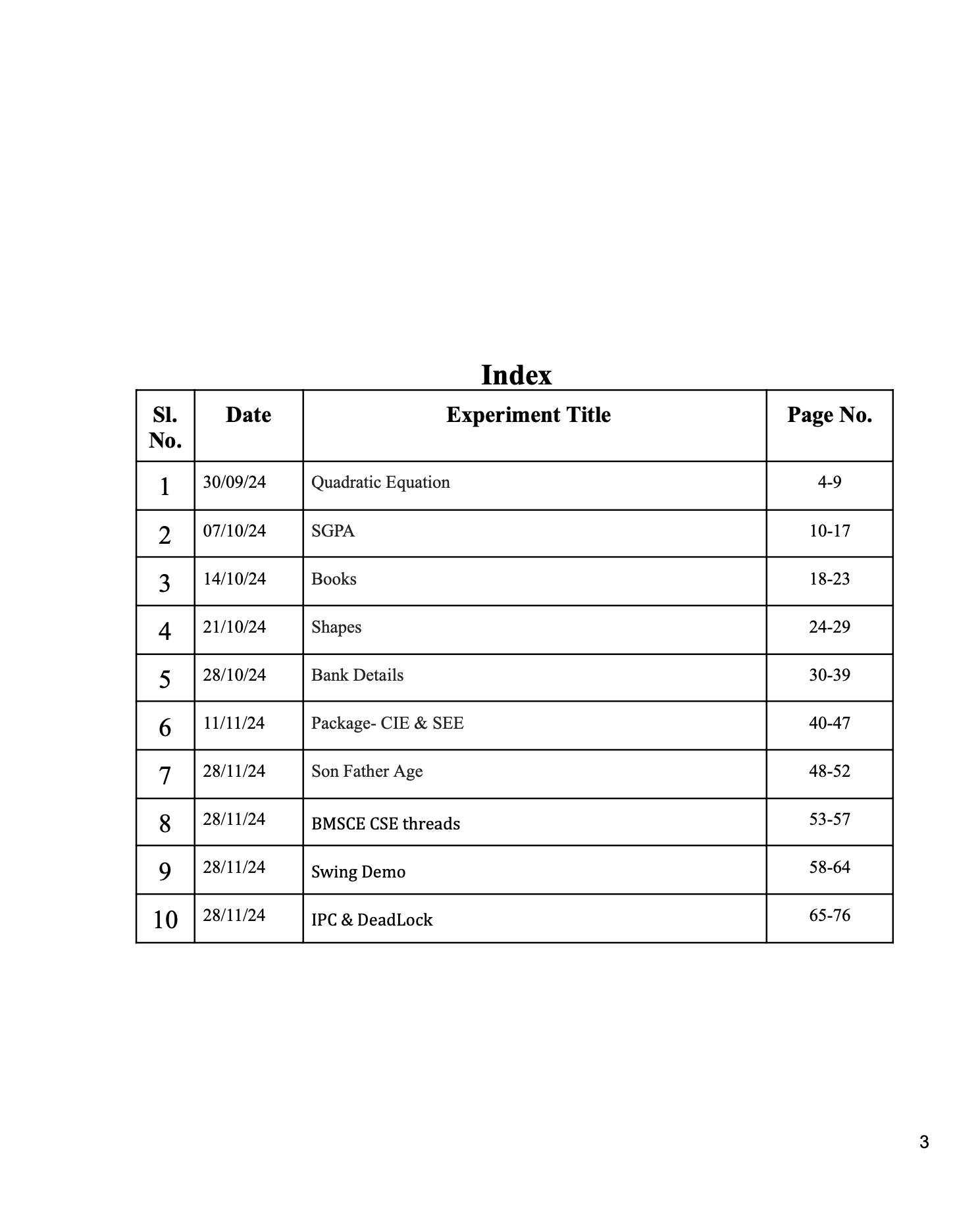
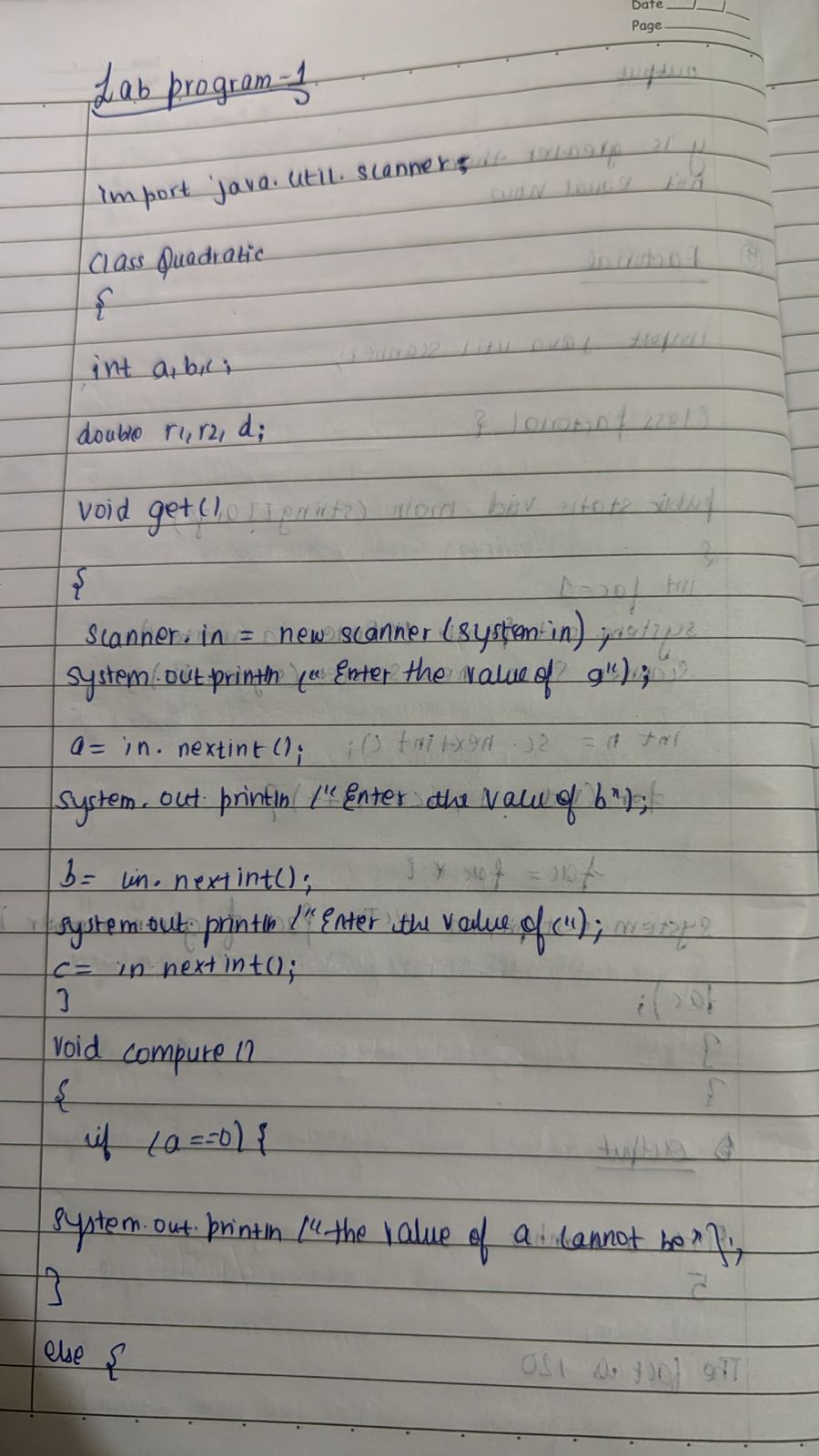
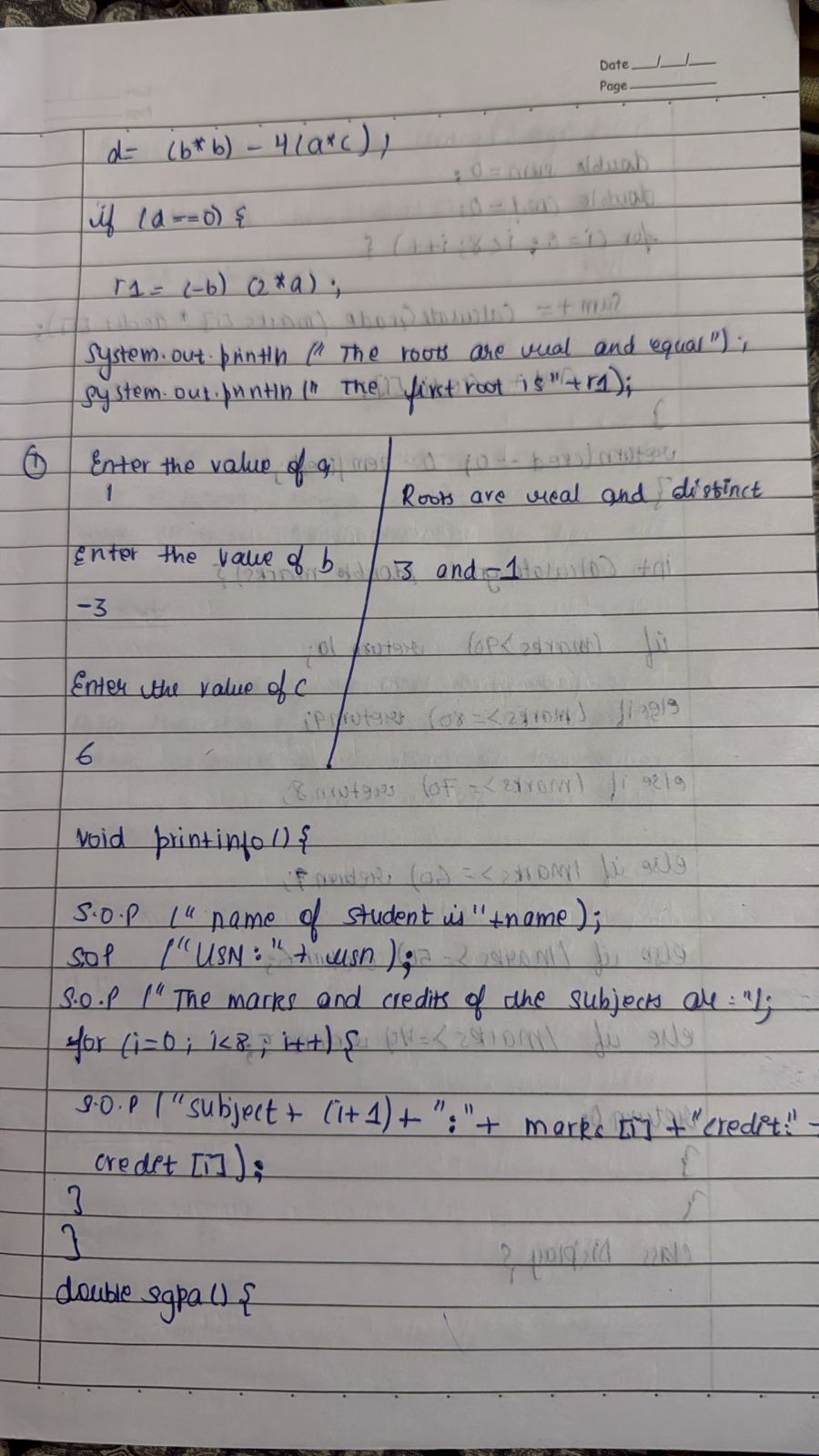




