**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**ON**

**Object Oriented Java Programming**

**(23CS3PCOOJ )**

***Submitted by  
Ameena Yasmeen (1BM23CS027)***

***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 **Ameena Yasmeen(1BM23CS027),** 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 | Basavaraj Jakkali.  Professor & HOD  Department of CSE, BMSCE |

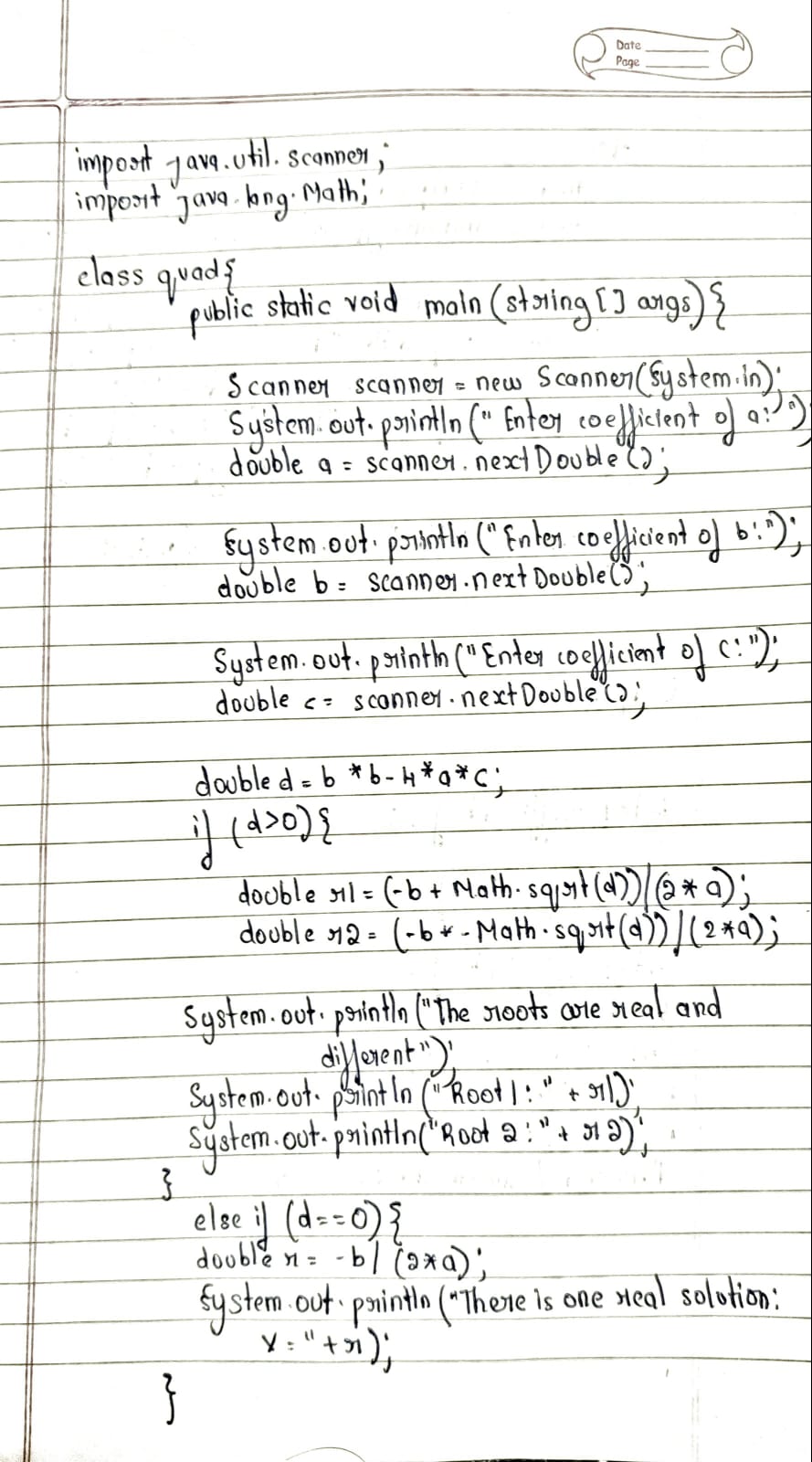
**Index**

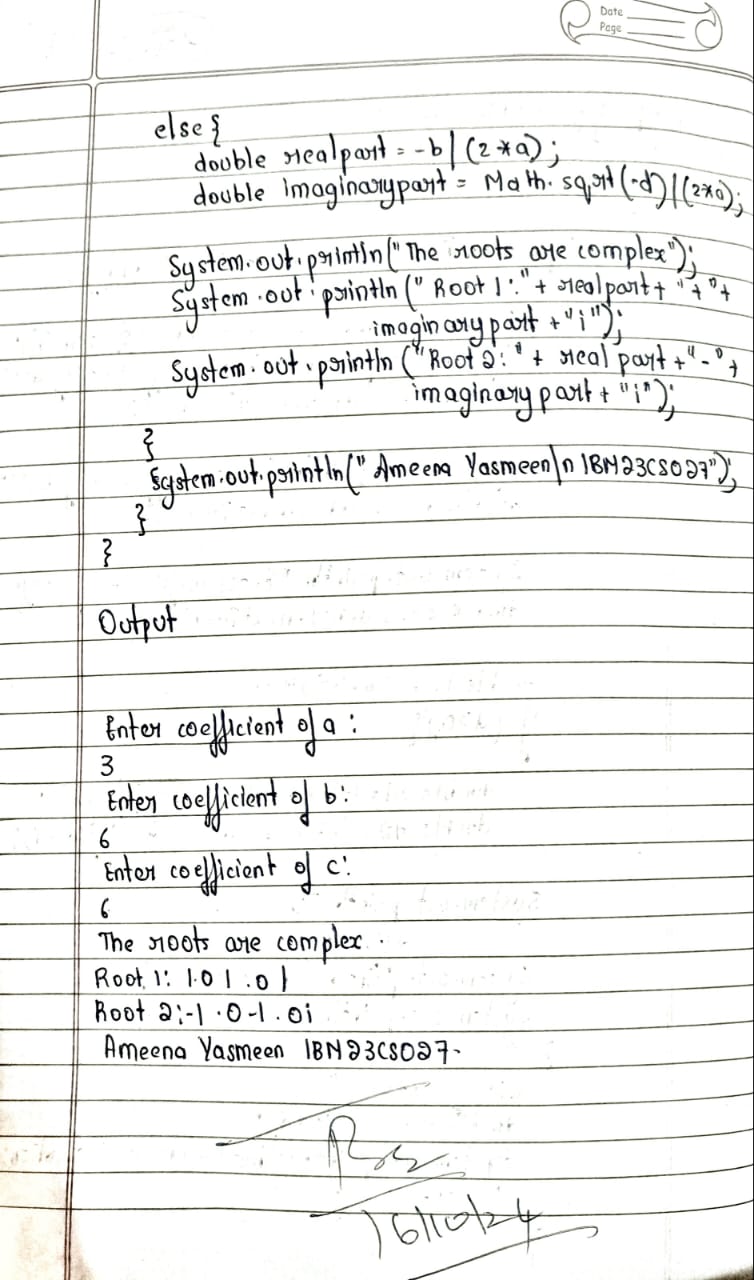
|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Date** | **Experiment Title** | **Page No.** |
| **1** | **9/10/24** | **Implement Quadratic Equation** | **4-6** |
| **2** | **16/10/24** | **Implement SGPA Calculator** | **6-11** |
| **3** | **23/10/24** | **Create Objects For Books** | **11-16** |
| **4** | **30/10/24** | **Implement Abstract Class** | **17-21** |
| **5** | **6/11/24** | **Bank Account Management** | **21-27** |
| **6** | **13/11/24** | **Implement Packages** | **28-39** |
| **7** | **20/11/24** | **Implement Exception Handling** | **40-50** |
| **8** | **27/11/24** | **Multithreading, Creating Threads In Java** | **51-65** |
| **9** | **27/11/24** | **Interface To Perform Integer Division** | **66-72** |
| **10** | **27/11/24** | **Implement Deadlock** | **73-86** |

[**https://github.com/Ameena1BM23CS27/java-lab/tree/main/Java%20lab%201**](https://github.com/Ameena1BM23CS27/java-lab/tree/main/Java%20lab%201)

**Program 1**

**Implement Quadratic Equation**

****

****

**Source Code:**

import java.util.Scanner;

public class QuadraticEquation {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("");

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 discriminant = b \* b - 4 \* a \* c;

if (a == 0) {

System.out.println("This is not a quadratic equation (a cannot be zero).");

} else {

if (discriminant > 0) {

double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

System.out.println("The roots are real and distinct:");

System.out.println("Root 1: " + root1);

System.out.println("Root 2: " + root2);

} else if (discriminant == 0) {

double root = -b / (2 \* a);

System.out.println("The roots are real and equal:");

System.out.println("Root: " + root);

} else {

double realPart = -b / (2 \* a);

double imaginaryPart = Math.sqrt(-discriminant) / (2 \* a);

System.out.println("The roots are complex:");

System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");

System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");

}

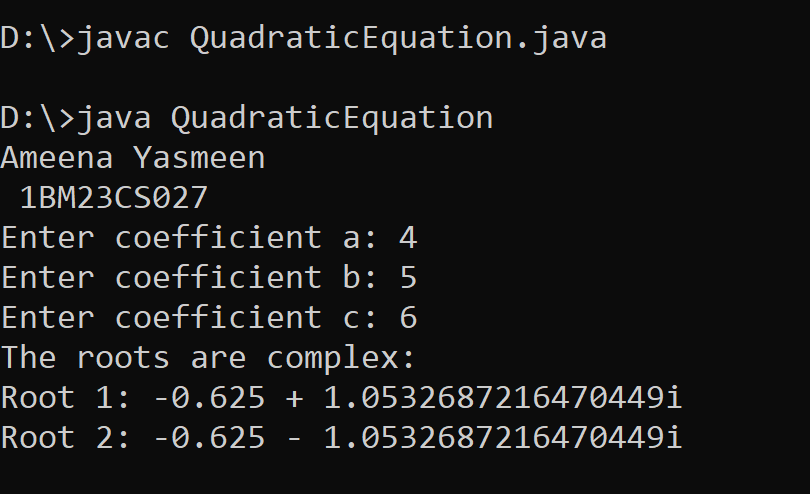
}

scanner.close();

}

}

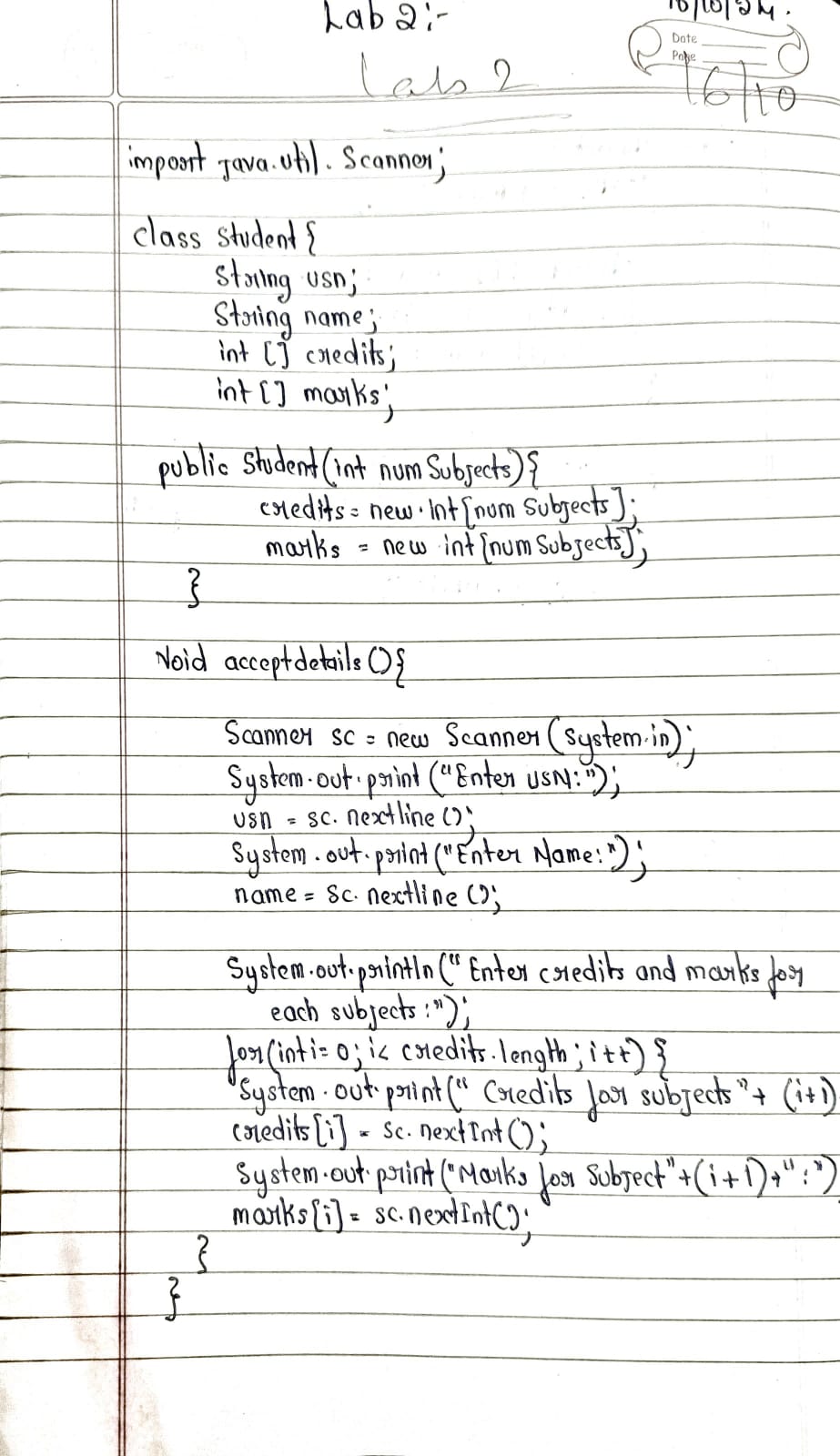
**Output:**

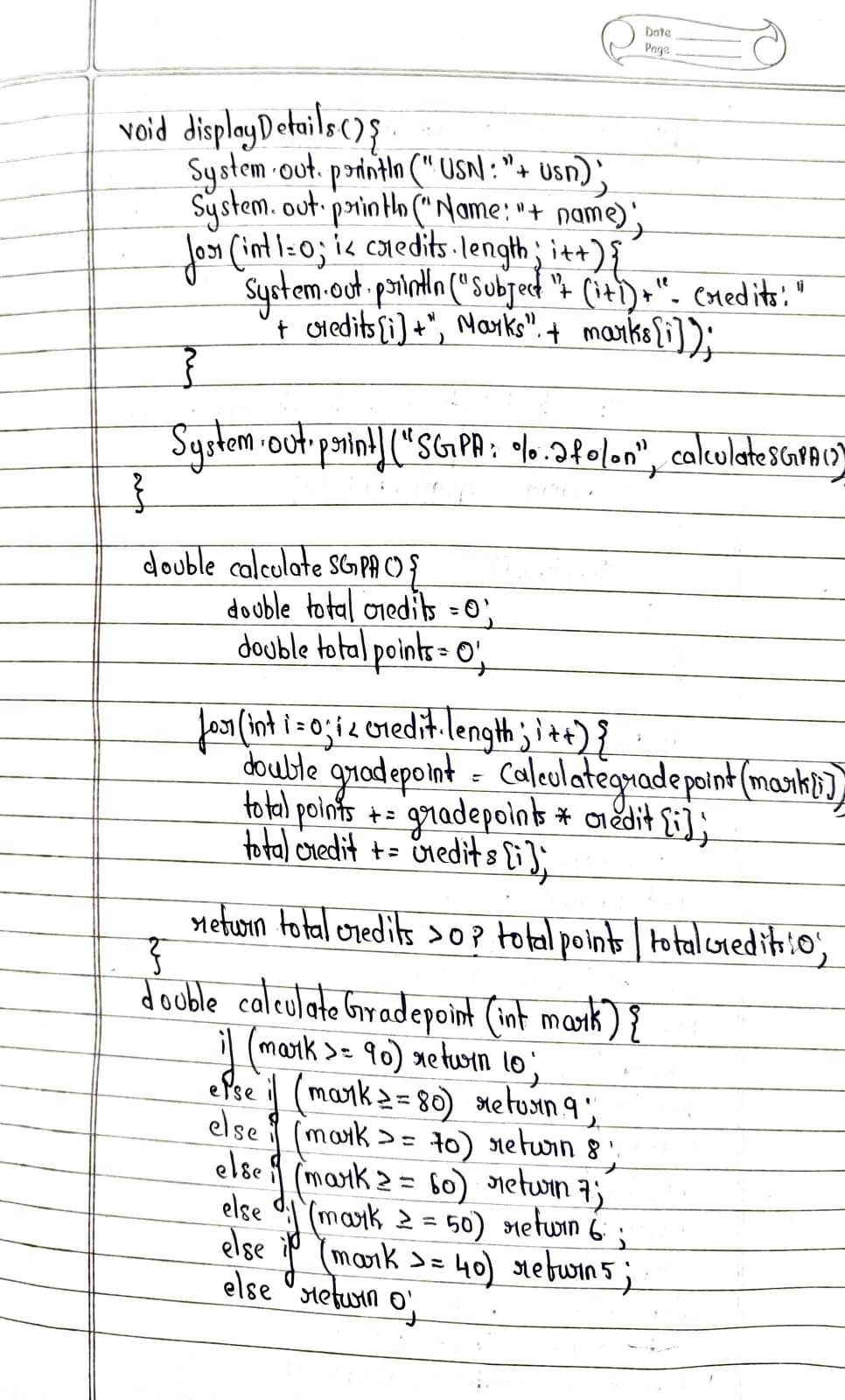
****

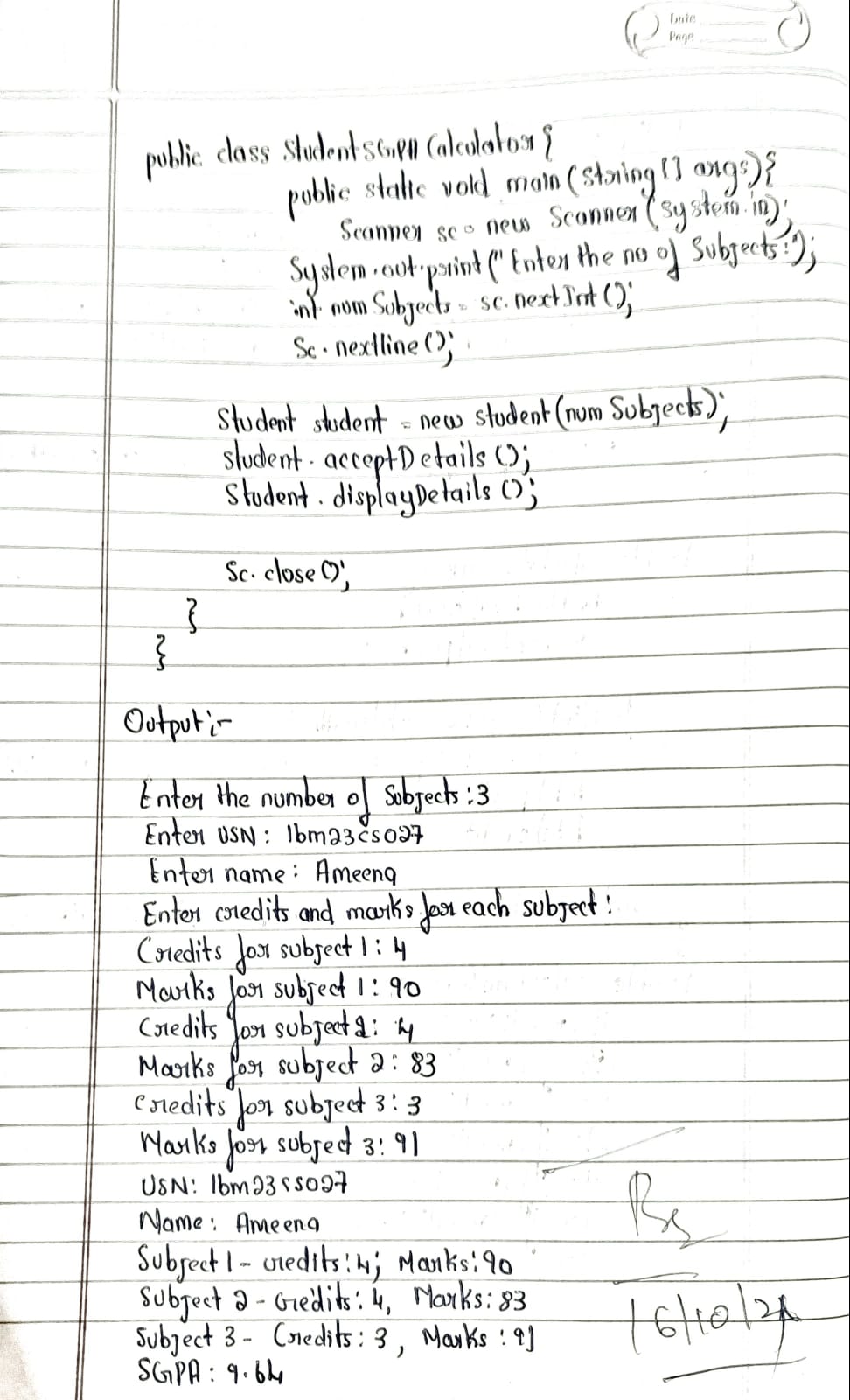
**<https://github.com/Ameena1BM23CS27/java-lab/commit/6ed0a38366980ad8e1173c9918e6a245ab470572>**

