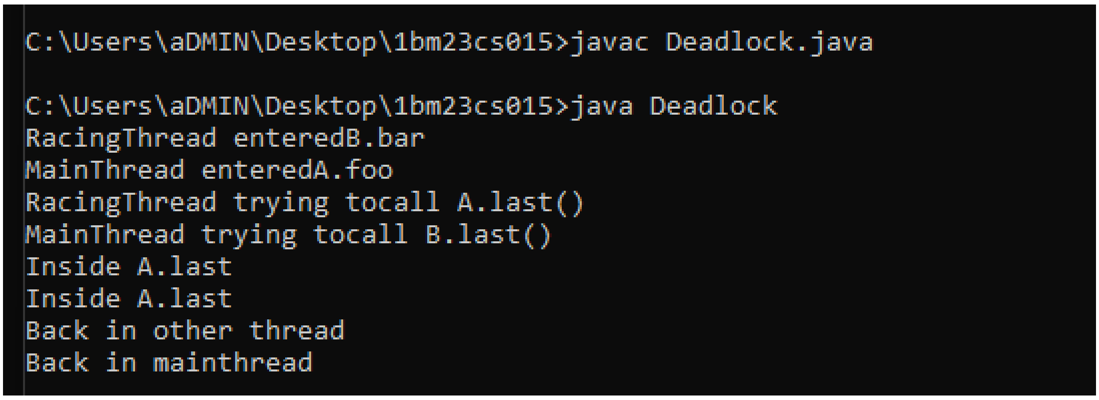
```
int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}
class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}

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

```
System.out.println(name + " trying tocall B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}
class B {
synchronized void bar(A a) {
String name =Thread.currentThread().getName();
System.out.println(name + " enteredB.bar");
try {
Thread.sleep(1000);
} catch(Exception e)
{
System.out.println("B Interrupted");
}
System.out.println(name + " trying tocall A.last()");
a.last();
}
void last() {
System.out.println("Inside A.last");
}
}
class Deadlock implements Runnable
{
A a = new A();
```

```
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in mainthread");
}
public void run() {
b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```



```
C:\Users\aDMIN\Desktop\1bm23cs015>java PCFixed
Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
consumed:0
Producer waiting
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer
consumed:2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate Producer
consumed:3
Put: 4
Intimate Consumer
Producer waiting
Got: 4
Intimate Producer
consumed:4
Put: 5
Intimate Consumer
Producer waiting
Got: 5
Intimate Producer
consumed:5
Put: 6
Intimate Consumer
Producer waiting
Got: 6
Intimate Producer
consumed:6
Put: 7
Intimate Consumer
Producer waiting
```



```
C:\Users\aDMIN\Desktop\1bm23cs015>javac Deadlock.java

C:\Users\aDMIN\Desktop\1bm23cs015>java Deadlock
RacingThread enteredB.bar
MainThread enteredA.foo
RacingThread trying tocall A.last()
MainThread trying tocall B.last()
Inside A.last
Inside A.last
Back in other thread
Back in mainthread
```