S







import java.util.Scanner;

class Quadratic{

int a,b,c;

double r1,r2,d;

void get()

{

Scanner in= new Scanner(System.in);

System.out.println("enter a");

a=in.nextInt();

System.out.println("enter b");

b=in.nextInt();

System.out.println("enter c");

c=in.nextInt();

}

void compute()

{

if(a==0) {

System.out.println("value of a cannot be 0");

}

else {

d=(b\*b)-(4\*a\*c);

if(d==0) {

r1=(-b)/(2\*a);

System.out.println("roots are equal");

System.out.println("first and seconf roots are "+ r1);

}

else if(d>0) {

r1=(-b + Math.sqrt(d))/(2\*a);

r2=(-b - Math.sqrt(d))/(2\*a);

System.out.println("roots are real and distnct");

System.out.println("first root is" + r1 + " and second root is " + r2);

}

else

{

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

double imaginary=((Math.sqrt(-d)))/(2\*a);

System.out.print("roots are imaginary");

System.out.println("first root is" + real + "+" + imaginary +"i");

System.out.println("second root is" + real + "-" + imaginary +"i");

}

}

}

}

class QuadraticMain

{

public static void main(String s[])

{

Quadratic quad=new Quadratic();

quad.get();

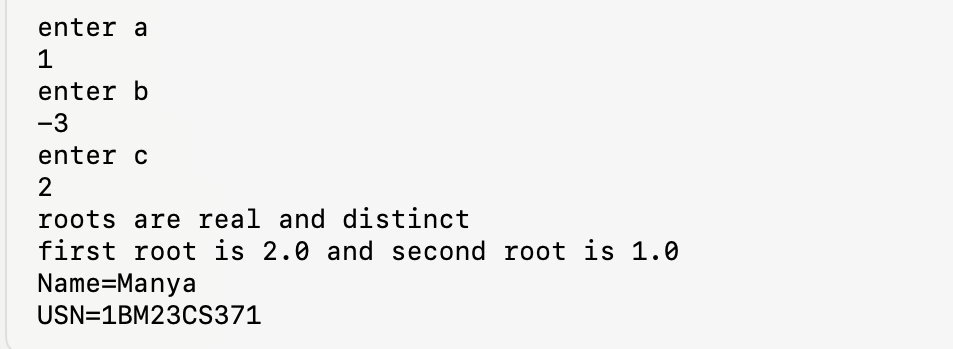
quad.compute();