**Program 2**

**Implement SGPA Calculator**

**­­\_**

****

****

**Source Code**

import java.util.Scanner;

class Student {

String usn;

String name;

int[] credits;

int[] marks;

public Student(int numSubjects) {

credits = new int[numSubjects];

marks = new int[numSubjects];

}

void acceptDetails() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter USN: ");

usn = sc.nextLine();

System.out.print("Enter Name: ");

name = sc.nextLine();

System.out.println("Enter credits and marks for each subject:");

for (int i = 0; i < credits.length; i++) {

System.out.print("Credits for subject " + (i + 1) + ": ");

credits[i] = sc.nextInt();

System.out.print("Marks for subject " + (i + 1) + ": ");

marks[i] = sc.nextInt();

}

}

void displayDetails() {

System.out.println("USN: " + usn);

System.out.println("Name: " + name);

for (int i = 0; i < credits.length; i++) {

System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);

}

System.out.printf("SGPA: %.2f%n", calculateSGPA());

}

double calculateSGPA() {

double totalCredits = 0;

double totalPoints = 0;

for (int i = 0; i < credits.length; i++) {

double gradePoint = calculateGradePoint(marks[i]);

totalPoints += gradePoint \* credits[i];

totalCredits += credits[i];

}

return totalCredits > 0 ? totalPoints / totalCredits : 0;

}

double calculateGradePoint(int mark) {

if (mark >= 90) return 10;

else if (mark >= 80) return 9;

else if (mark >= 70) return 8;

else if (mark >= 60) return 7;

else if (mark >= 50) return 6;

else if (mark >= 40) return 5;

else return 0;

}

}

public class StudentSGPACalculator {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of subjects: ");

int numSubjects = sc.nextInt();

sc.nextLine();

Student student = new Student(numSubjects);

student.acceptDetails();

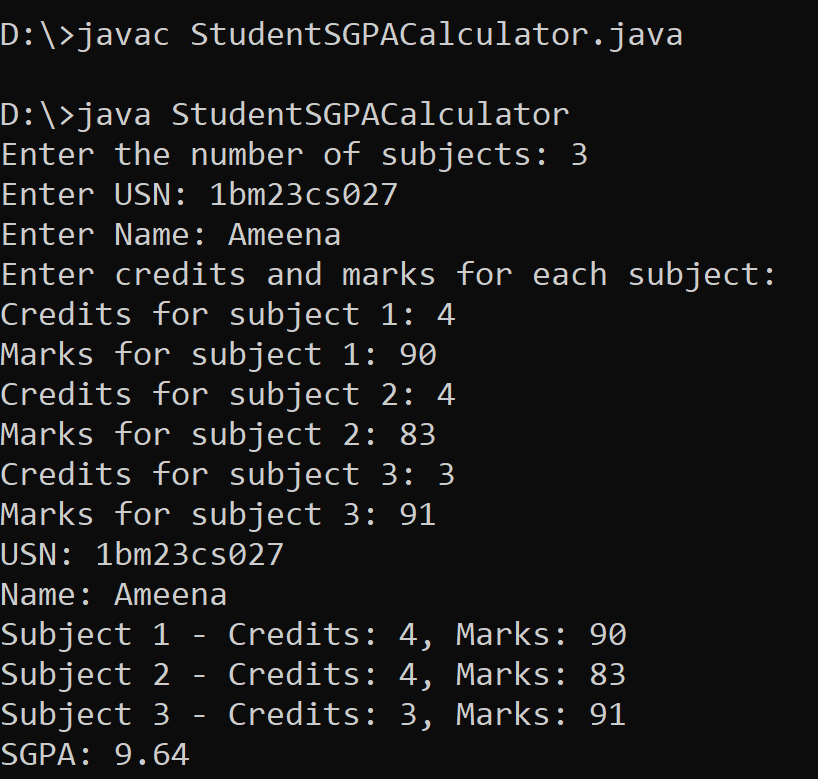
student.displayDetails();

sc.close();

}

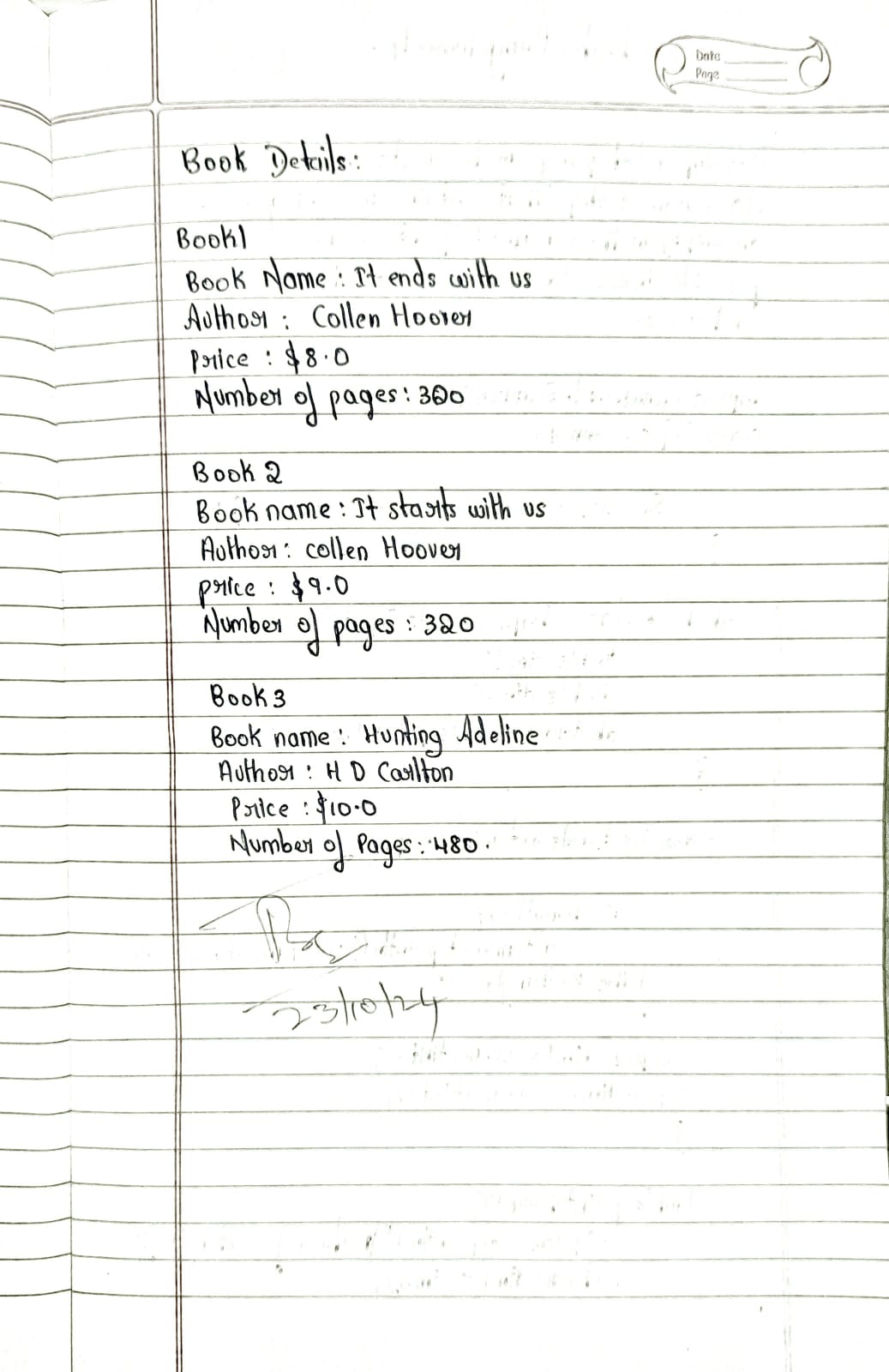
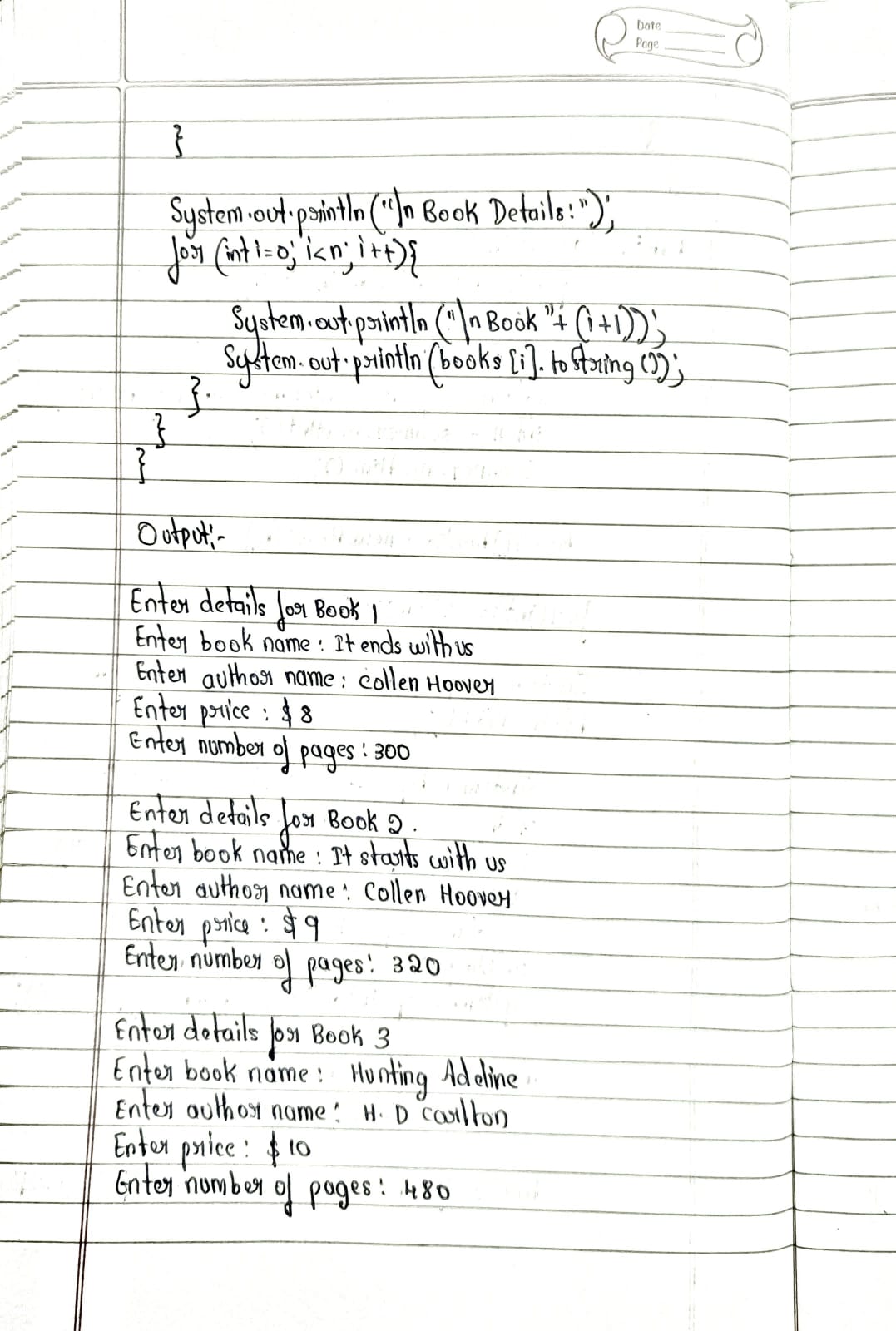
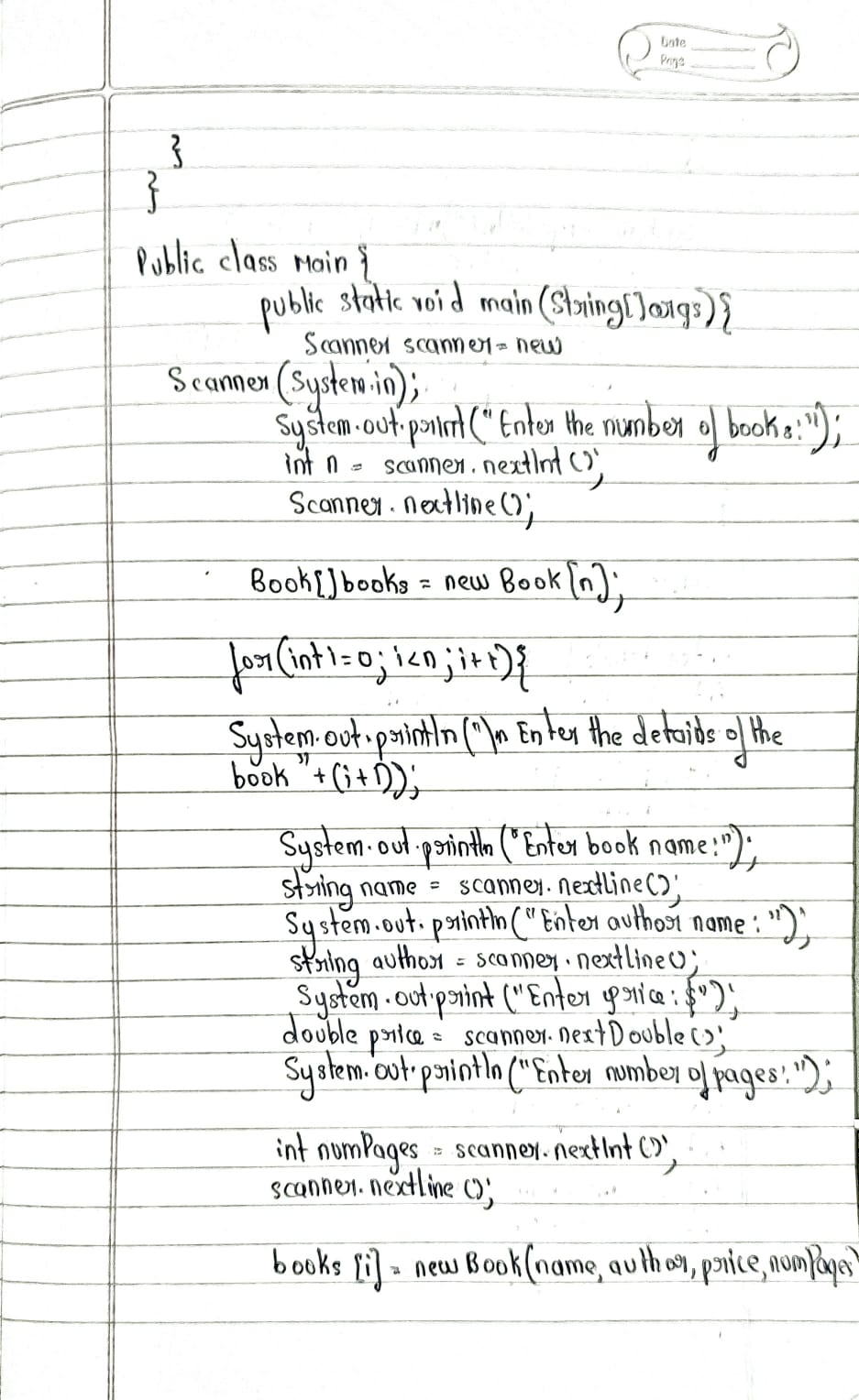
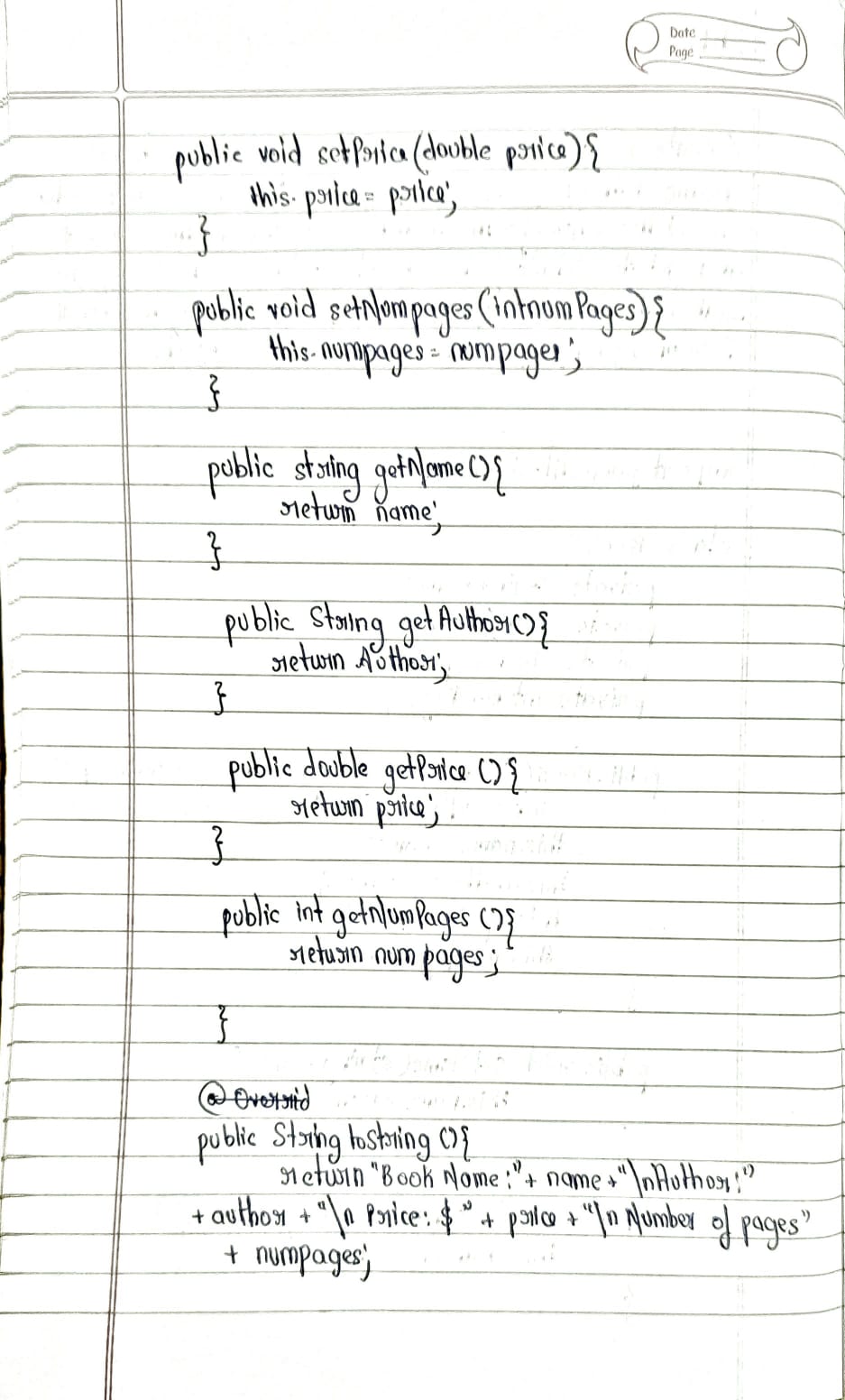
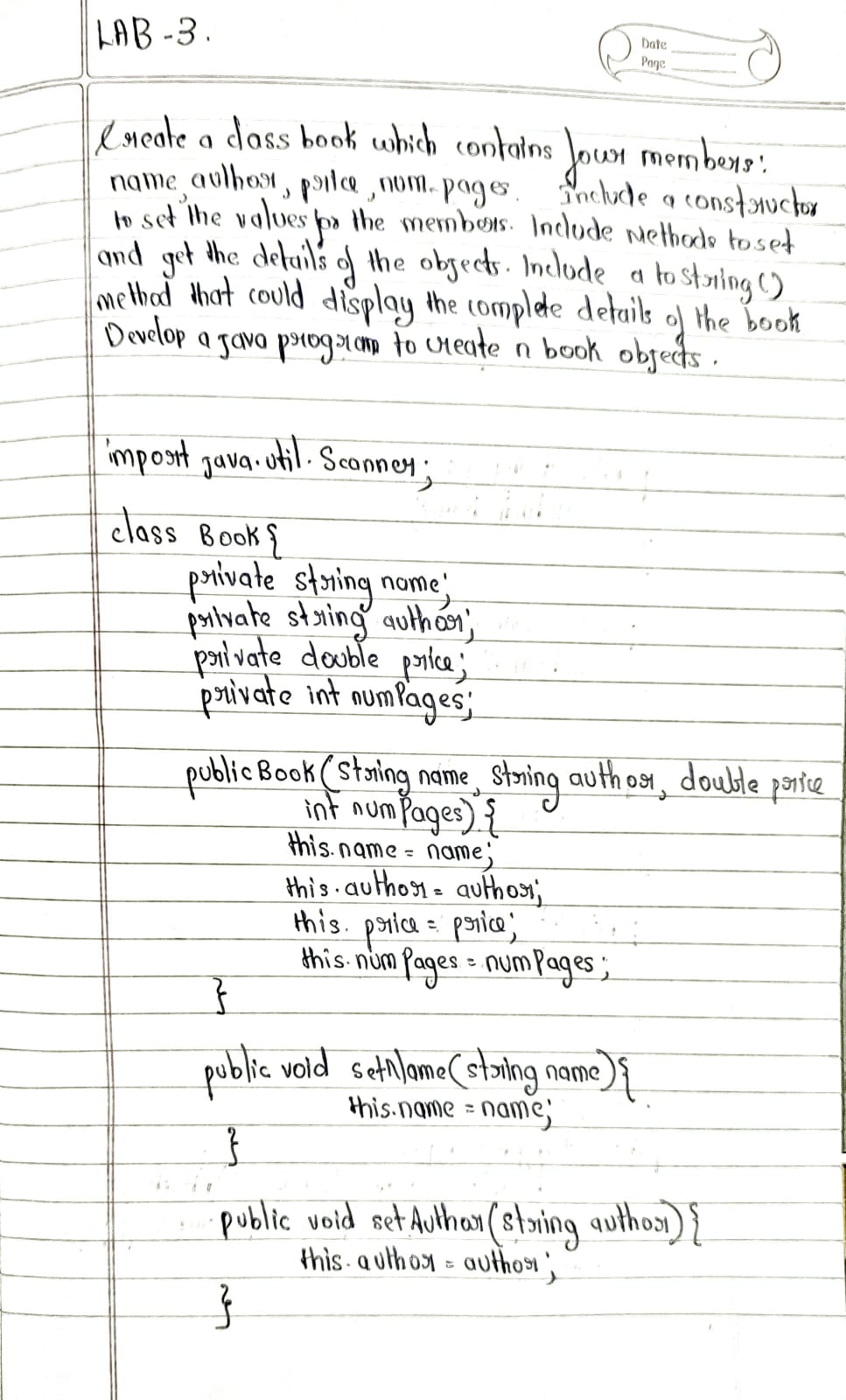
}