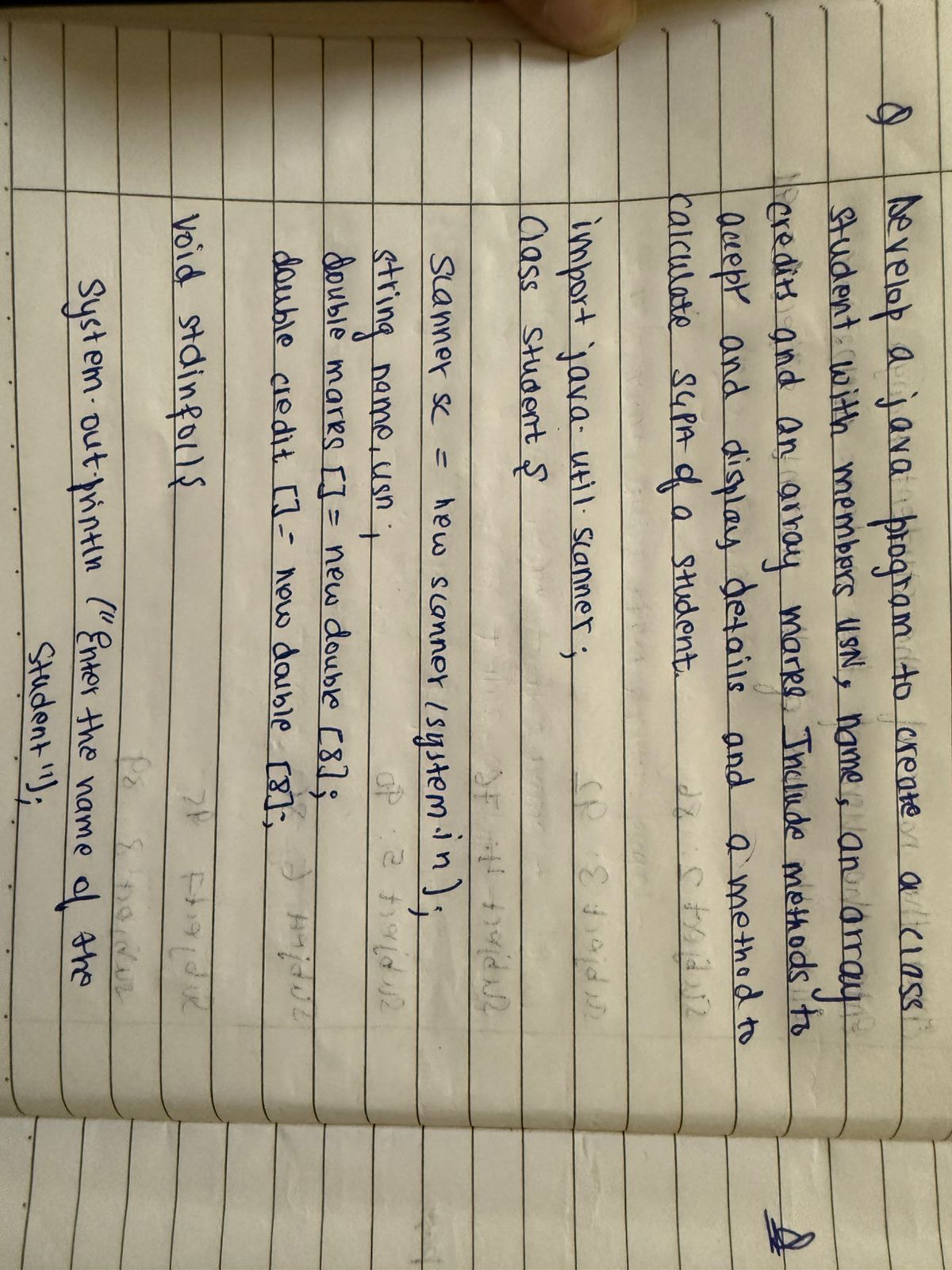
System.out.println("Name=Manya");

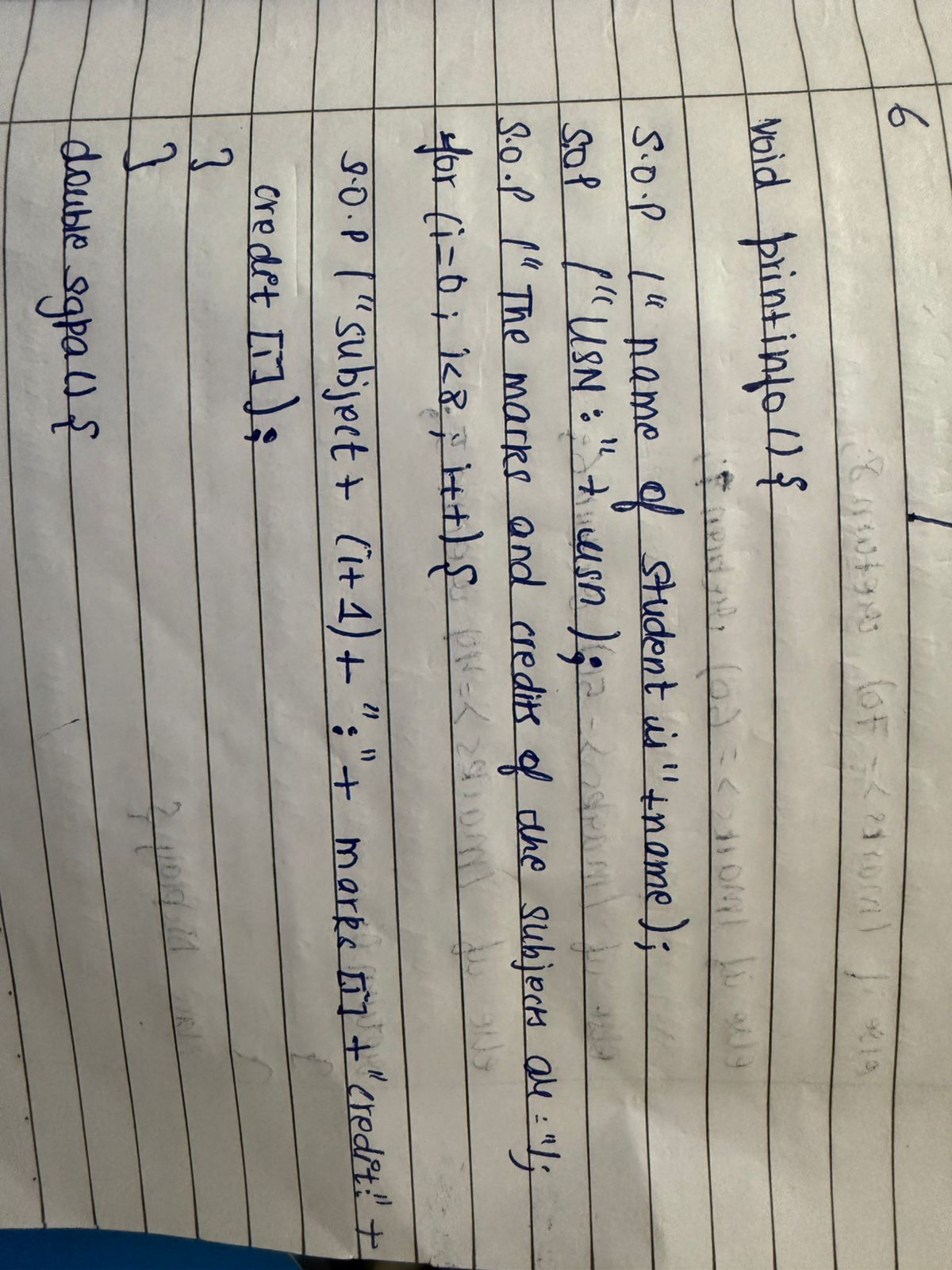
System.out.println("USN=1BM23CS371");

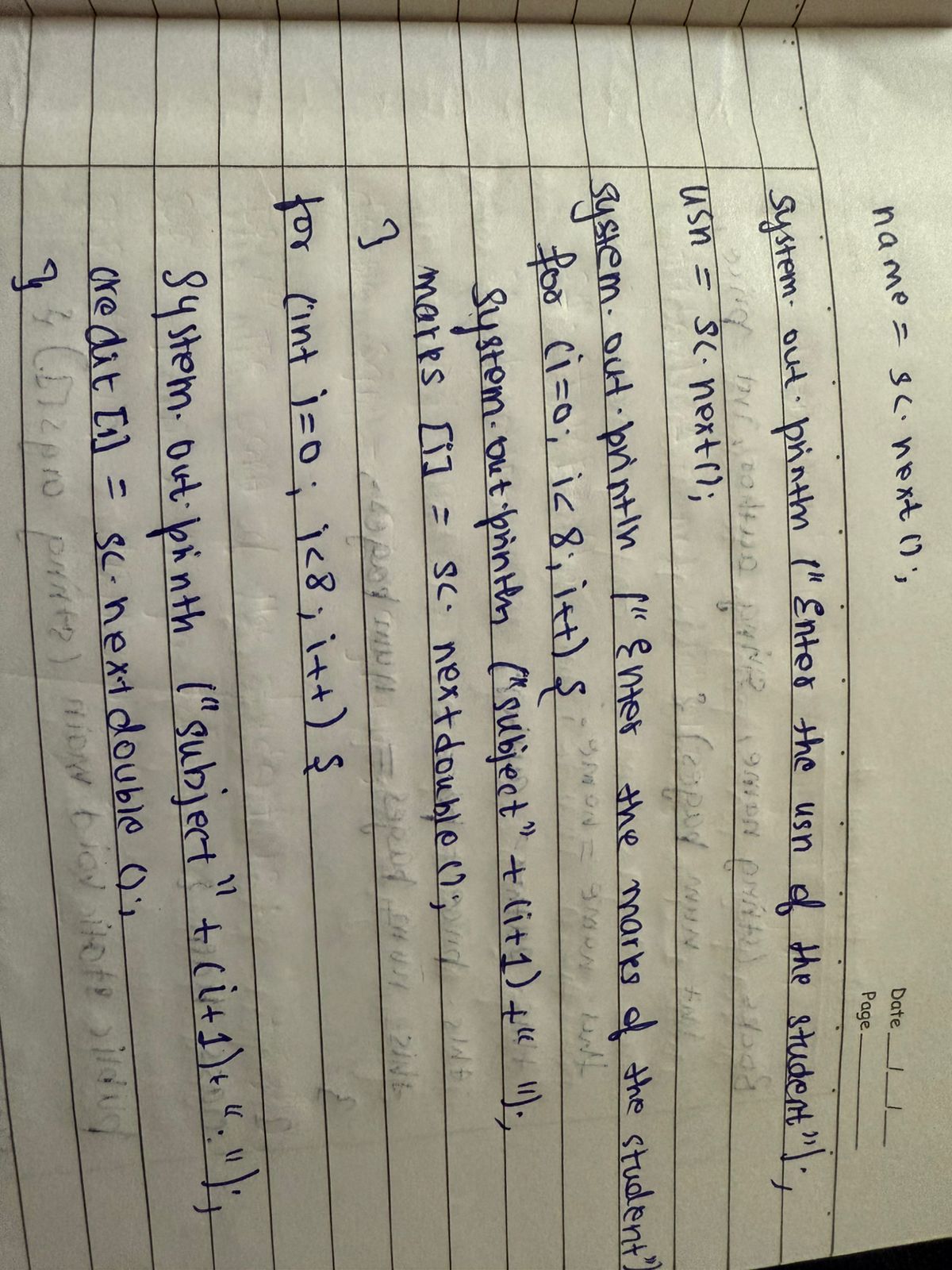
}

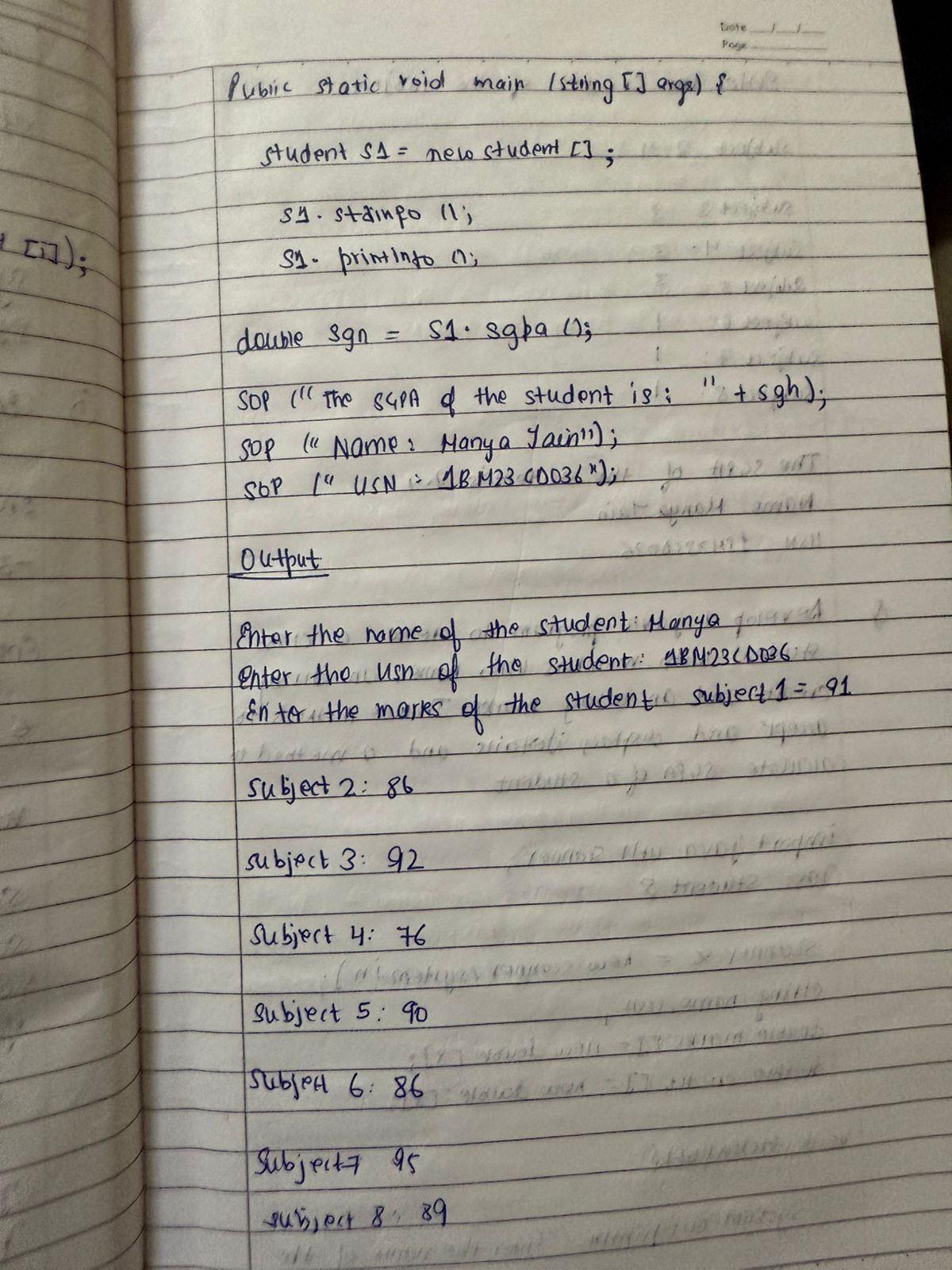
}











import java.util.Scanner;

class Subject {

int grade;

int credits;

}

class Student {

String usn;

String name;

double SGPA;

Subject[] subjects;

Student() {

subjects = new Subject[8];

for (int i = 0; i < 8; i++) {

subjects[i] = new Subject();

}

}

void getDetails(Scanner sc) {

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

usn = sc.nextLine();

System.out.println("Enter name:");

name = sc.nextLine();

}

void getMarks(Scanner sc) {

double totalScore = 0;

int totalCredits = 0;

System.out.println("Enter marks for 8 subjects:");

for (int j = 0; j < 8; j++) {

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

int marks = sc.nextInt();

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

int credits = sc.nextInt();