**Output:**

****

[**https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%203**](https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%203)

**Program 3**

**Create Objects For Book**

**Sorce 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++) {

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);

}

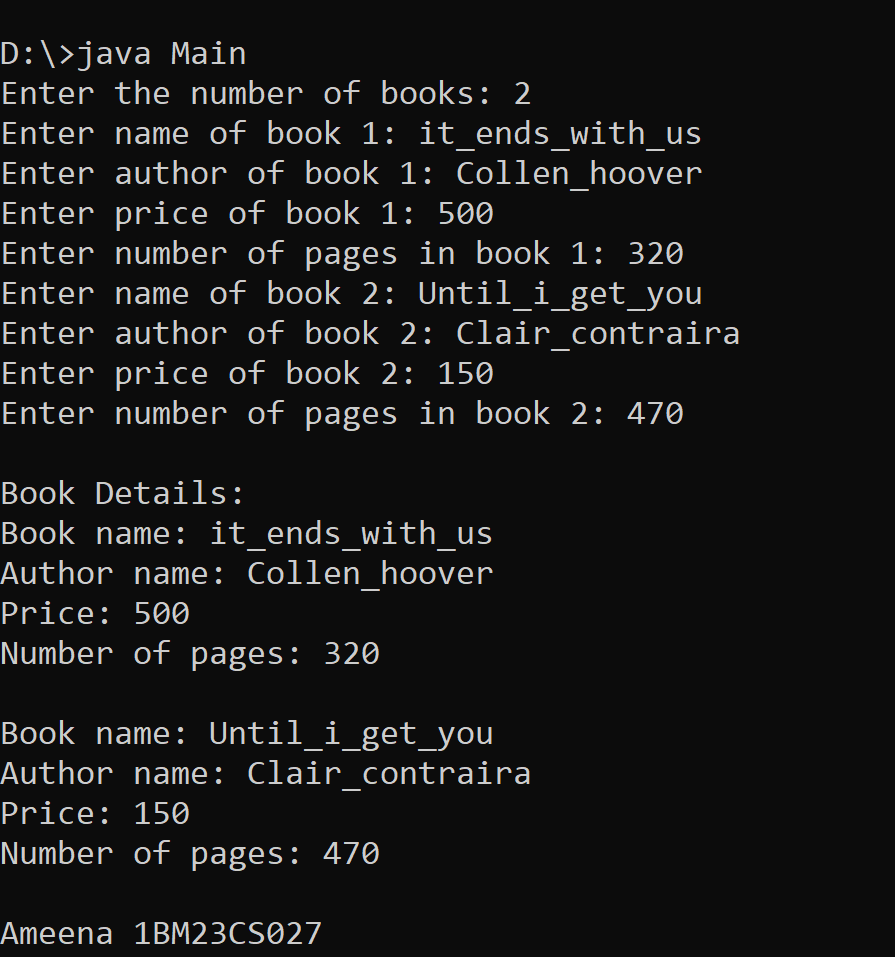
s.close();

System.out.println("Ameena 1BM23CS027");

}

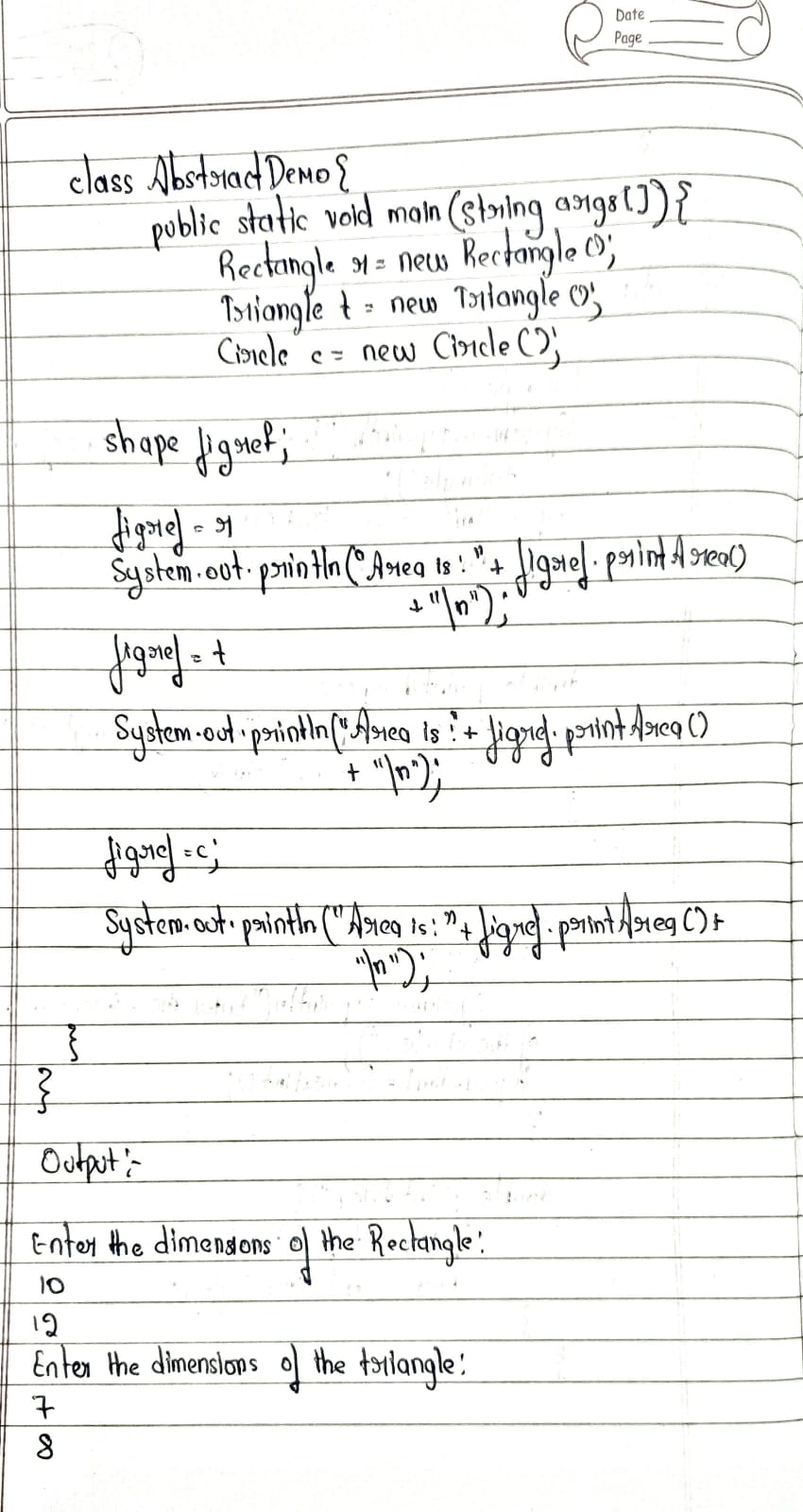
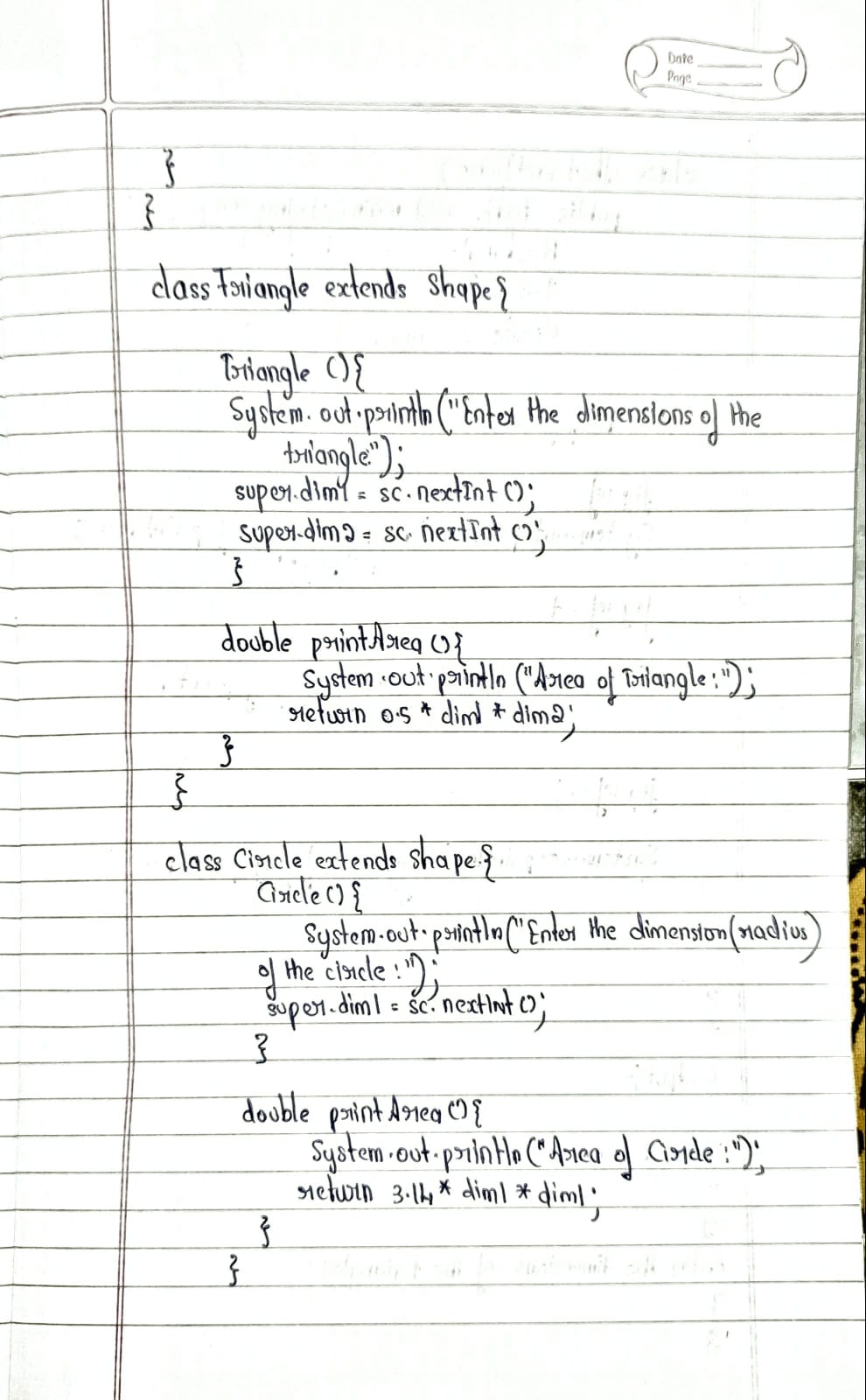
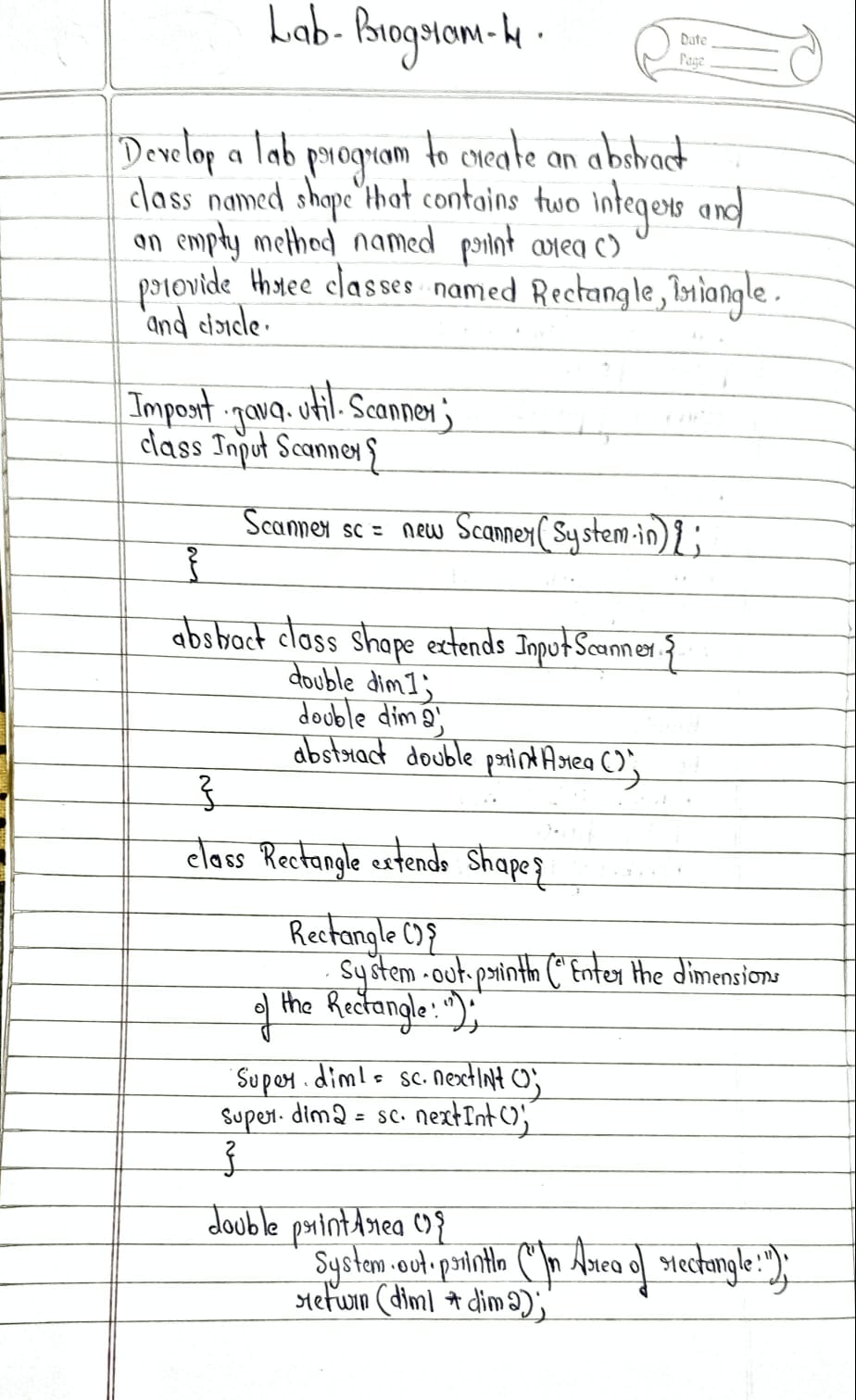
}

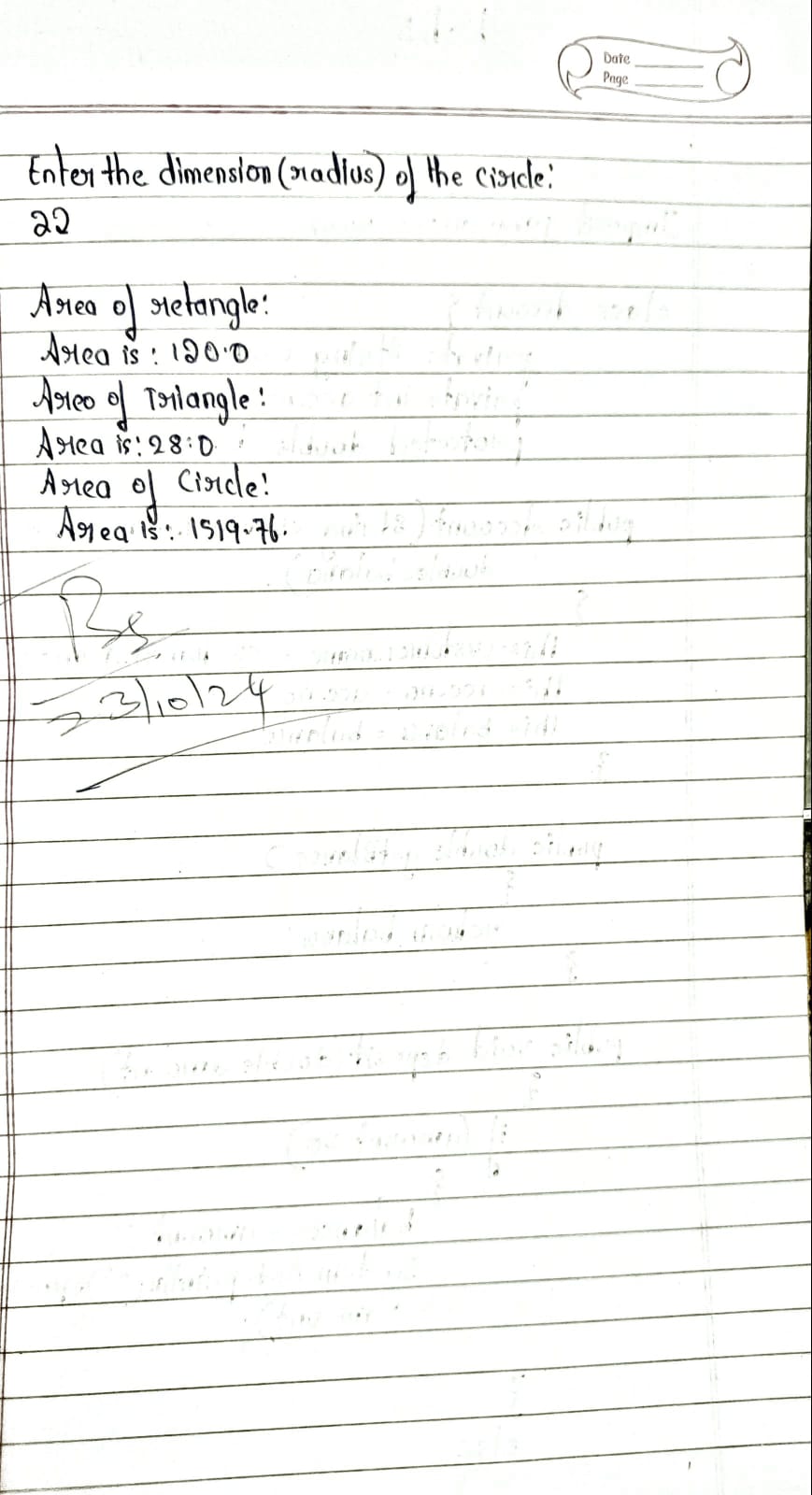
**Output:**

****

[**https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%204**](https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%204)