int grade = (marks / 10) + 1;

if (grade > 10) grade = 10;

subjects[j].credits = credits;

subjects[j].grade = grade;

totalScore += grade \* credits;

totalCredits += credits;

}

SGPA = totalScore / totalCredits;

}

void displaySGPA() {

System.out.println("SGPA of student " + name + " (" + usn + "): " + SGPA);

}

}

public class Main{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Name: MANYA JAIN");

System.out.println("USN:1BM23CS371");

System.out.println("Enter the number of students:");

int numStudents = sc.nextInt();

sc.nextLine();

Student[] students = new Student[numStudents];

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

System.out.println("Enter details for student " + (i + 1));

students[i] = new Student();

students[i].getDetails(sc);

students[i].getMarks(sc);

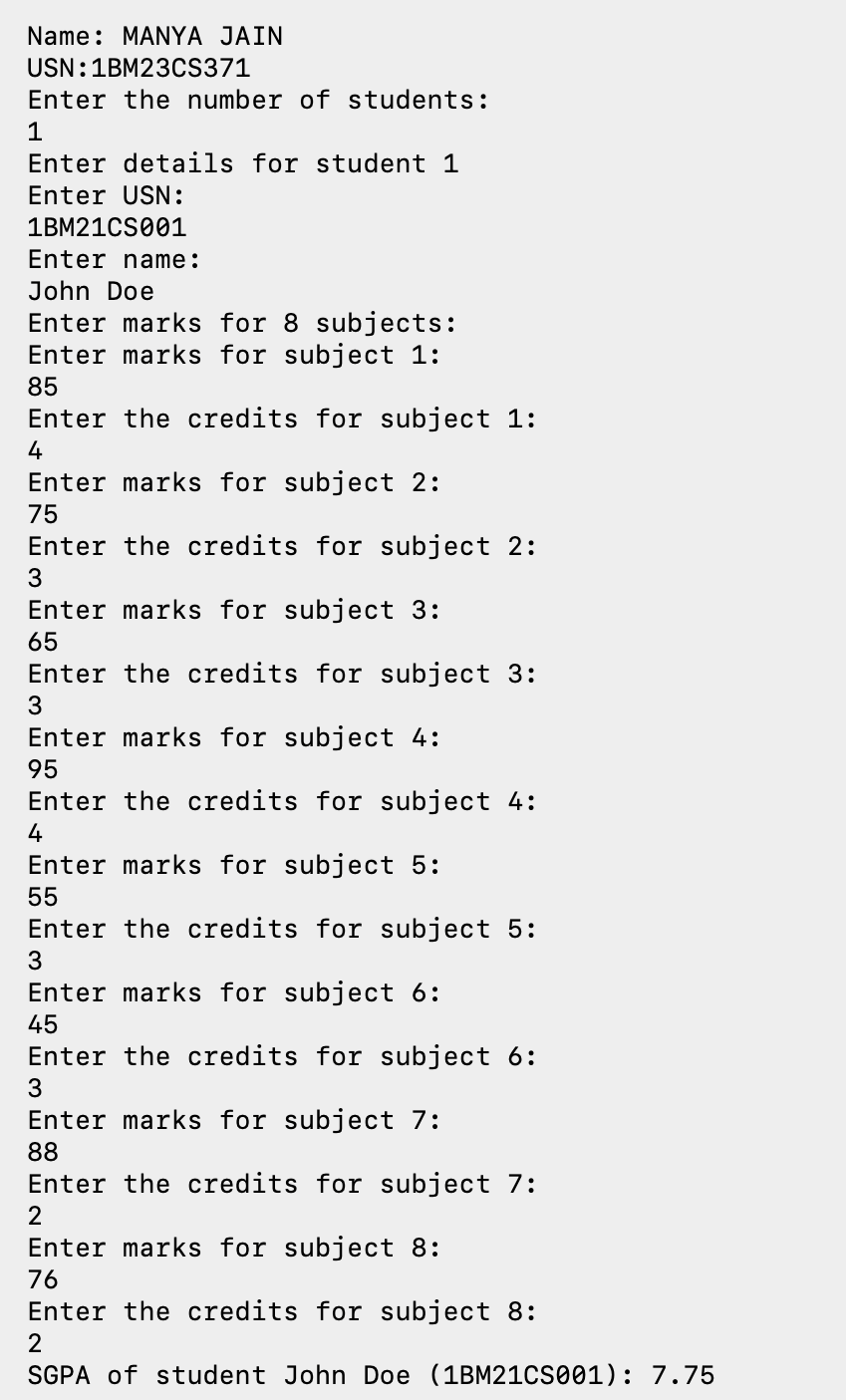
students[i].displaySGPA();

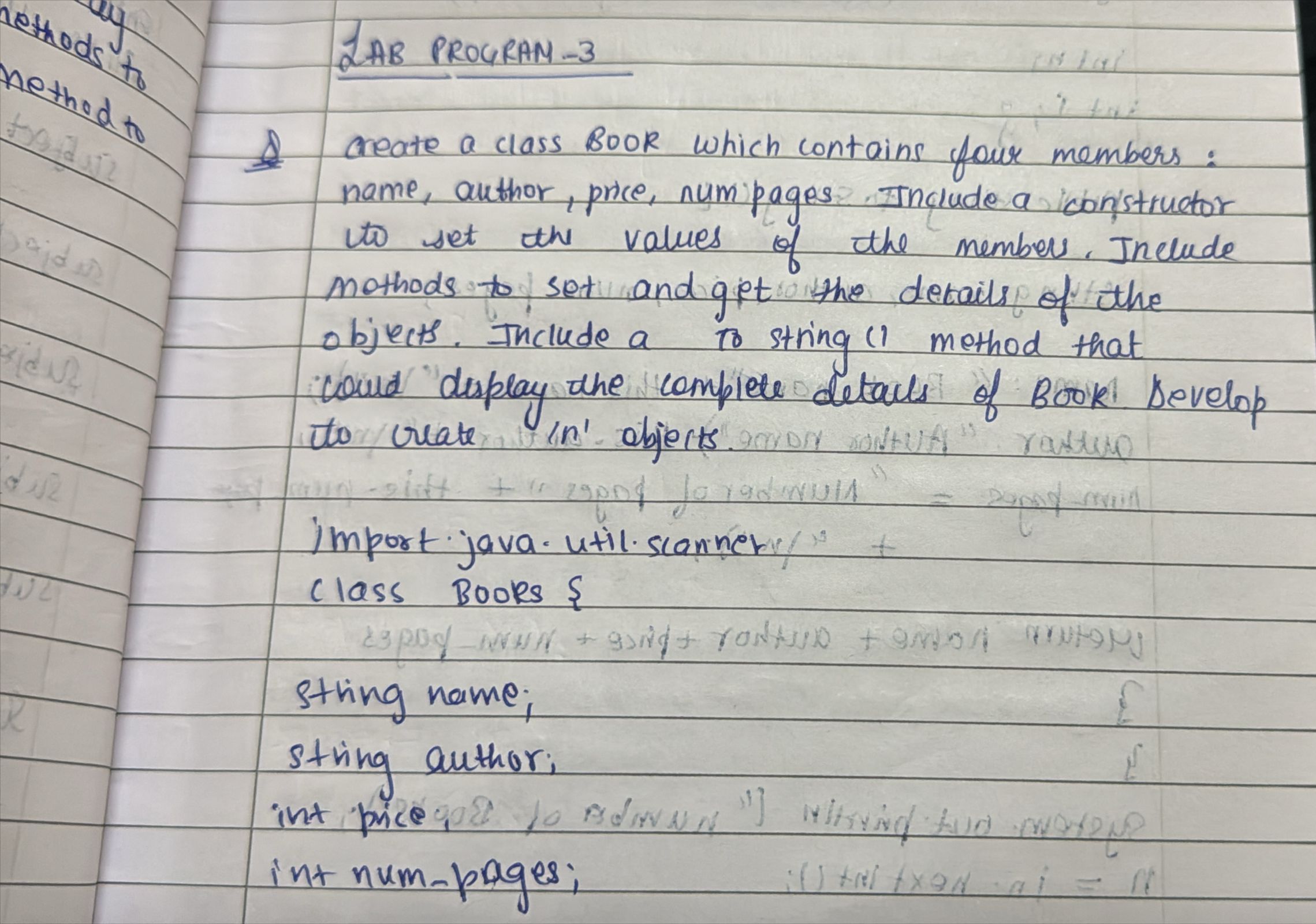
}

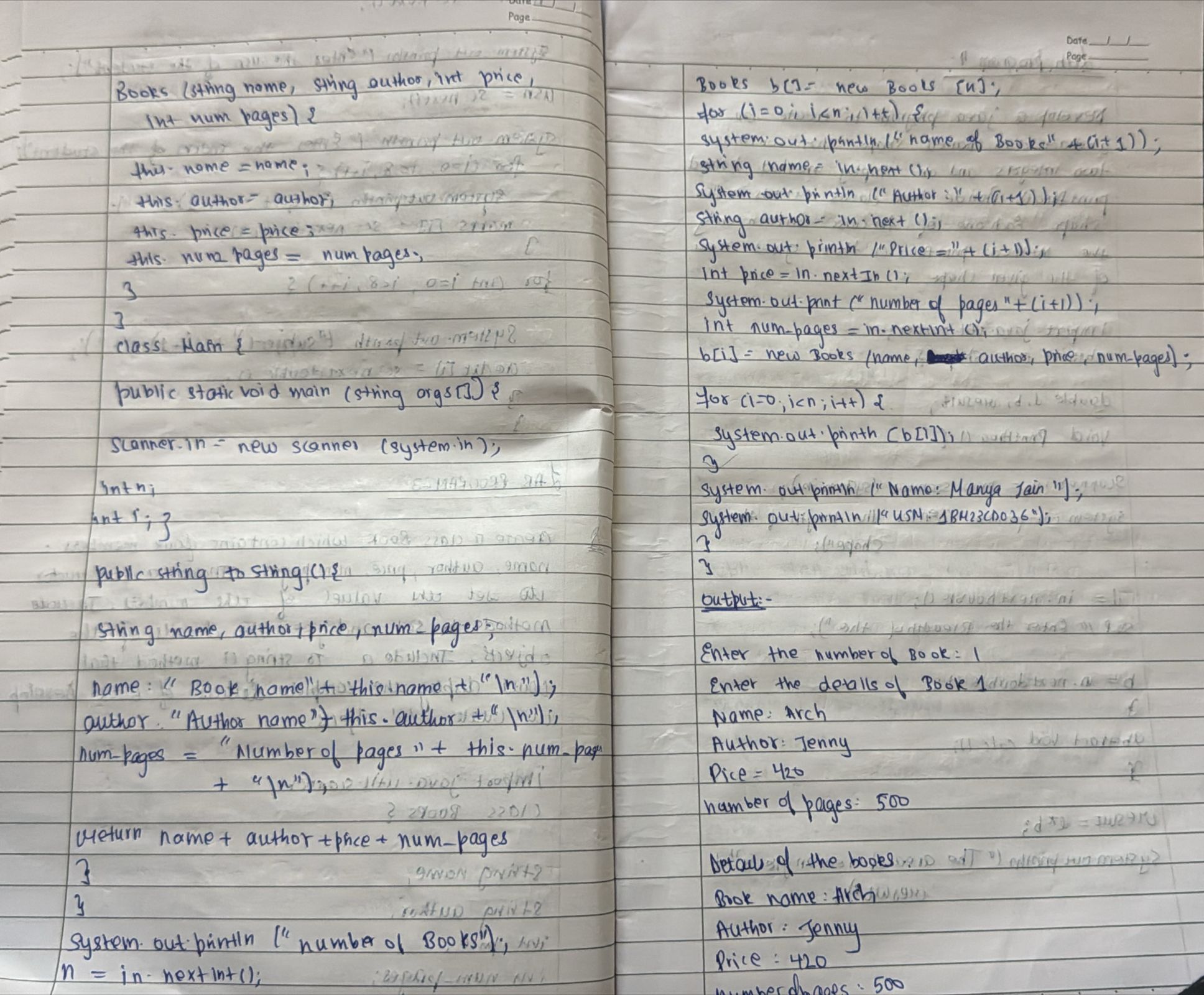
sc.close();