**Program 4**

**Implement Abstract Class**

****

**Source 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();

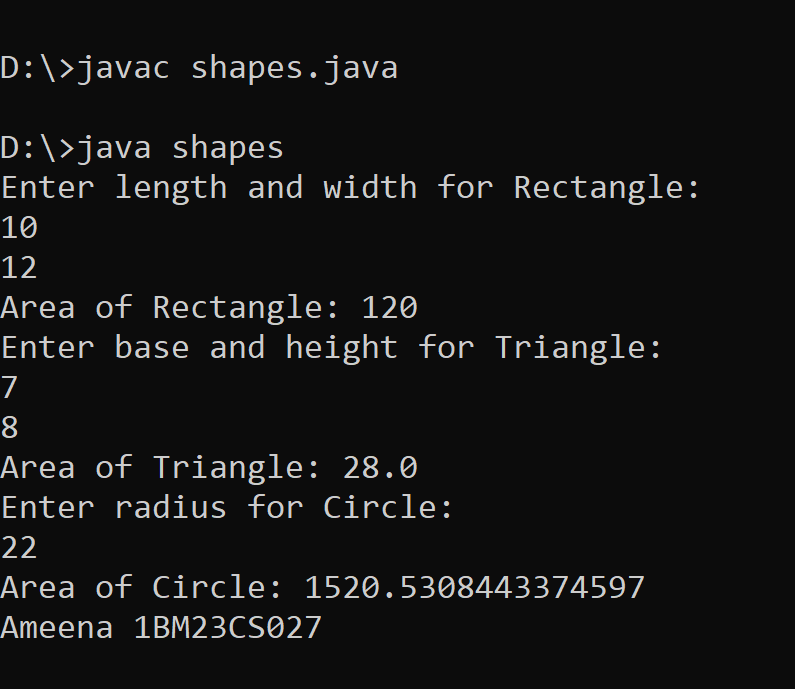
in.close();

System.out.println("Ameena 1BM23CS027");

}

}

**Output:**

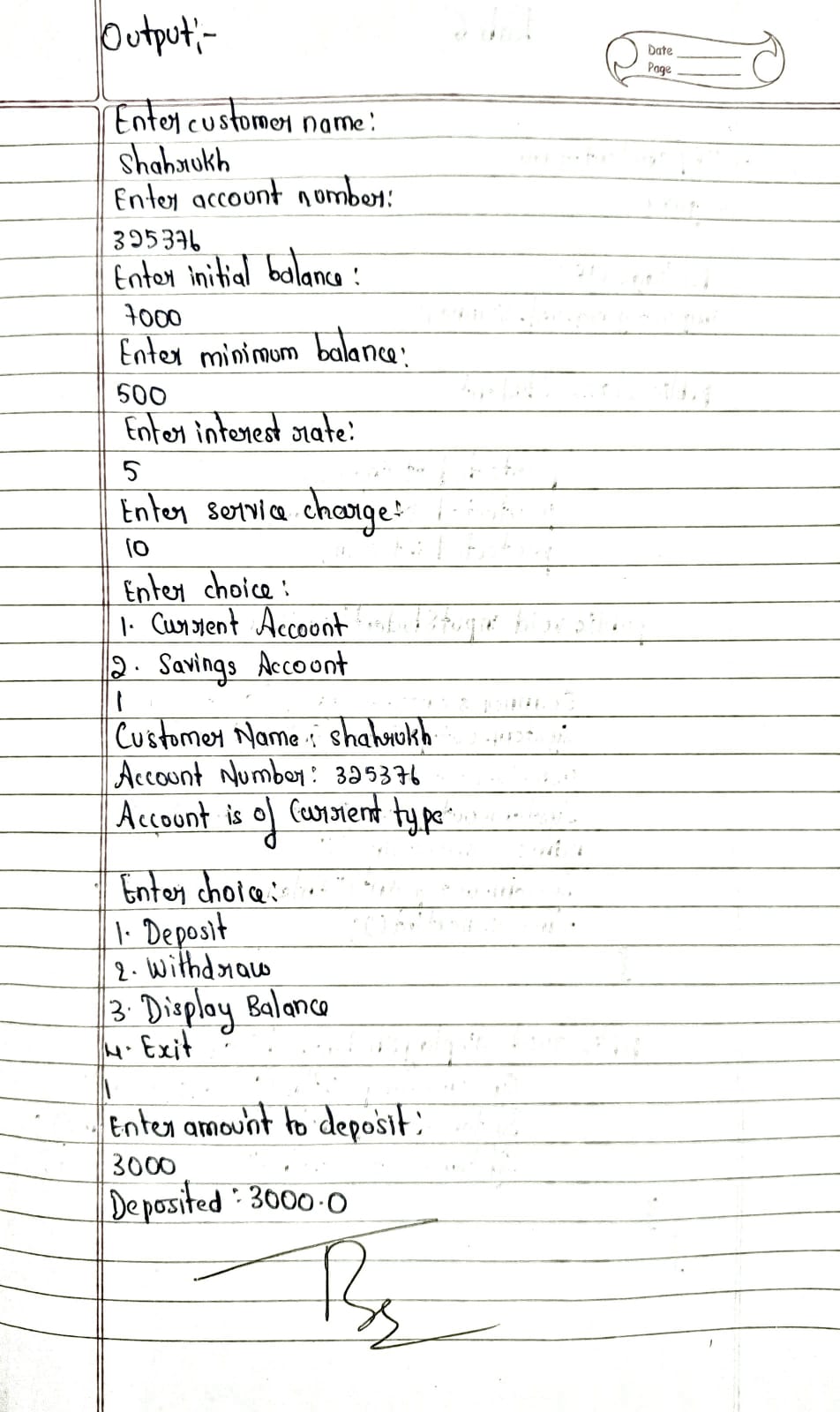
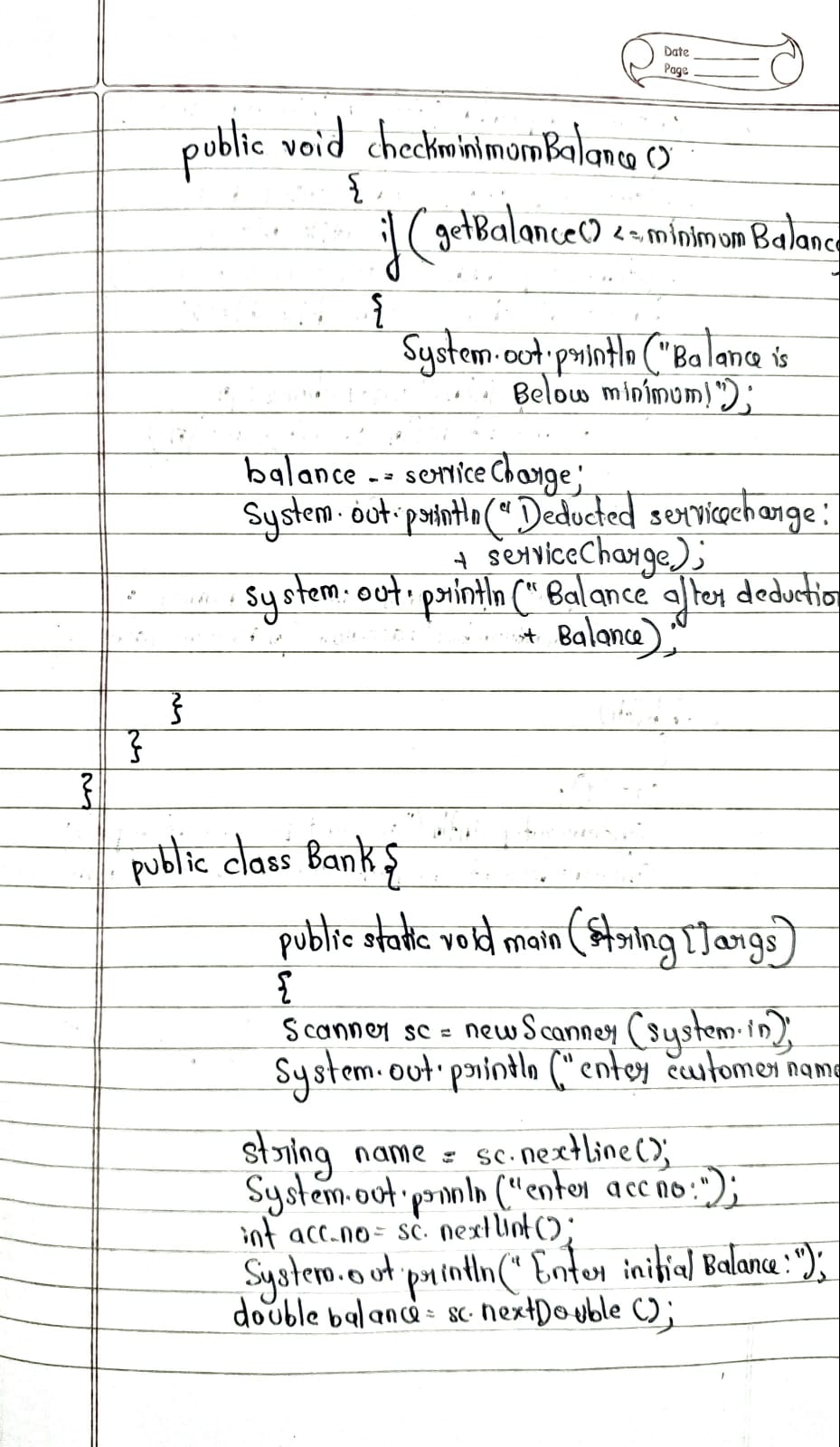
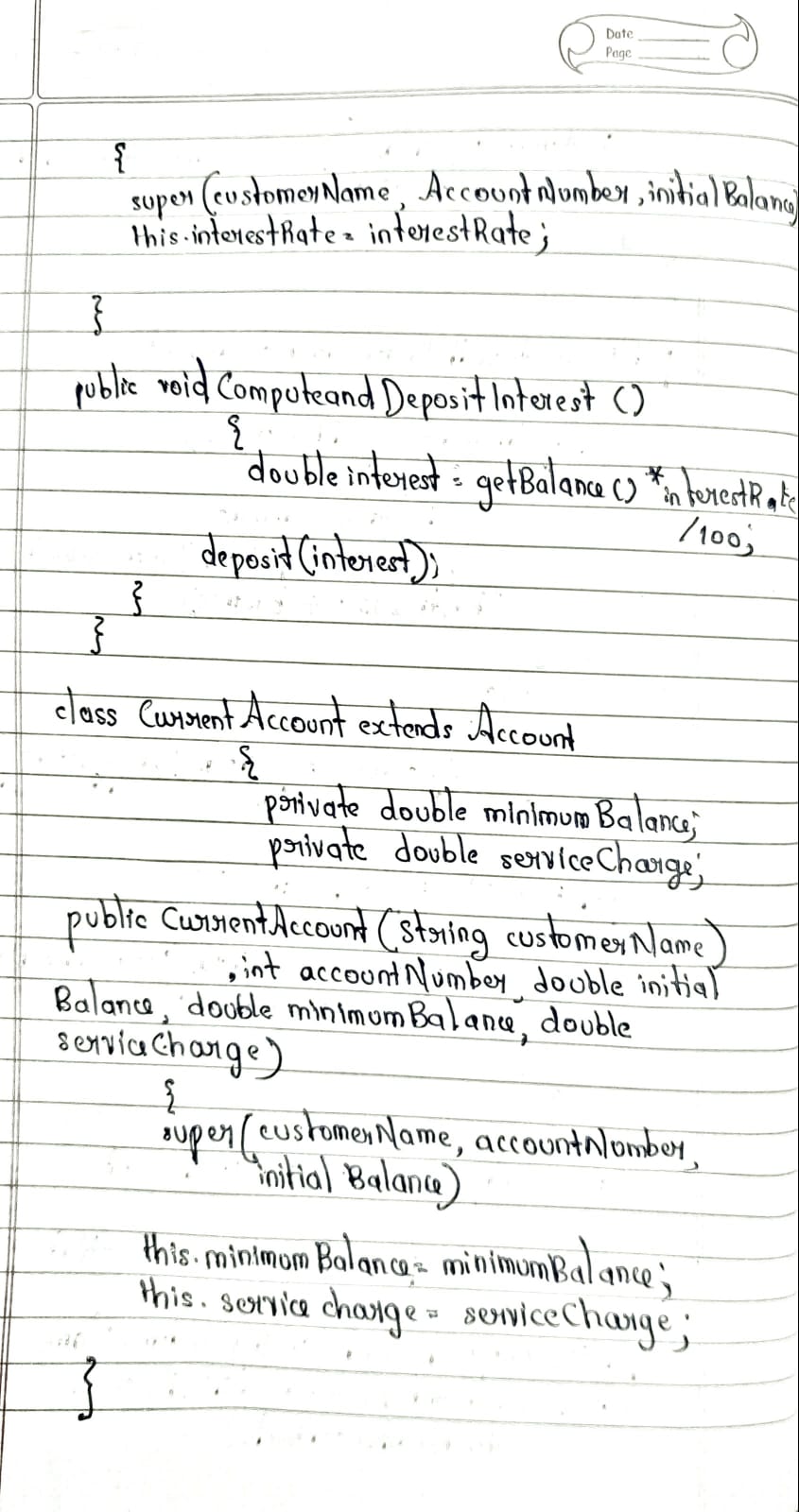
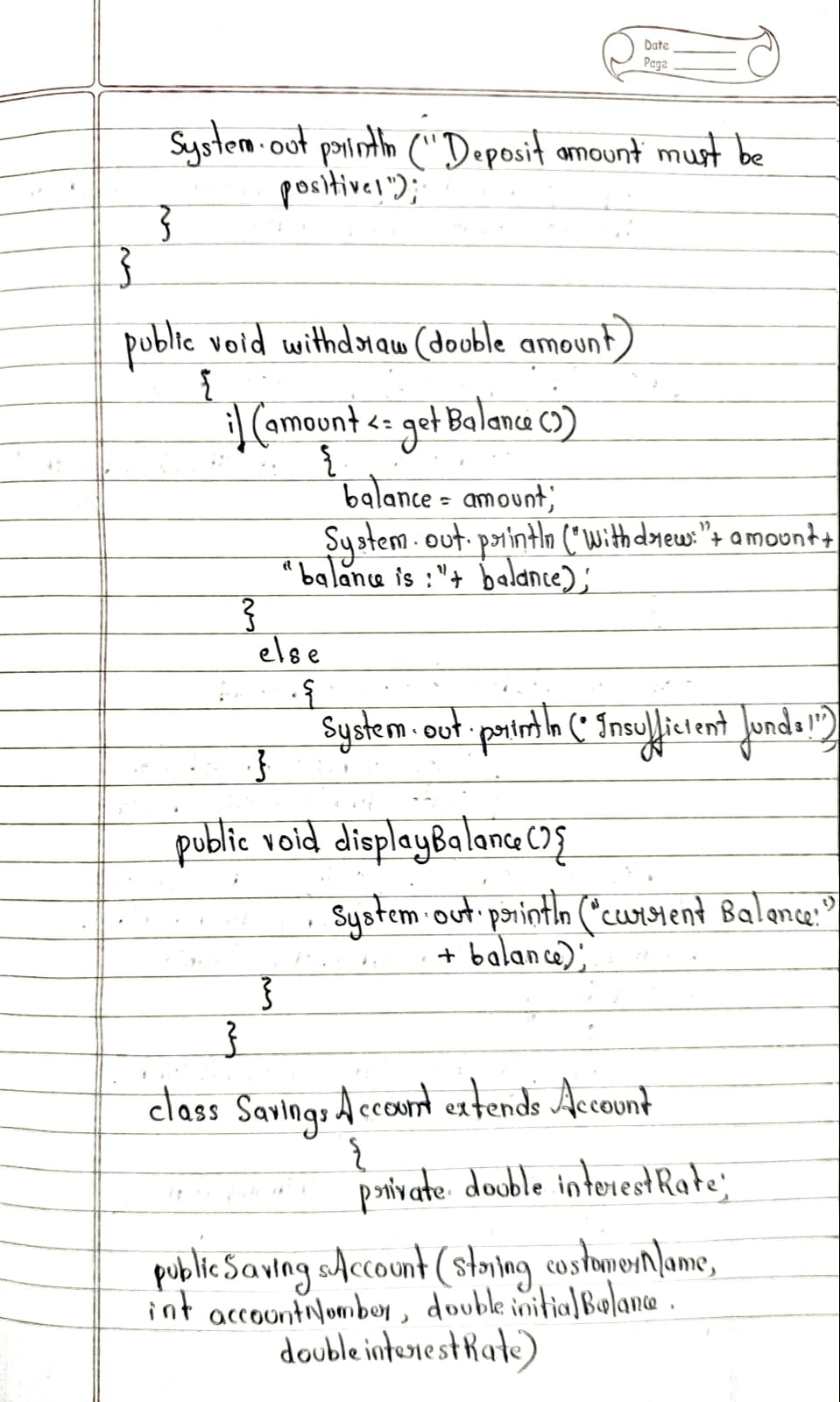
****

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%205th%20laba>

**Program 5**

**Bank Account Management**

****

****

**Source 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 the minimum!");

balance -= serviceCharge;

System.out.println("Deducted service charge: " + serviceCharge);

System.out.println("Balance after deduction: " + 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 account number:");

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:\n1. Current Account\n2. Savings Account");

int ch = sc.nextInt();

System.out.println("Customer Name: " + name);

System.out.println("Account Number: " + acc\_no);

switch (ch) {

case 1: // Current Account

System.out.println("Account is of Current type.");

CurrentAccount ca = new CurrentAccount(name, acc\_no, balance, minimum\_balance, service\_charge);

while (true) {

System.out.println("Enter choice:\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");

int c = sc.nextInt();

if (c == 1) {

System.out.println("Enter amount to deposit:");

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.checkMinimumBalance();

ca.displayBalance();

} else {

System.out.println("Exiting Current Account...");

break;

}

}

break;

case 2: // Savings Account

System.out.println("Account is of Savings type.");

SavingsAccount sa = new SavingsAccount(name, acc\_no, balance, interest\_rate);

while (true) {

System.out.println("Enter choice:\n1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");

int c = sc.nextInt();

if (c == 1) {

System.out.println("Enter amount to deposit:");

double amt = sc.nextDouble();

sa.deposit(amt);

} else if (c == 2) {

System.out.println("Enter amount to withdraw:");

double amt = sc.nextDouble();

sa.withdraw(amt);

} else if (c == 3) {

sa.computeAndDepositInterest();

sa.displayBalance();

} else {

System.out.println("Exiting Savings Account...");

break;

}

}

break;

default:

System.out.println("Invalid choice! Exiting...");

}

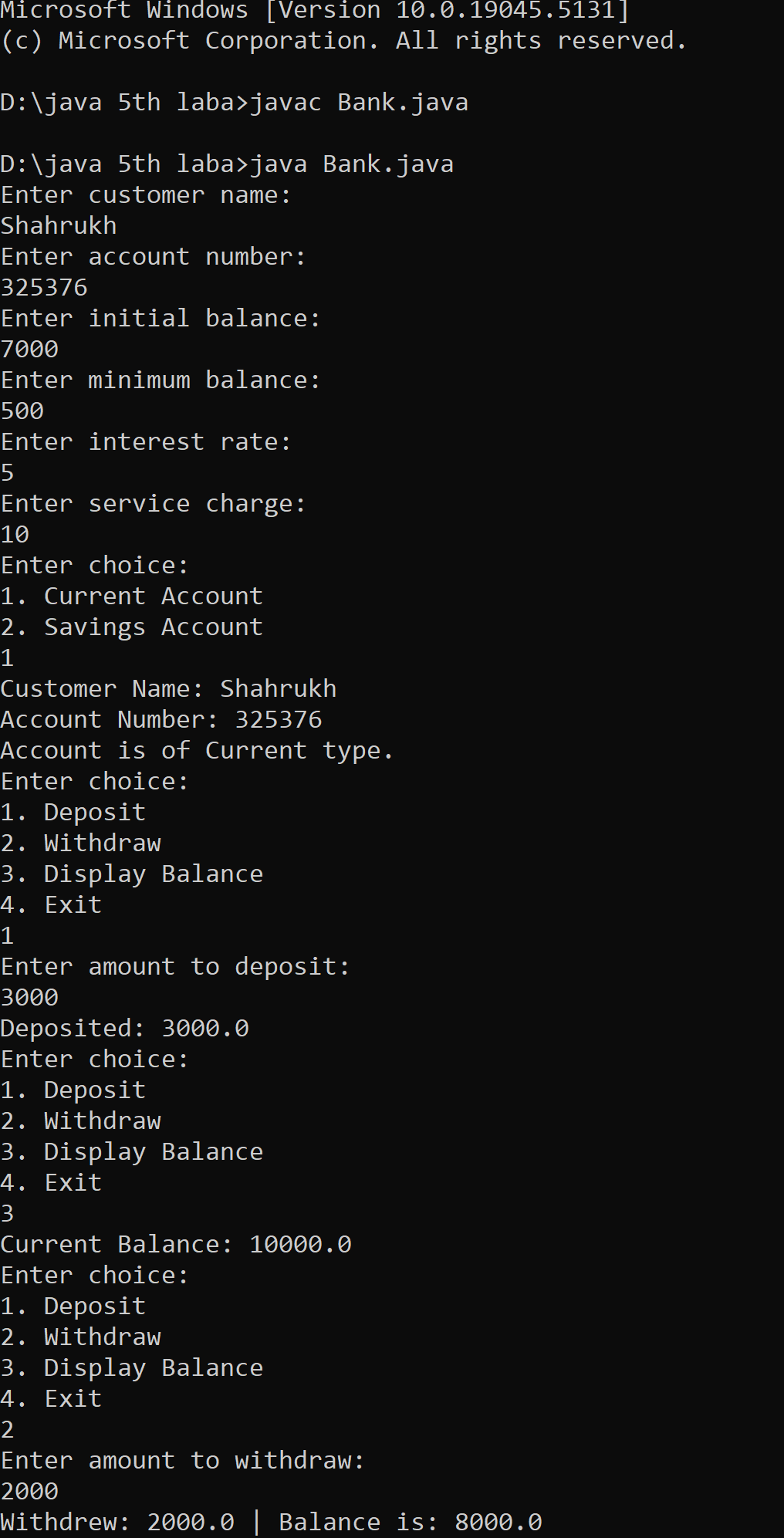
System.out.println("Name: Ameena Yasmeen\nUSN: 1BM23CS027");

sc.close();

}

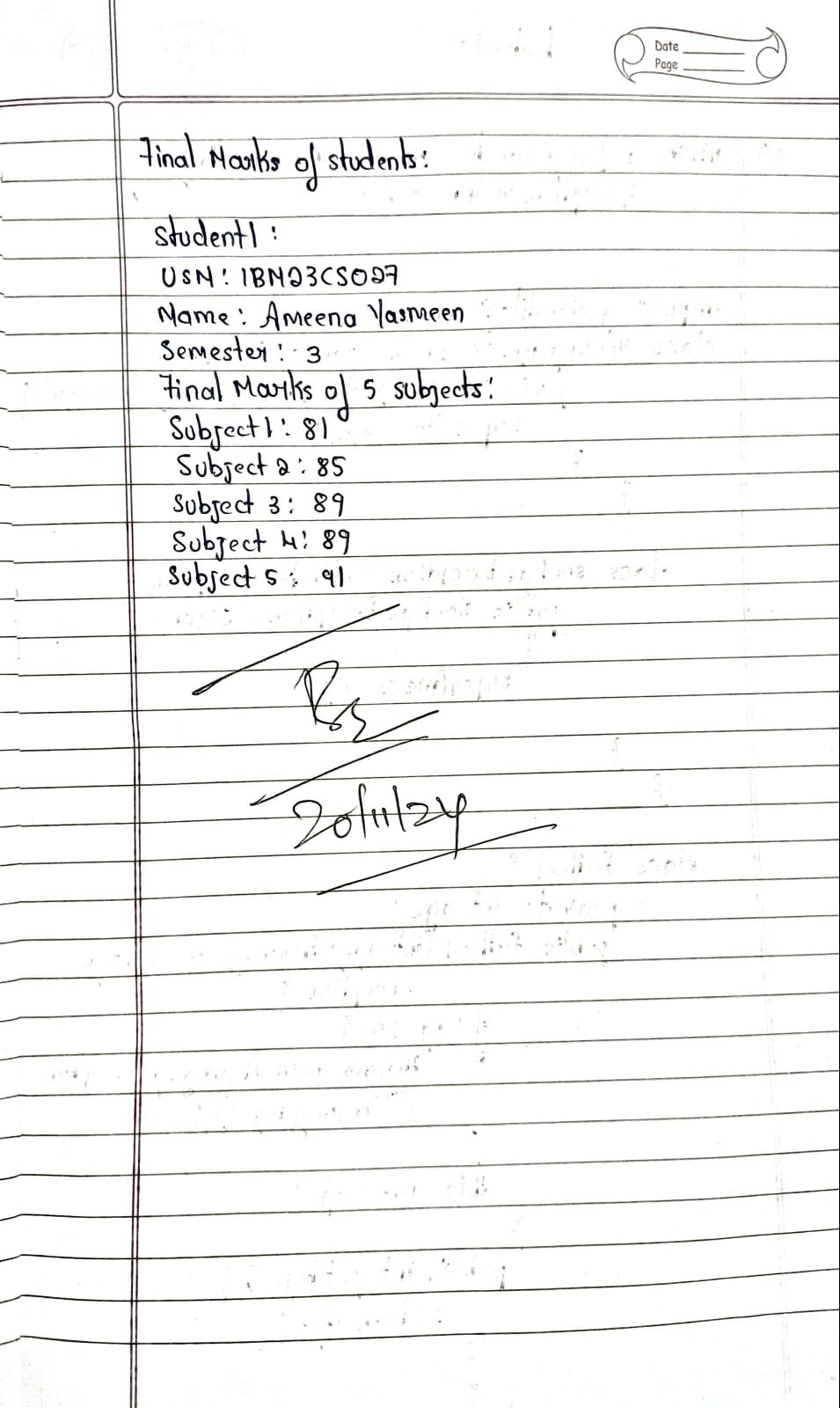
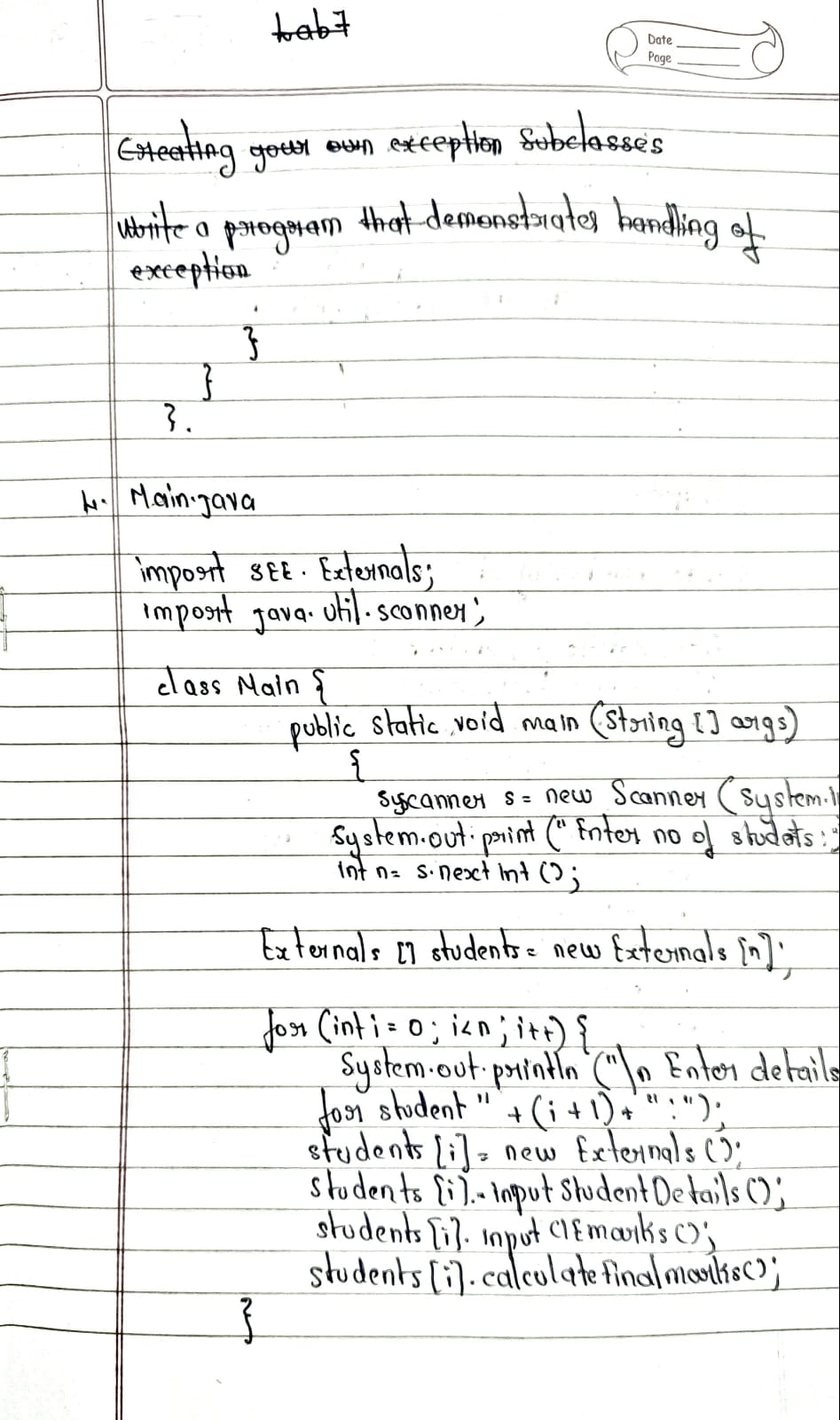
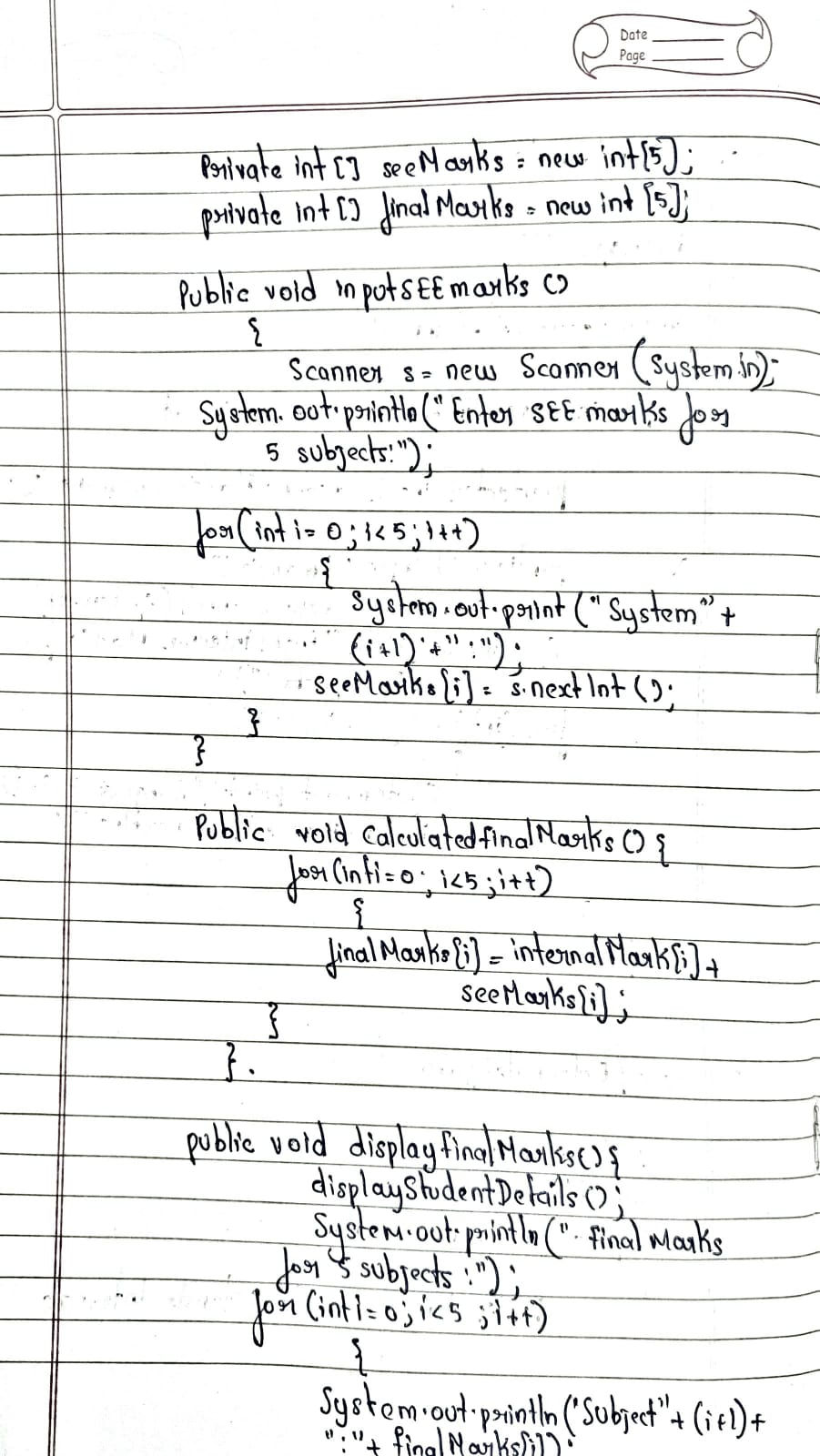
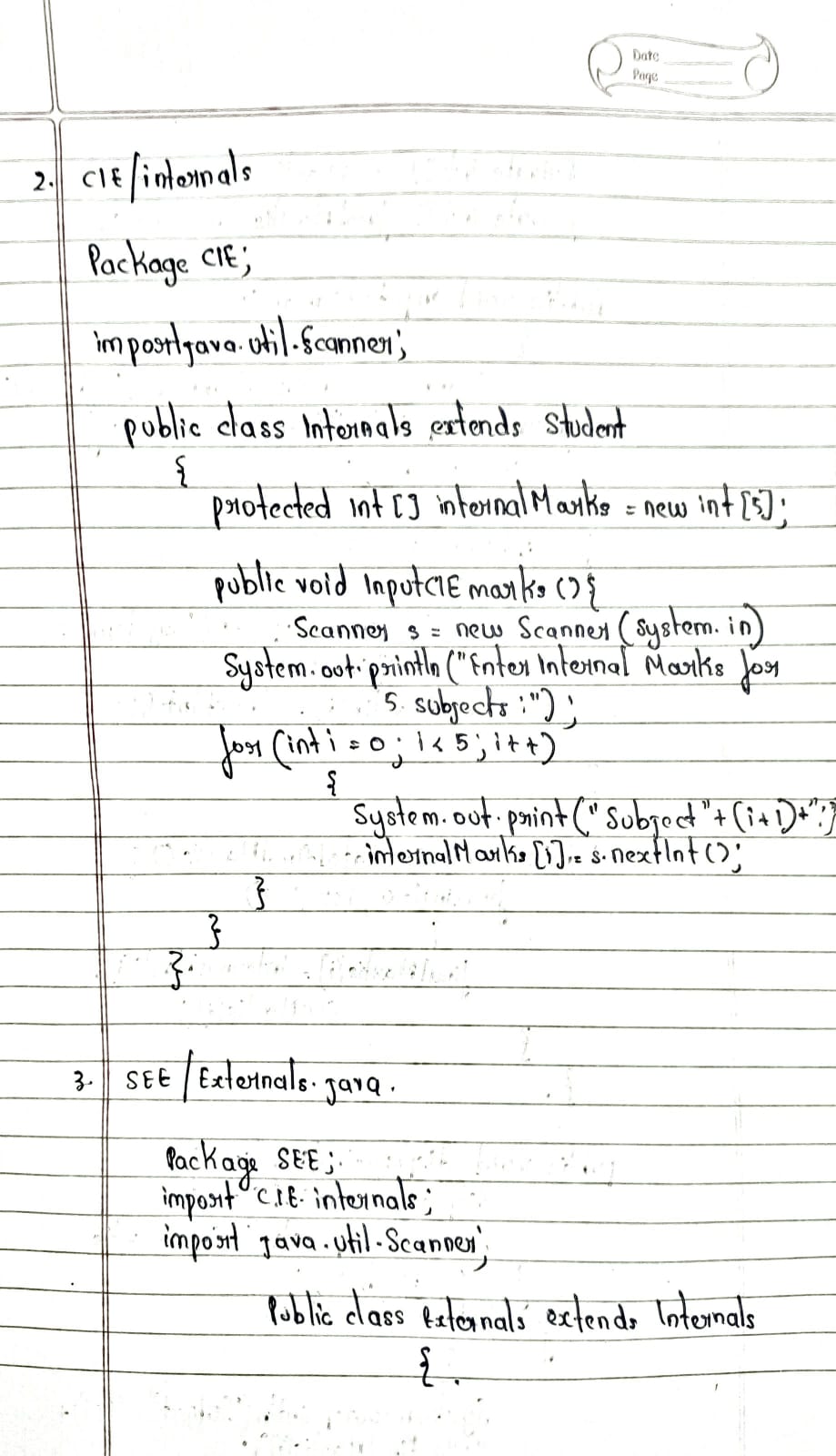
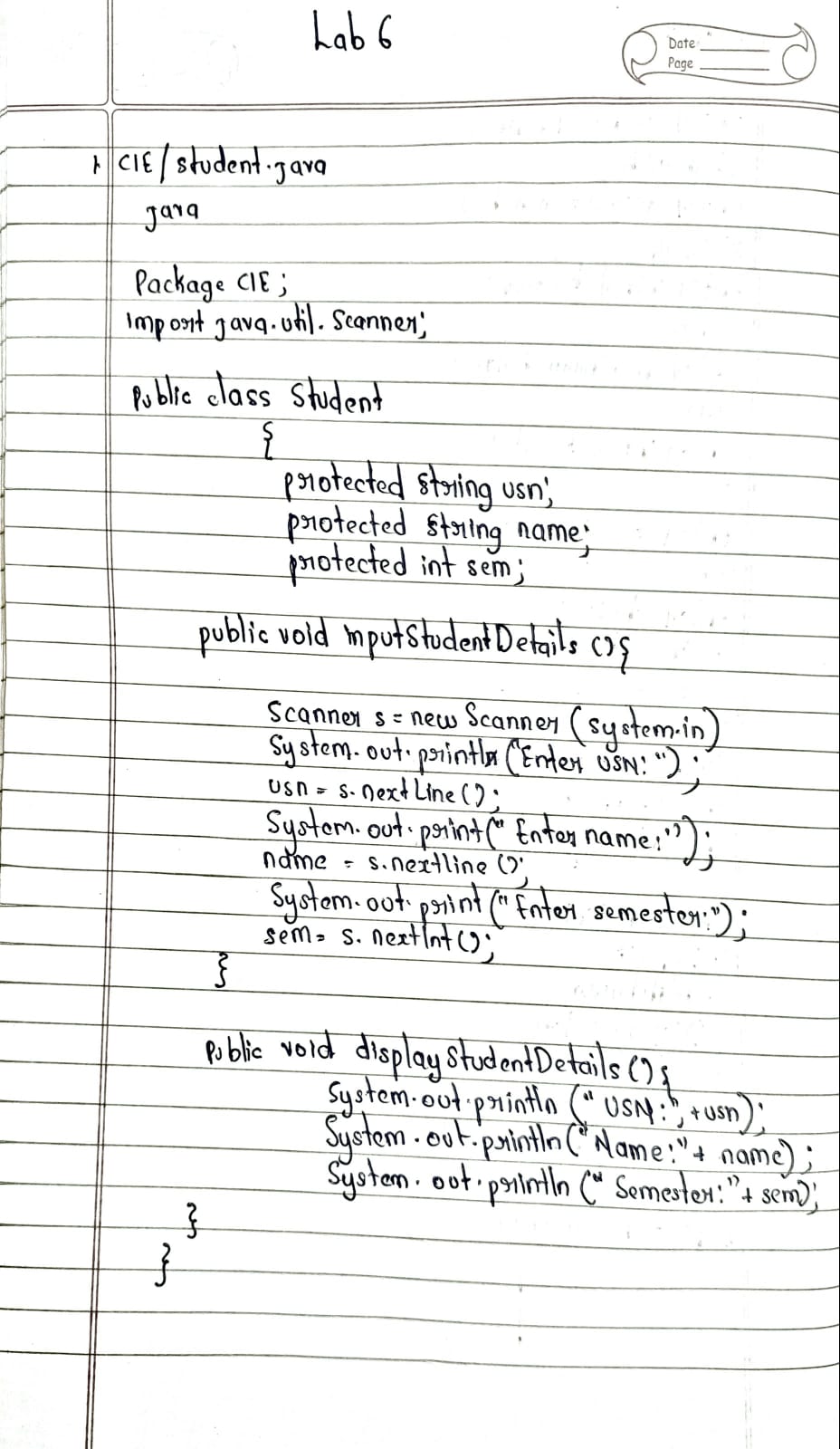
}

**Output**:



<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%206>

**Program 6**

**Implement Packages**

**Source Code:**

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);

}

}

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();

}

}

}

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]);

}

}

}

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();

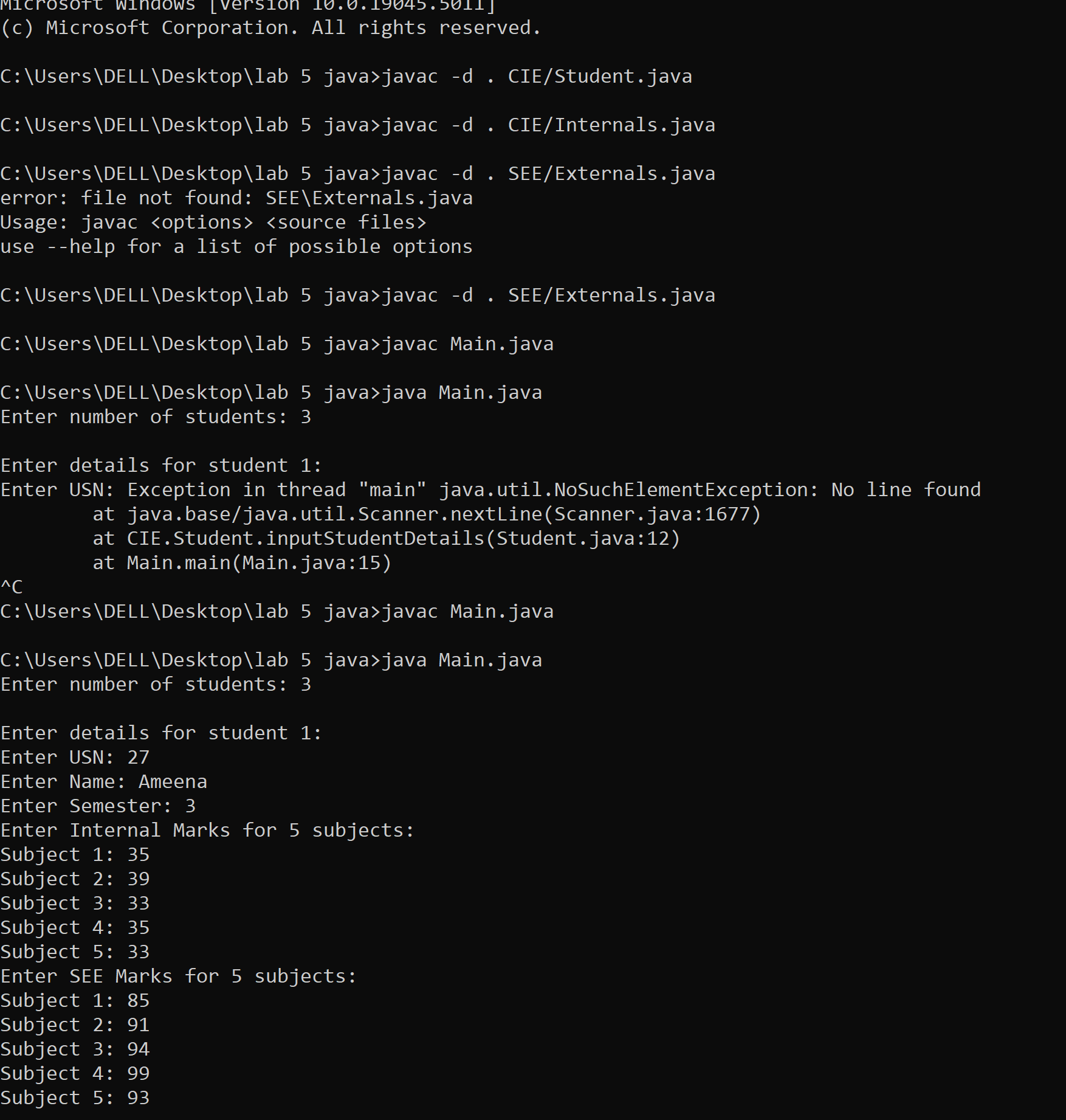
}

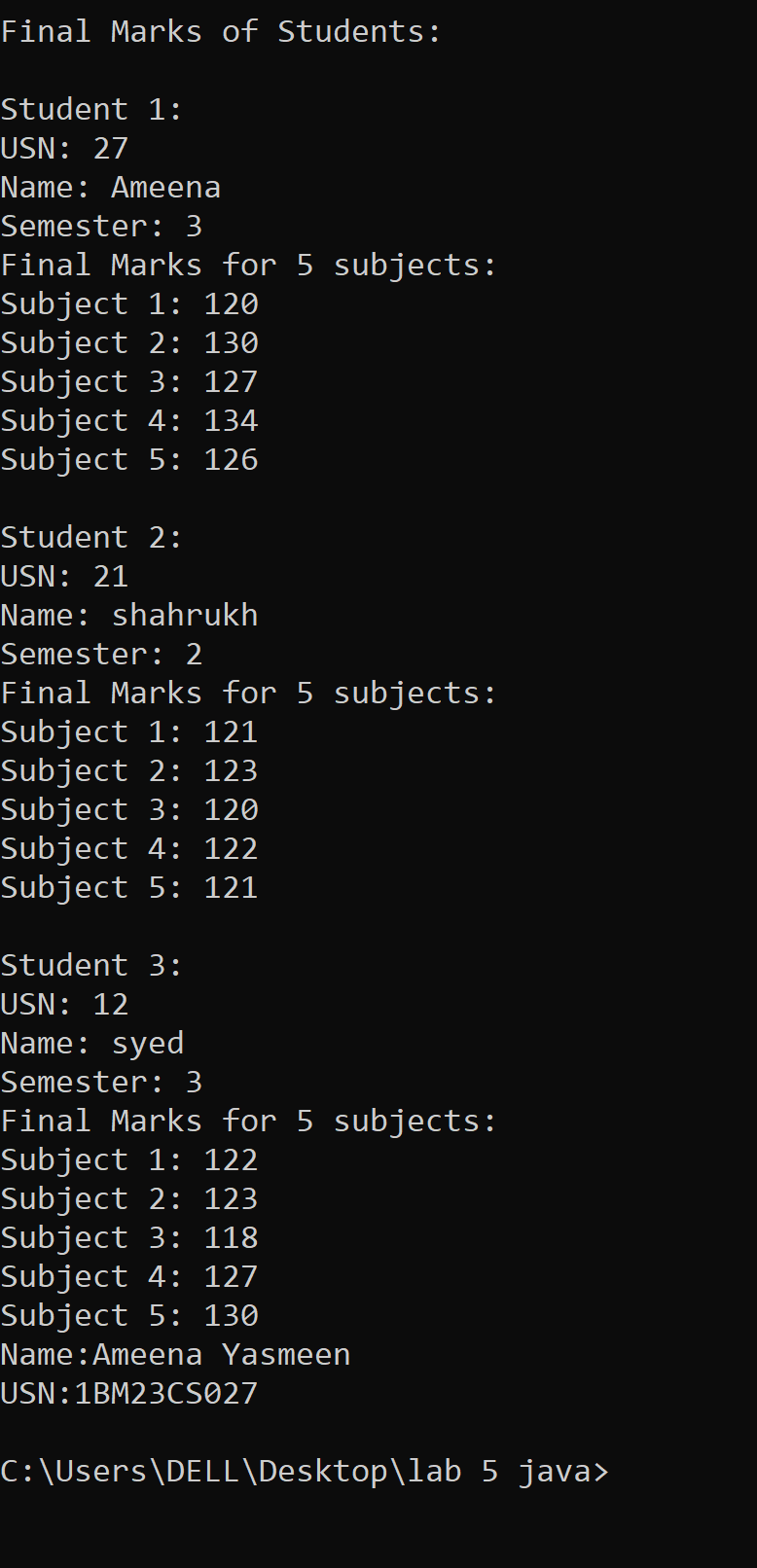
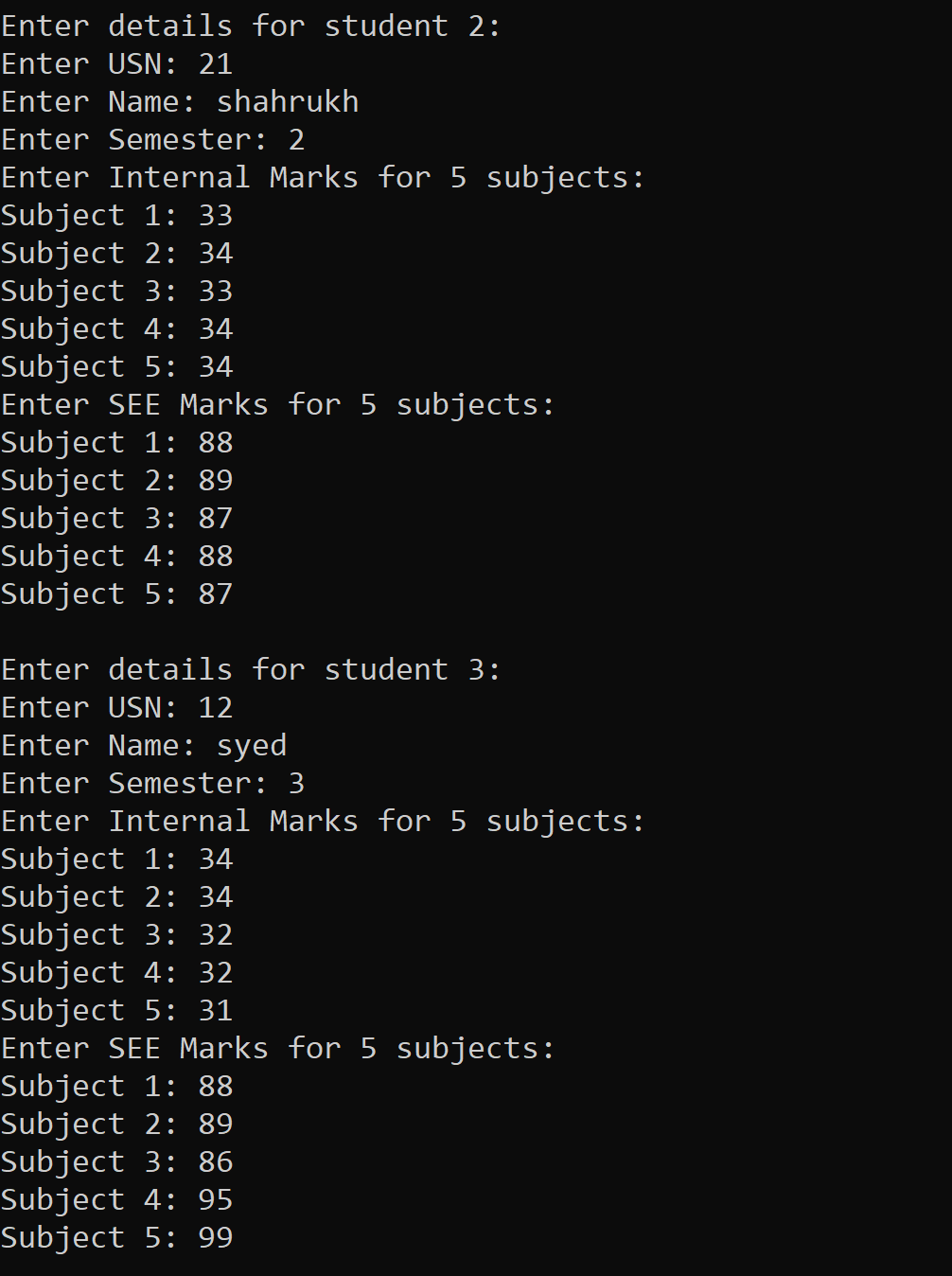
System.out.println("Name:Ameena Yasmeen\nUSN:1BM23CS027");

}

}

**Output:**

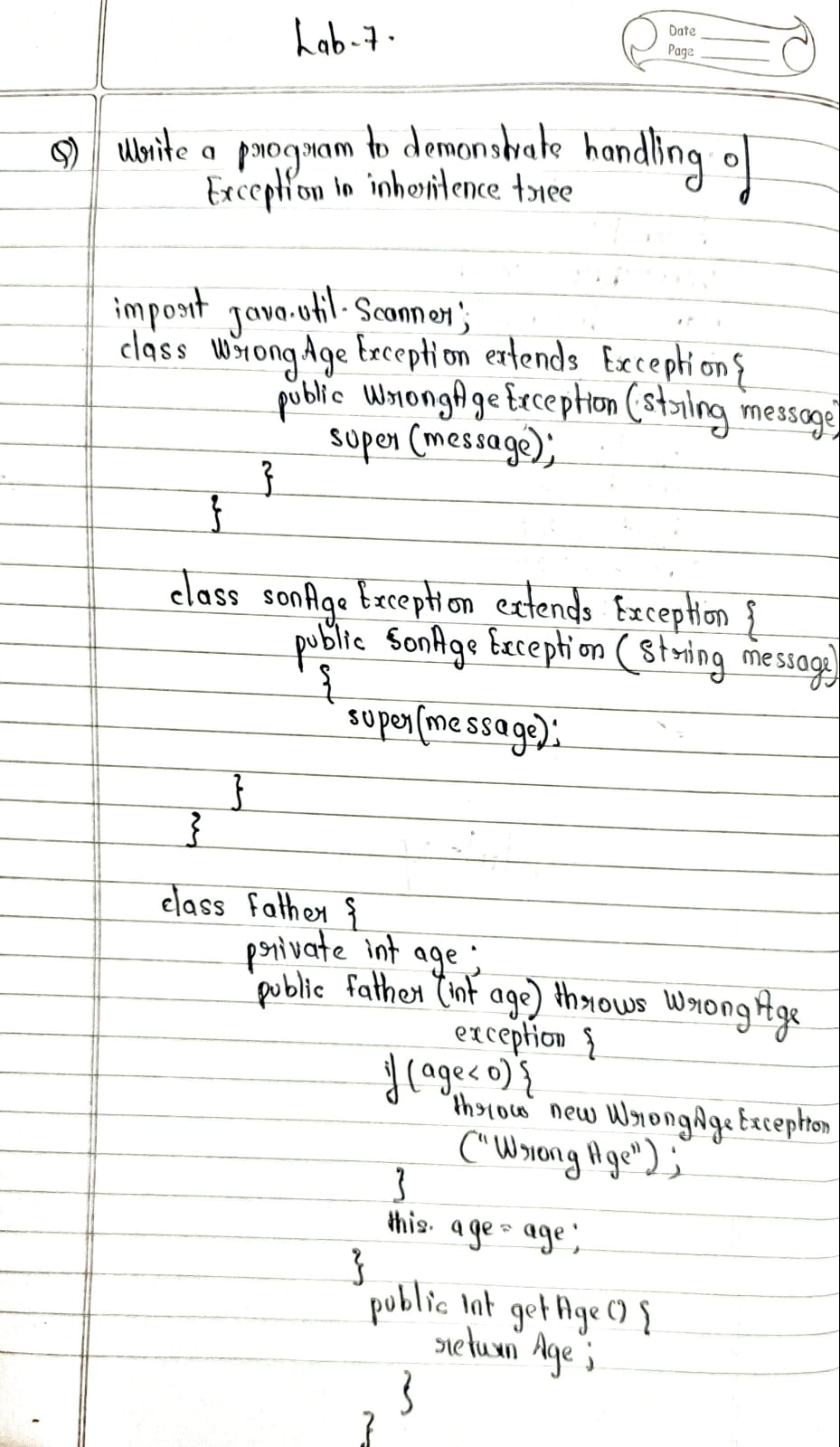
****

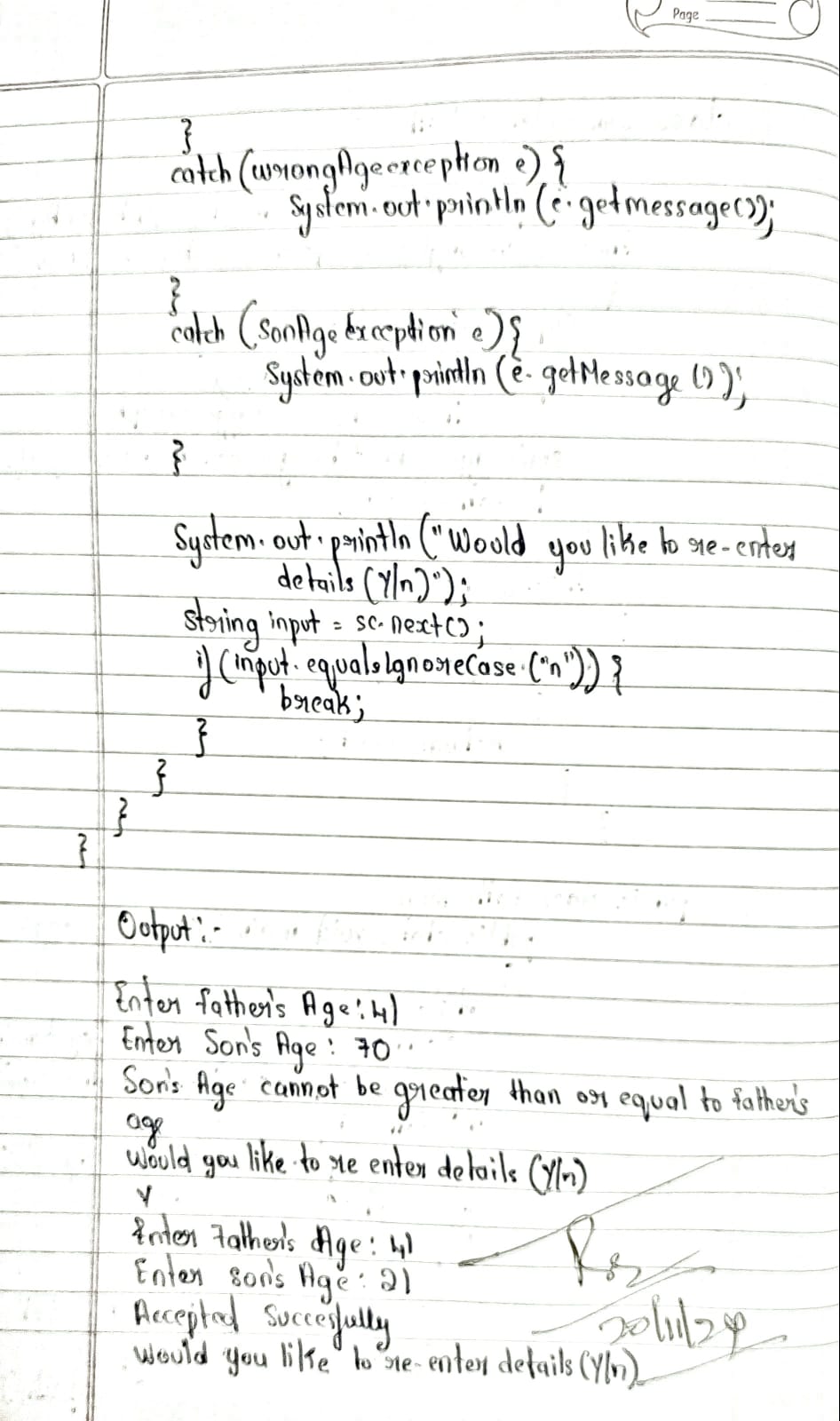
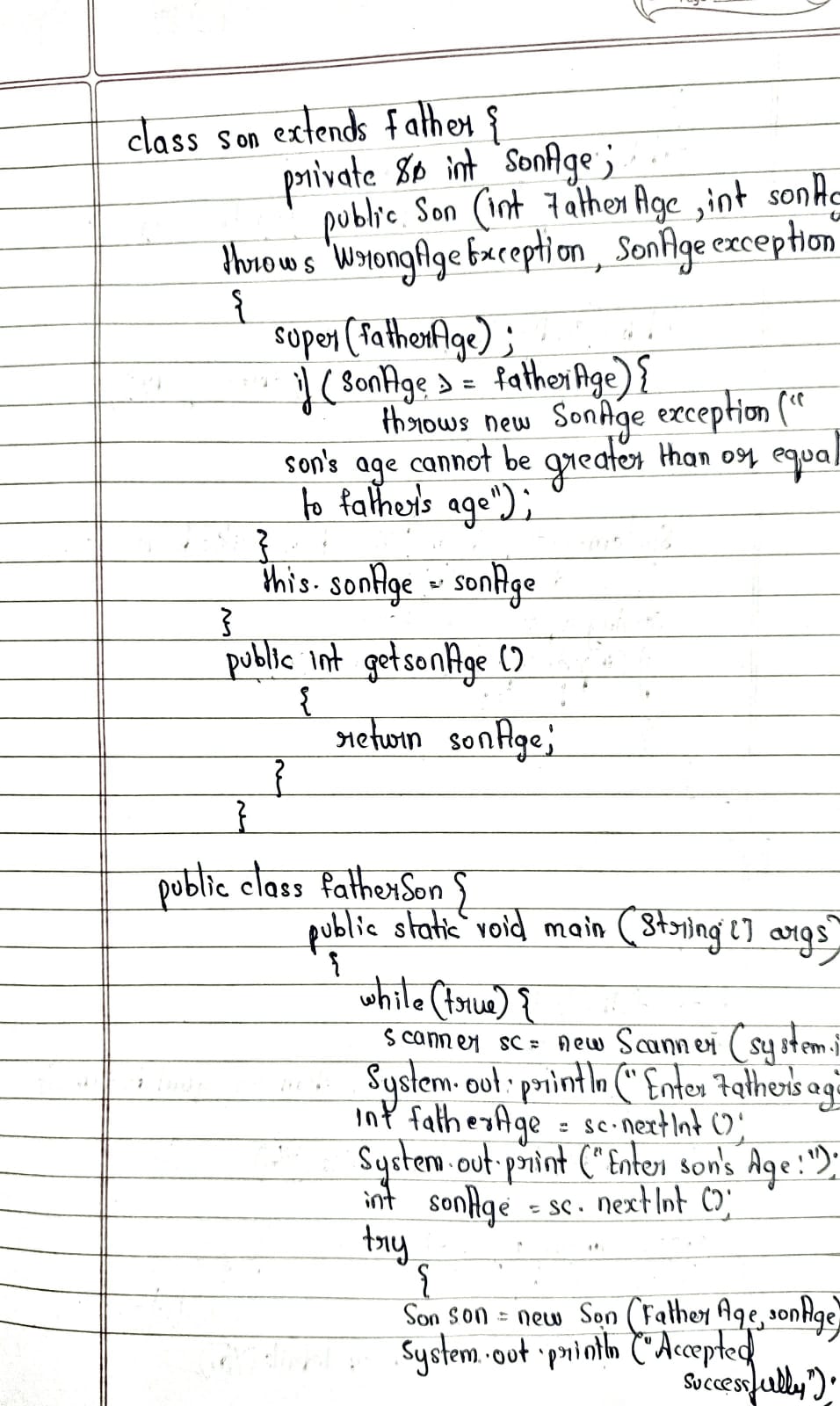
****

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%207>

**Program 7**

**Implement Exception Handling**

****



**Source 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 FatherandSon{

public static void main(String[] args) {

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("Would you like to re-enter details (Y/n)");

String input = sc.next();

if (input.equalsIgnoreCase("n")) {

break;

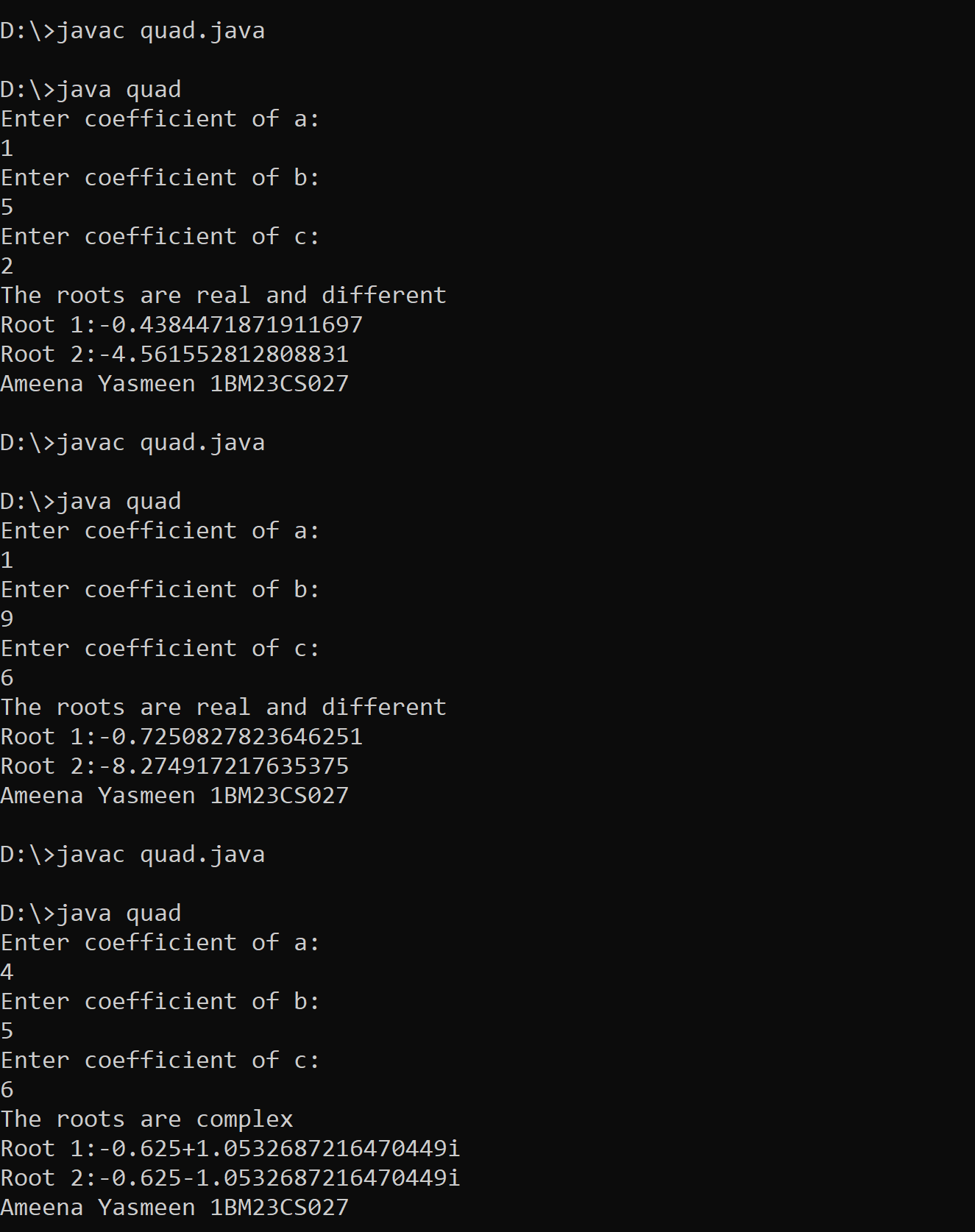
}

}

}

}

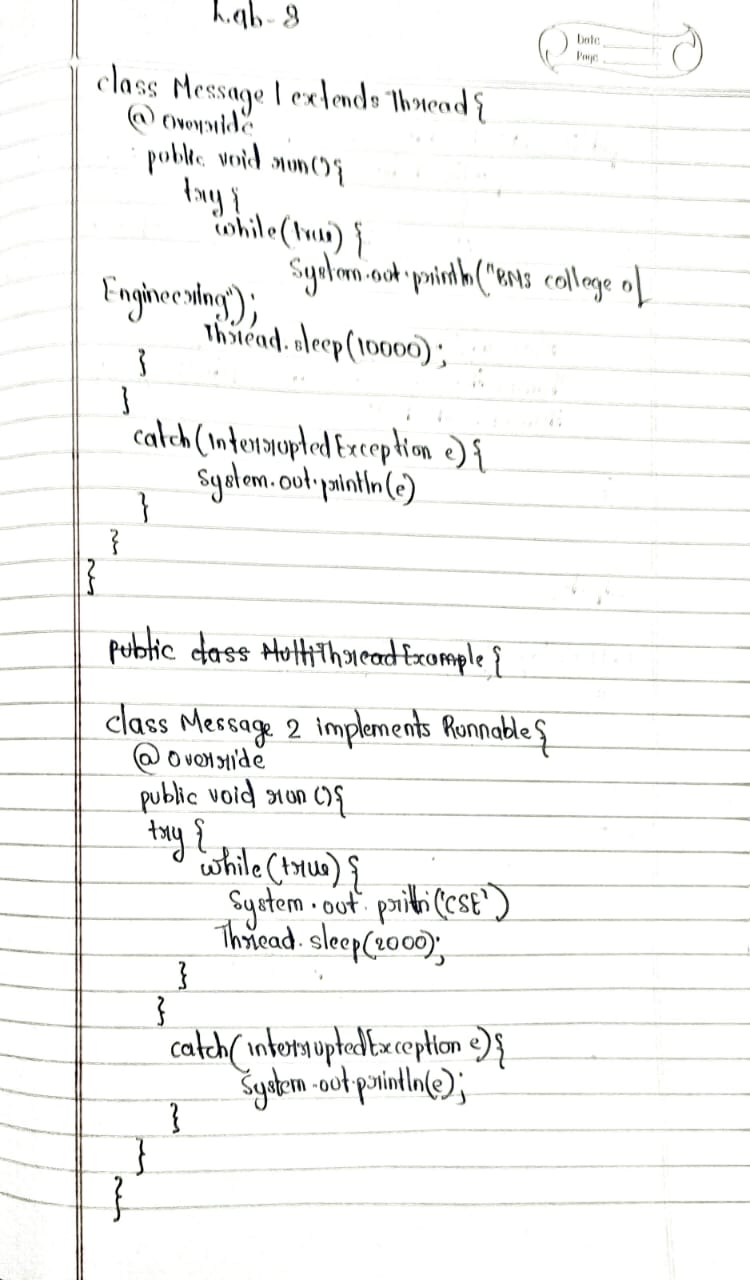
**Output:**

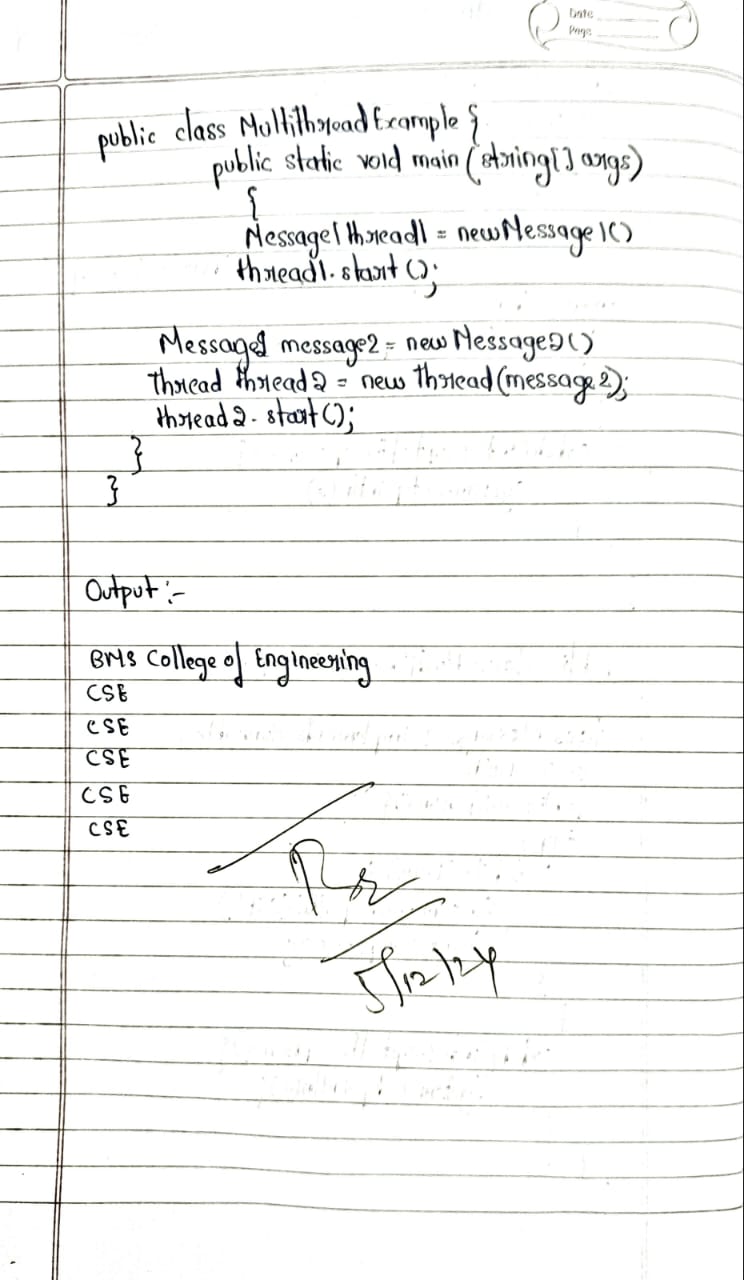


<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%208>

**Program 8**

**MultiThreading, creating Threads in Java**





**Source code:**

class Message1 extends Thread {

@Override

public void run() {

try {

while (true) {

System.out.println("BMS College of Engineering");

Thread.sleep(10000);

}

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

class Message2 implements Runnable {

@Override

public void run() {

try {

while (true) {

System.out.println("CSE");

Thread.sleep(2000);

}

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

public class MultiThreadExample {

public static void main(String[] args) {

System.out.println("Ameena Yasmeen\n1BM23CS027");

Message1 thread1 = new Message1();

thread1.start();

Message2 message2 = new Message2();

Thread thread2 = new Thread(message2);

thread2.start();

}

}

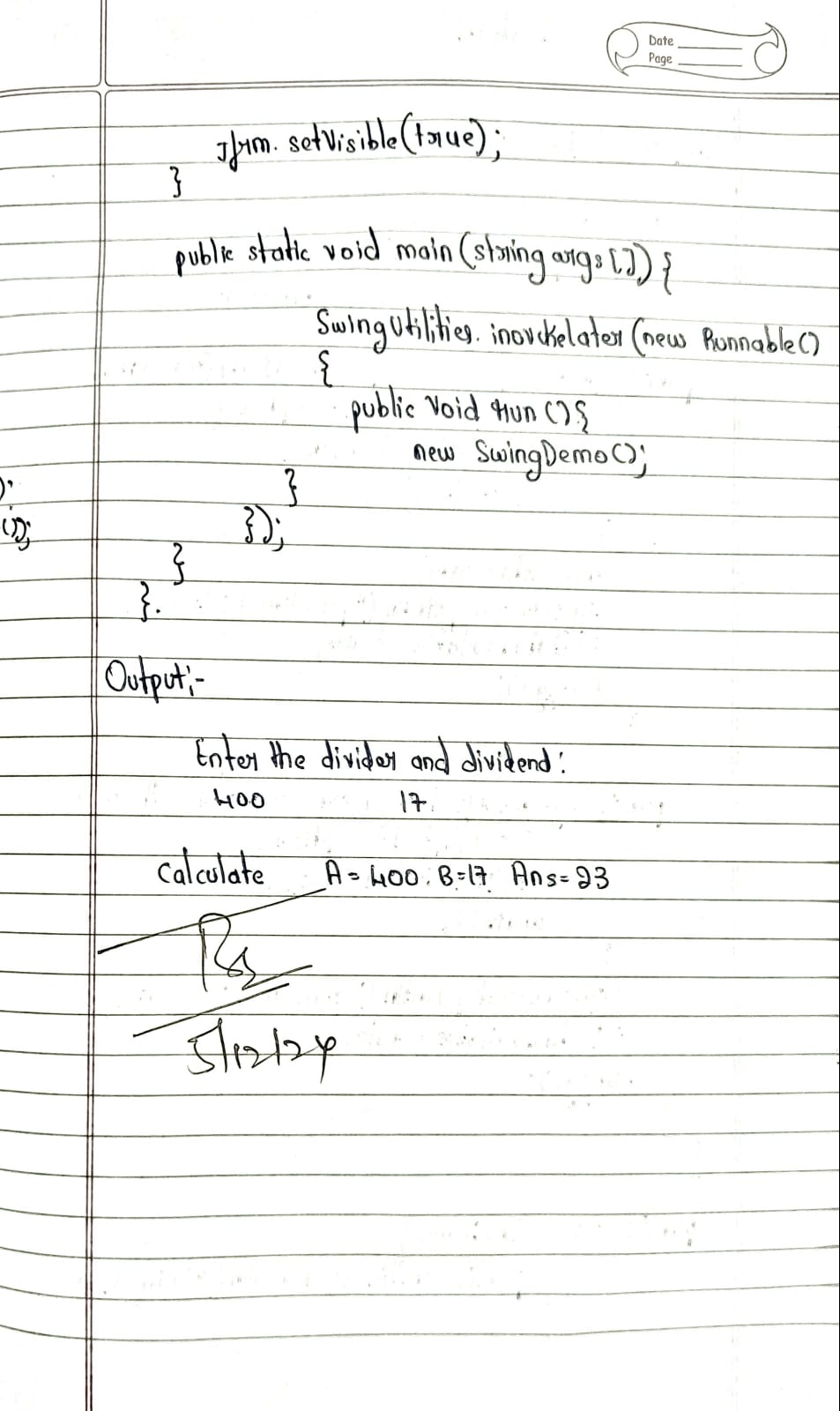
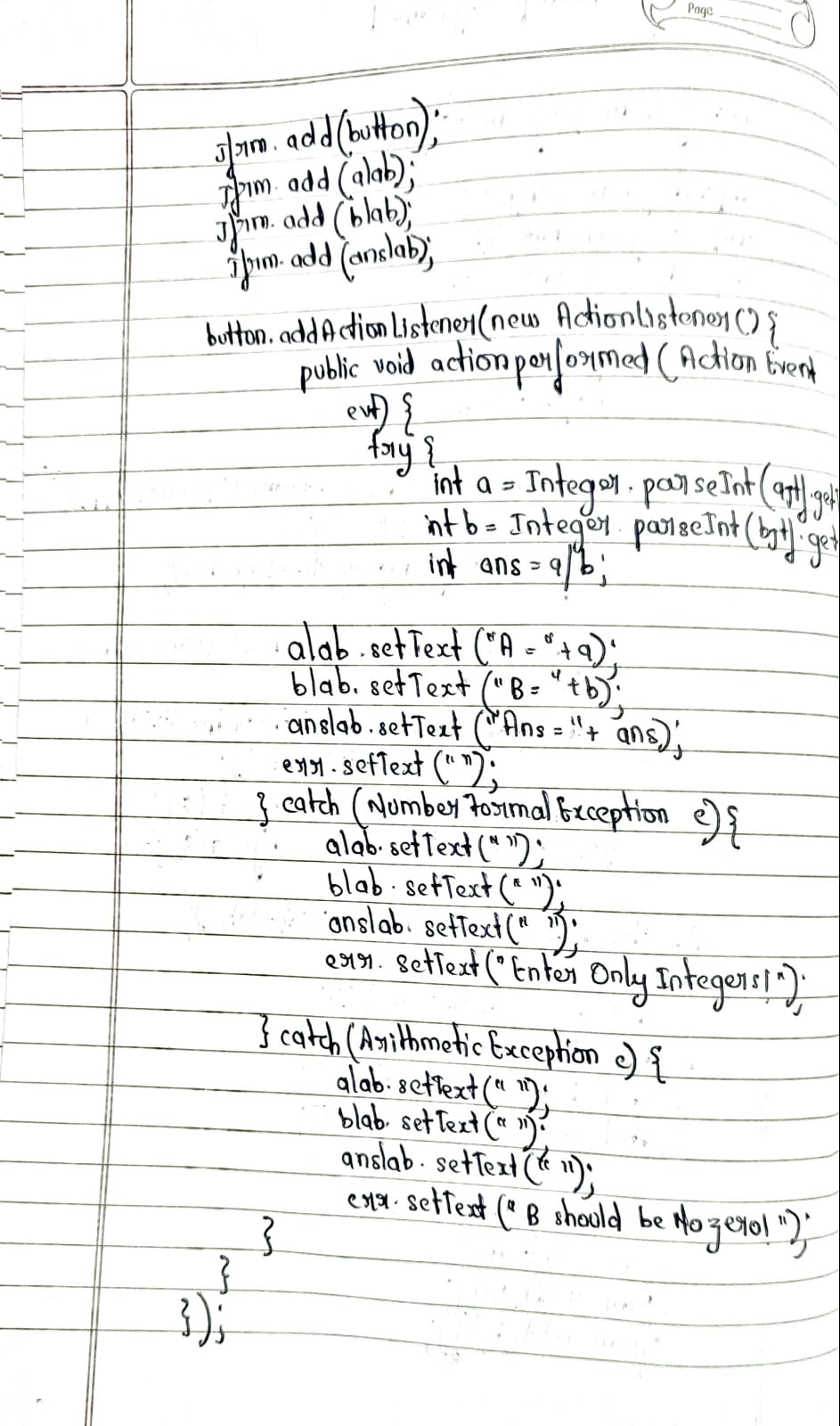
**Output:**

****

[**https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%209**](https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%209)

**Program 9**

**Interface To Perform Integer Division**

****

**Source Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class SwingDemo {

SwingDemo() {

JFrame jfrm = new JFrame("Divider App");

jfrm.setSize(275, 150);

jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel jlab = new JLabel("Enter the divider and dividend:");

JTextField ajtf = new JTextField(8);

JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

JLabel err = new JLabel();

JLabel alab = new JLabel();

JLabel blab = new JLabel();

JLabel anslab = new JLabel();

jfrm.add(err);

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {

try {

int a = Integer.parseInt(ajtf.getText());

int b = Integer.parseInt(bjtf.getText());

int ans = a / b;

alab.setText("A = " + a);

blab.setText("B = " + b);

anslab.setText("Ans = " + ans);

err.setText("");

} catch (NumberFormatException e) {

alab.setText("");

blab.setText("");

anslab.setText("");

err.setText("Enter Only Integers!");

} catch (ArithmeticException e) {

alab.setText("");

blab.setText("");

anslab.setText("");

err.setText("B should be NON zero!");

}

}

});

jfrm.setVisible(true);

}

public static void main(String args[]) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new SwingDemo();

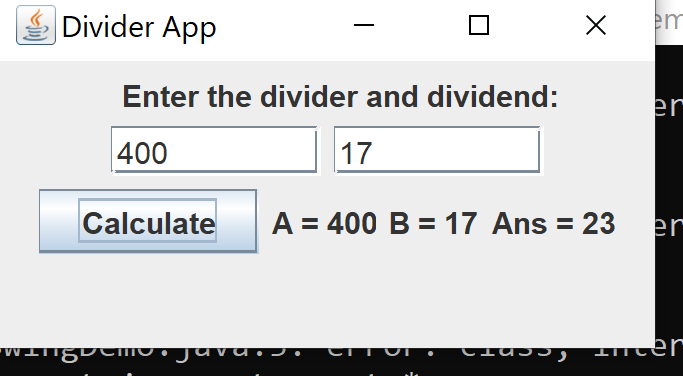
}

});

}

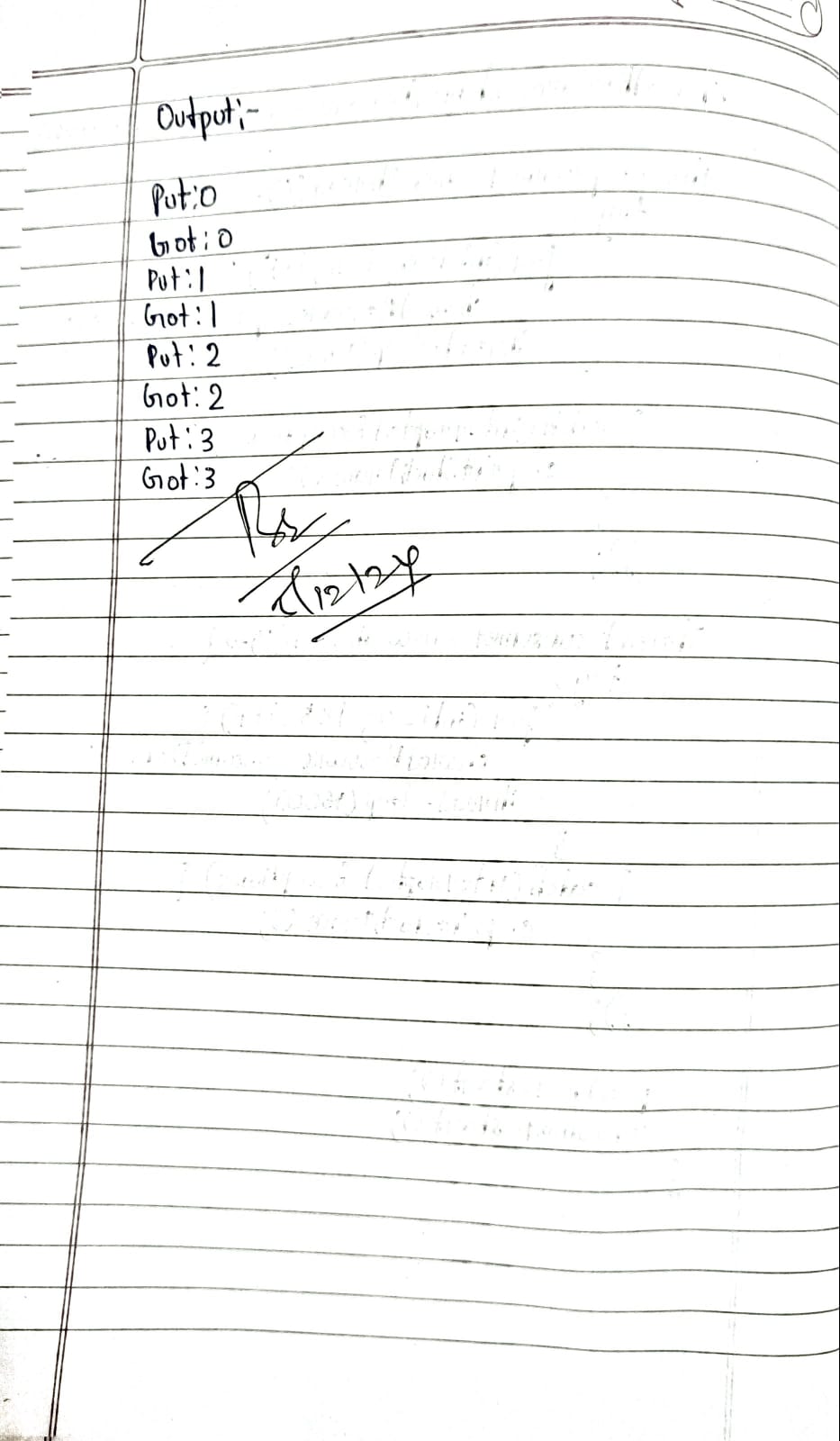
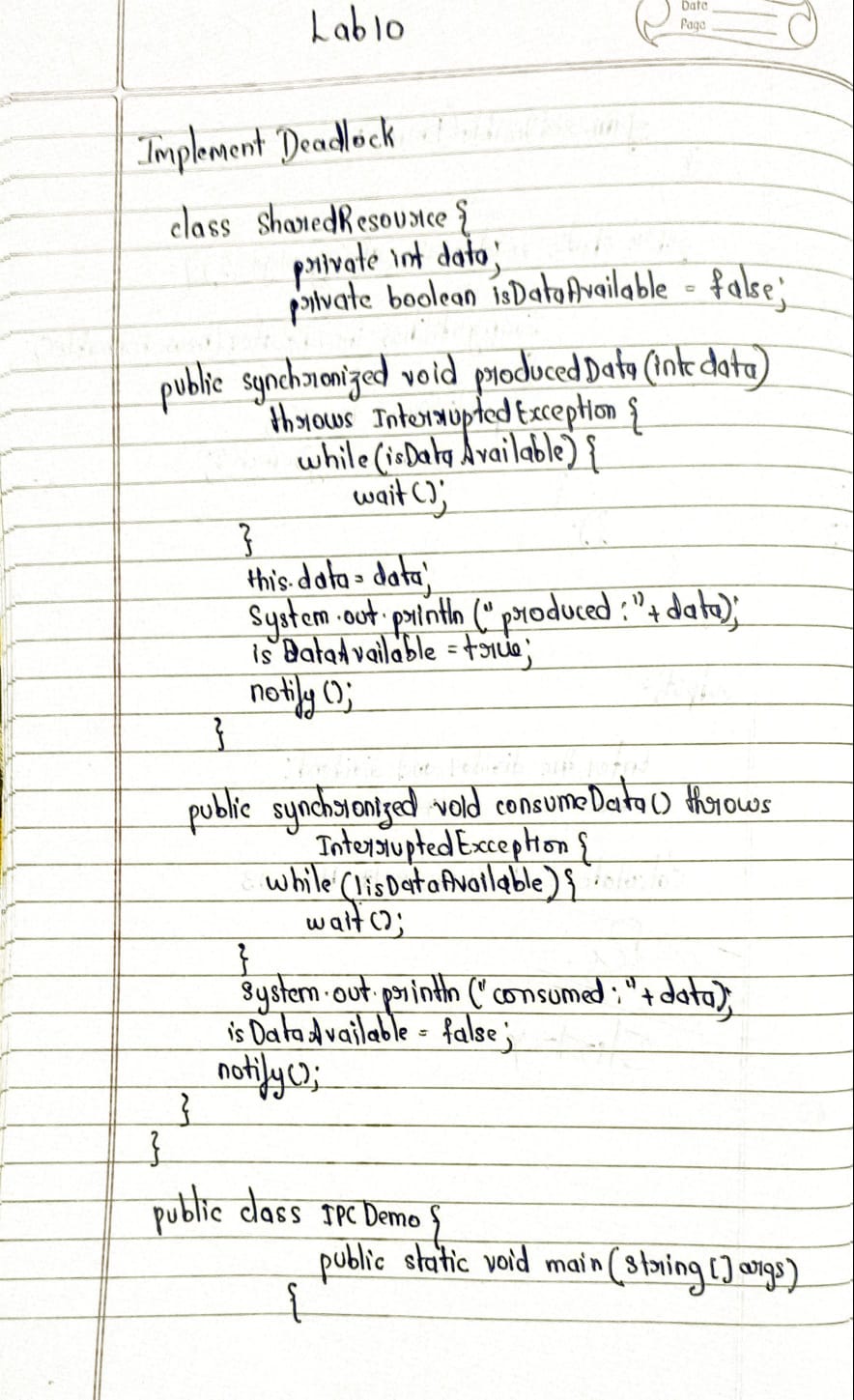
}

**Output:**

****

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%2010%20(a)>

**Program 10 (a)**

**Implement Deadlock**

**Source Code:**

class SharedResource {

private int data;

private boolean isDataAvailable = false;

public synchronized void produceData(int data) throws InterruptedException {

while (isDataAvailable) {

wait();

}

this.data = data;

System.out.println("Put: " + data);

isDataAvailable = true;

notify();

}

public synchronized void consumeData() throws InterruptedException {

while (!isDataAvailable) {

wait();

}

System.out.println("Got: " + data);

isDataAvailable = false;

notify();

}

}

public class IPCDemo {

public static void main(String[] args) {

// Print your name once at the beginning

System.out.println("Ameena Yasmeen\n 1BM23CS027");

SharedResource sharedResource = new SharedResource();

Thread producer = new Thread(() -> {

try {

for (int i = 0; i < 5; i++) {

sharedResource.produceData(i);

Thread.sleep(1000);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

});

Thread consumer = new Thread(() -> {

try {

for (int i = 0; i < 5; i++) {

sharedResource.consumeData();

Thread.sleep(1500);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

});

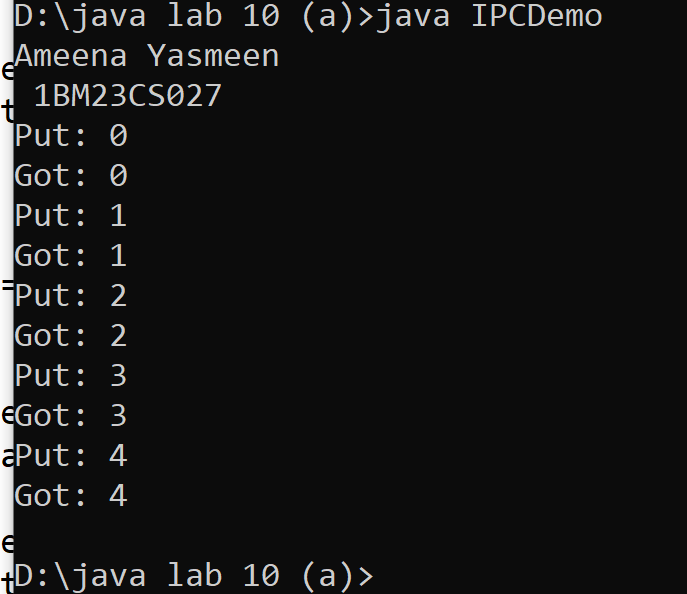
producer.start();

consumer.start();

}

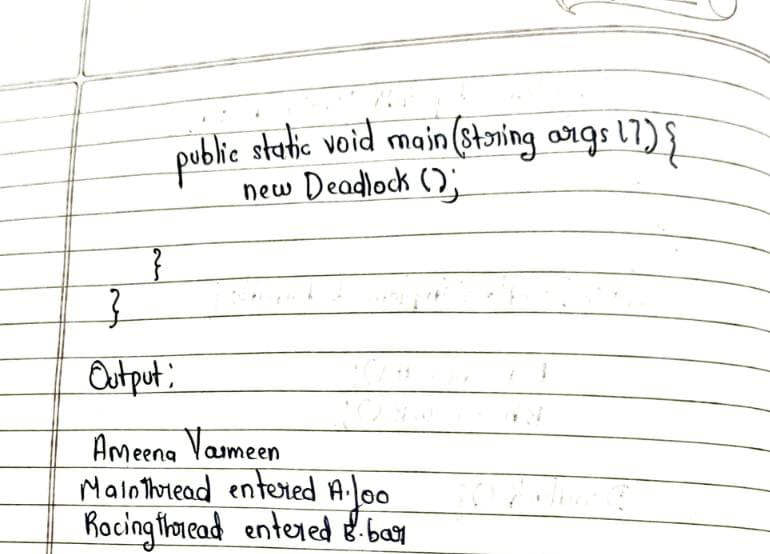
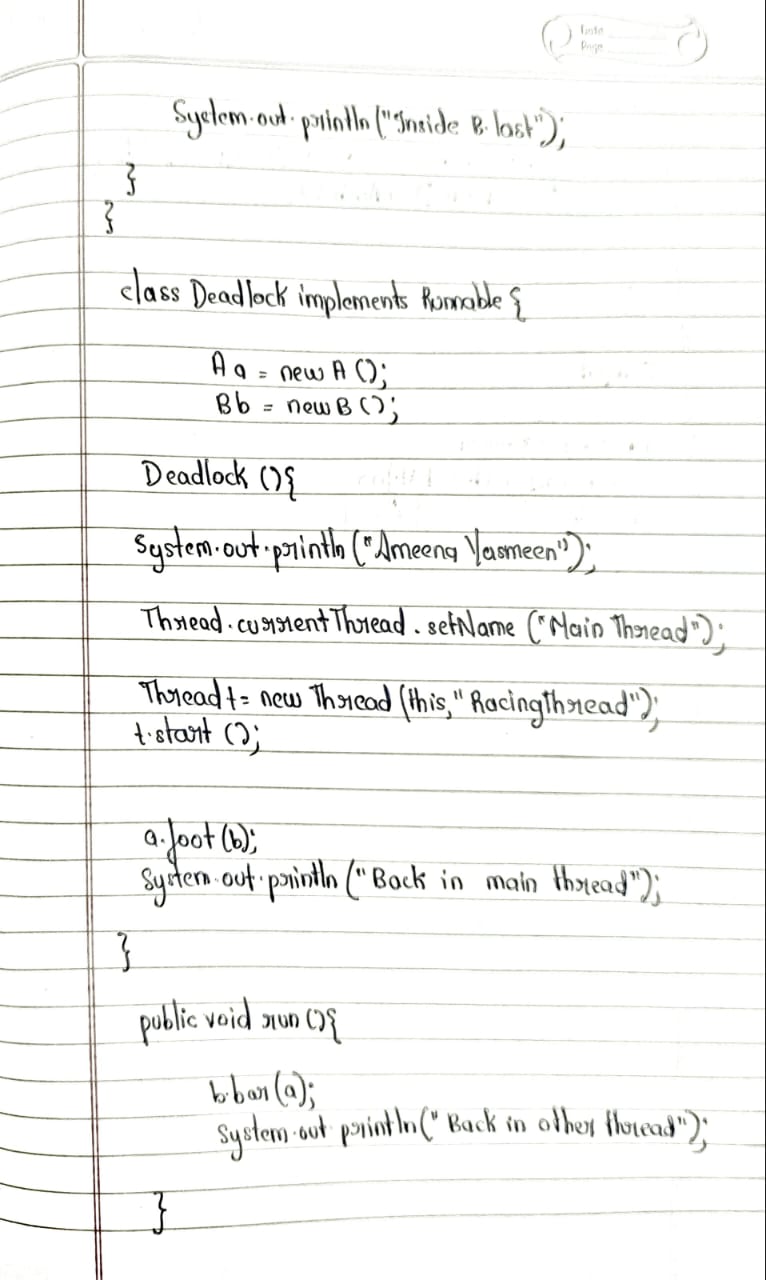
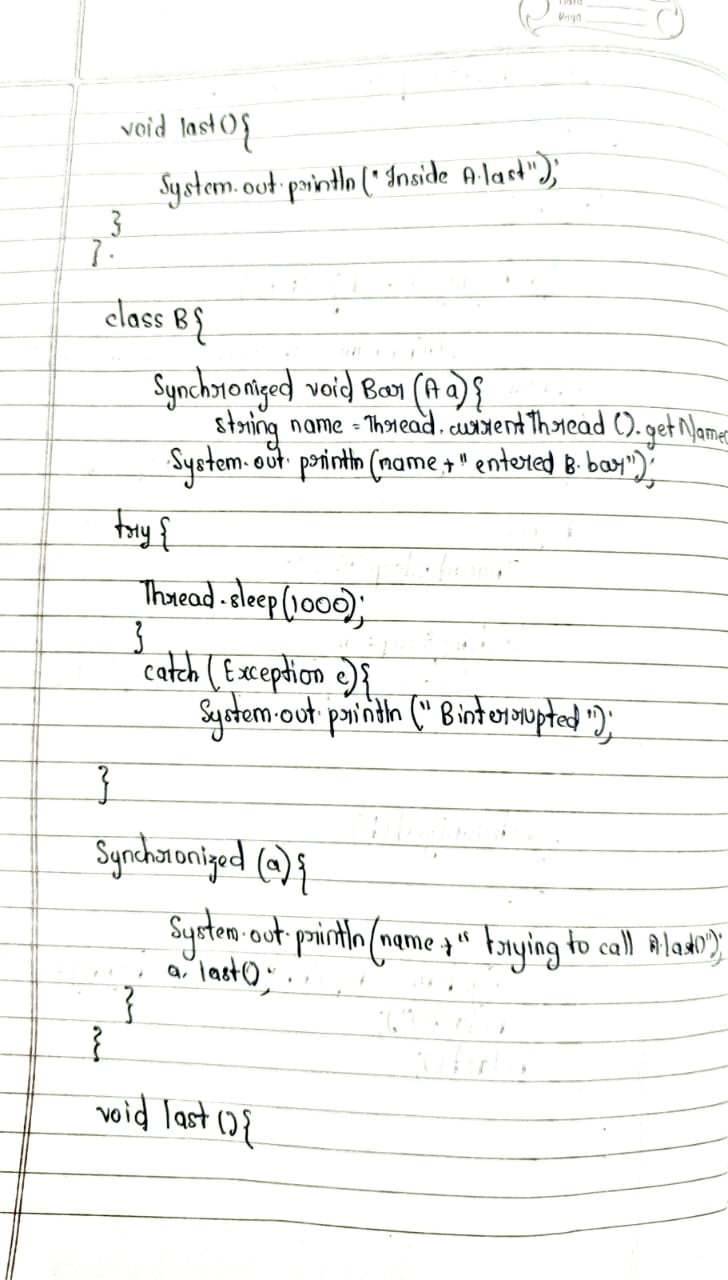
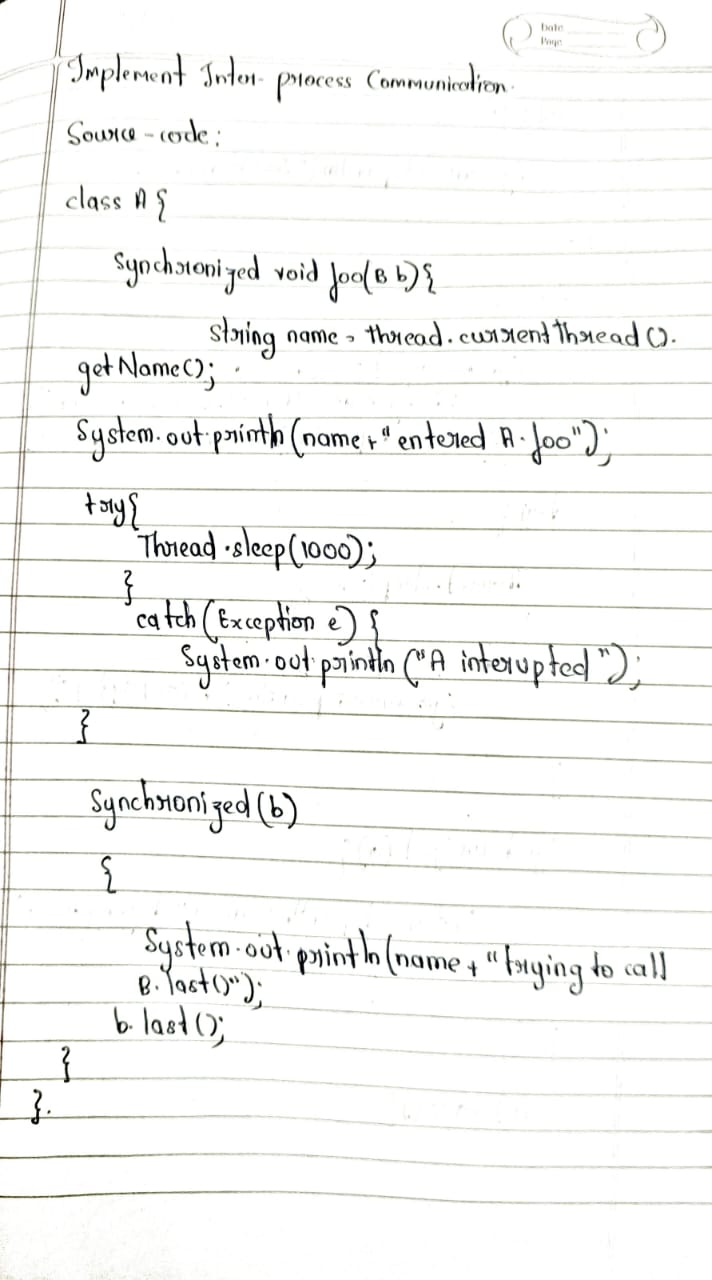
}

Output:



<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%2010%20b>

**Program 10(b)**

**Implement Inter-Process Communication**

**Source Code:**

class A {

synchronized void foo(B b) {

String name = Thread.currentThread().getName();

System.out.println(name + " entered A.foo");

try {

Thread.sleep(1000);

} catch (Exception e) {

System.out.println("A Interrupted");

}

synchronized (b) {

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

b.last();

}

}

void last() {

System.out.println("Inside A.last");

}

}

class B {

synchronized void bar(A a) {

String name = Thread.currentThread().getName();

System.out.println(name + " entered B.bar");

try {

Thread.sleep(1000);

} catch (Exception e) {

System.out.println("B Interrupted");

}

synchronized (a) {

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

a.last();

}

}

void last() {

System.out.println("Inside B.last");

}

}

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock() {

System.out.println("Ameena Yasmeen\n1BM23CS027");

Thread.currentThread().setName("MainThread");

Thread t = new Thread(this, "RacingThread");

t.start();

a.foo(b);

System.out.println("Back in main thread");

}

public void run() {

b.bar(a);

System.out.println("Back in other thread");

}

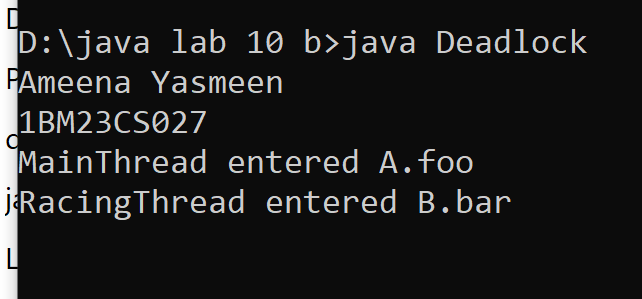
public static void main(String args[]) {

new Deadlock();

}

}

**Output:**

****

**-------**