}

}







import java.util.Scanner;

class Books{

String name;

String author;

int price;

int num\_pages;

Books(String name, String author, int price, int num\_pages){

this.name=name;

this.author=author;

this.price=price;

this.num\_pages=num\_pages;

}

public String toString(){

String name,author,price,num\_pages;

name="Book name: "+this.name+"/n";

author="Author name: "+this.author+"\n";

price="Price: "+this.price+"\n";

num\_pages="Number of pages: "+this.num\_pages+"\n";

return name+author+price+num\_pages;

}

}

class Main\_{

public static void main(String args[])

{

Scanner in =new Scanner(System.in);

int n;

int i;

System.out.println("number of books");

n=in.nextInt();

Books b[]= new Books[n];

for (i=0;i<n;i++){

System.out.println("Name of book"+(i+1));

String name=in.next();

System.out.println("Author"+(i+1));

String author=in.next();

System.out.println("Price"+(i+1));

int price=in.nextInt();

System.out.println("Number of pages"+(i+1));

int num\_pages=in.nextInt();

b[i]=new Books(name,author,price,num\_pages);

}

for (i=0;i<n;i++){

System.out.println(b[i]);

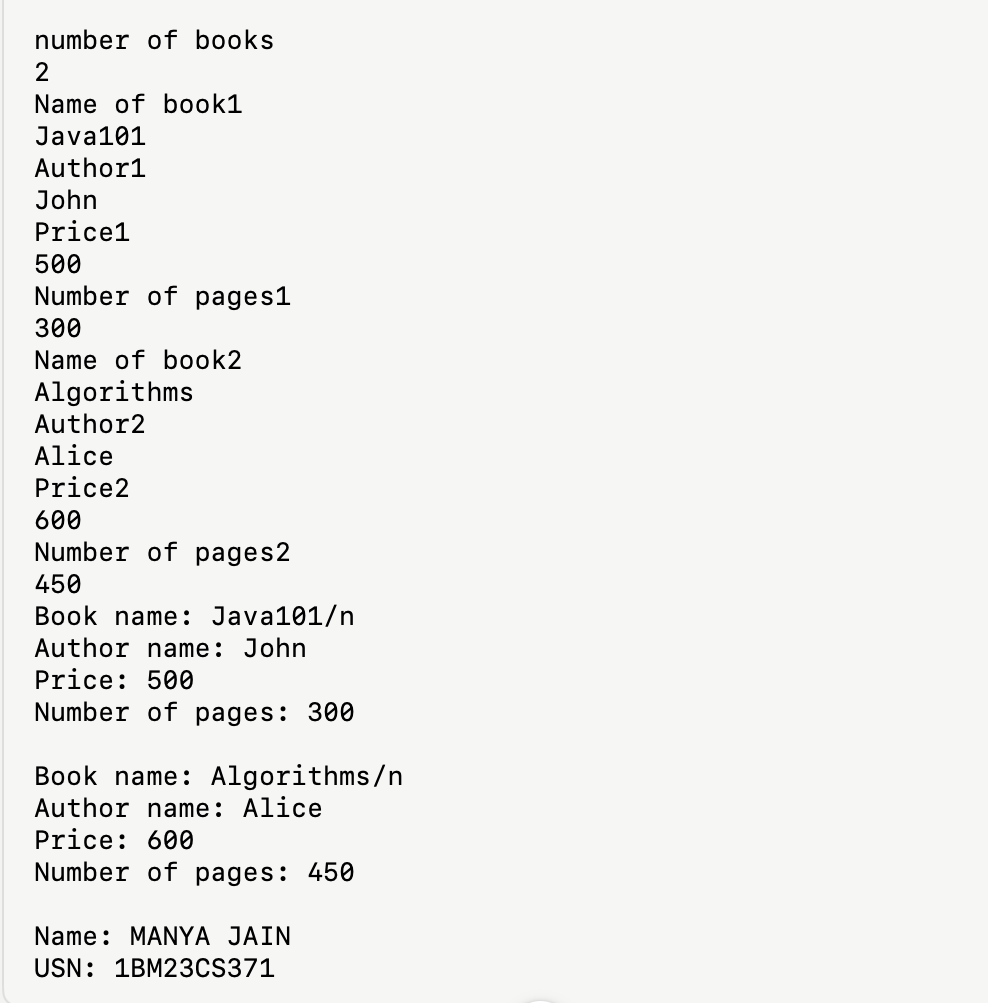
}

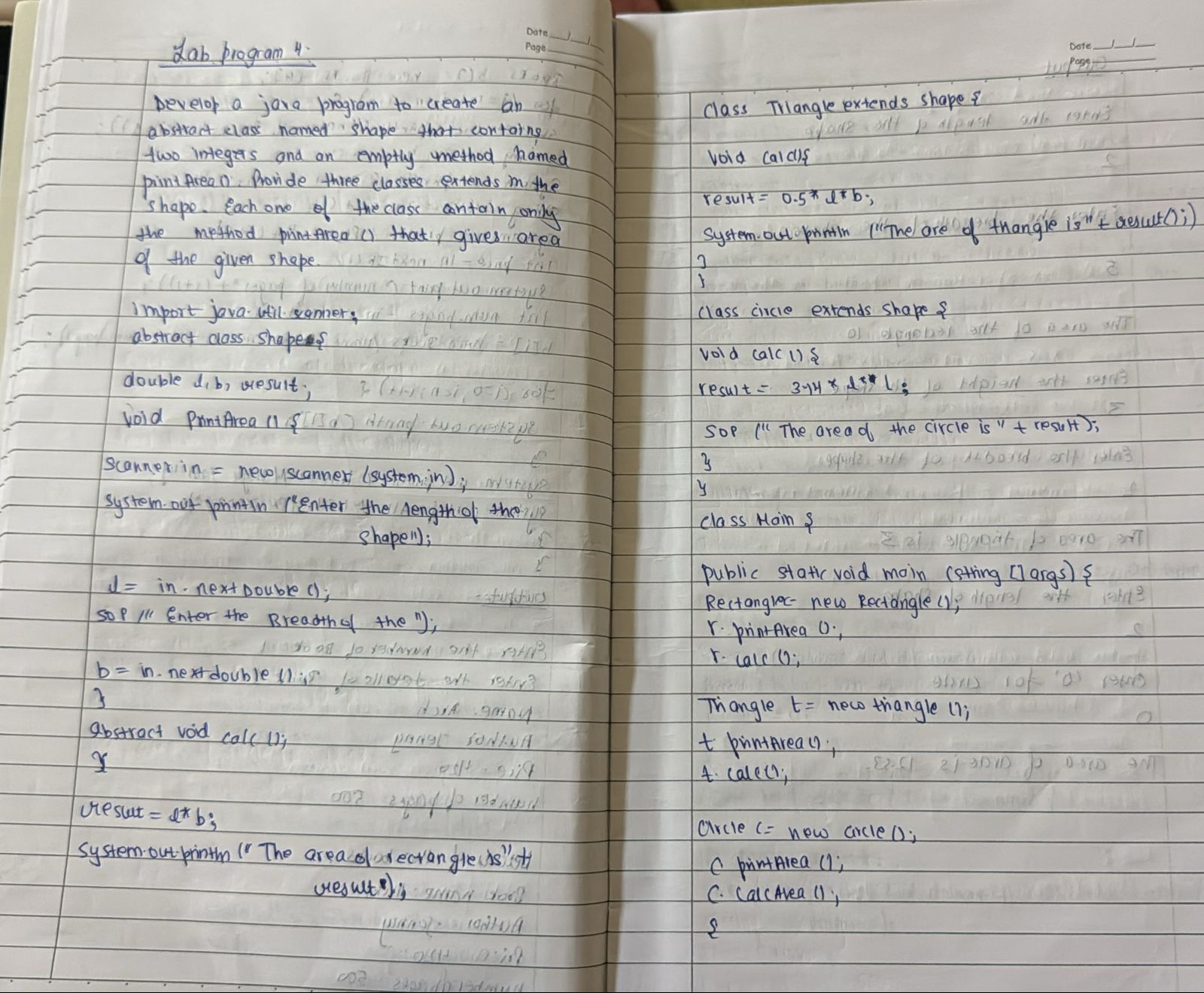
System.out.println("Name: MANYA JAIN");

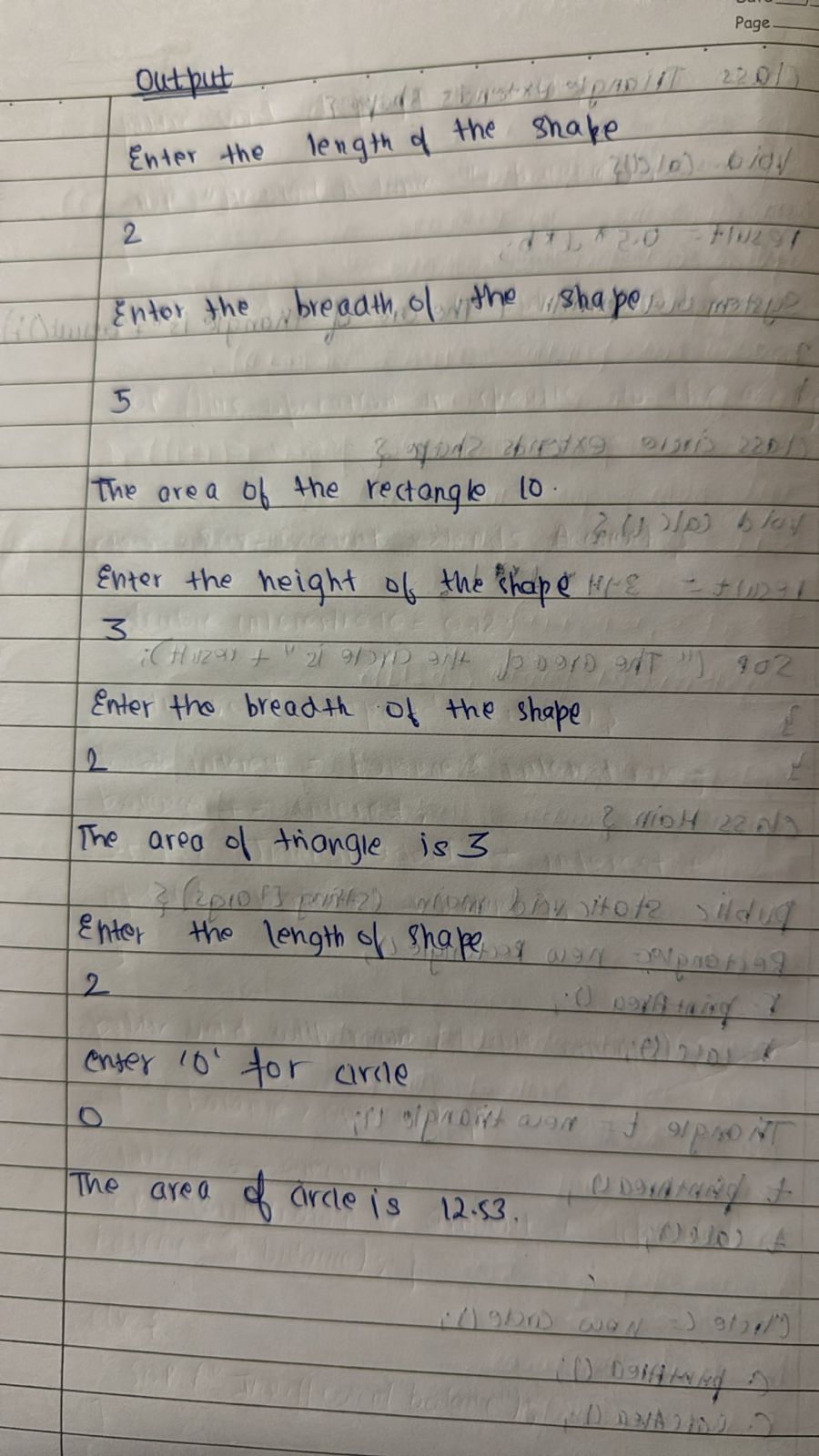
System.out.println("USN: 1BM23CS371);

}

}







import java.util.Scanner;

abstract class Shape{

double l,b,result;

void PrintArea() {

Scanner in=new Scanner(System.in);

System.out.println("enter 1st dimension(length for rectangle, base for triangle, radius for circle):");

l=in.nextDouble();

System.out.println("enter 2nd dimension(breadth for rectangle, height for triangle, zero for circle):");

b=in.nextDouble();

}

abstract void Calc();

}

class Rectangle extends Shape{

void Calc() {

result=l\*b;

System.out.println(" area of rectangle is=" + result);

}

}

class Triangle extends Shape{

void Calc() {

result=0.5\*l\*b;

System.out.println(" area of triangle is=" + result);

}

}

class Circle extends Shape{

void Calc() {

result=3.14\*l\*l;

System.out.println(" area of circle is=" + result);

}

}

class Main\_Shape{

public static void main(String args[]) {

Rectangle r=new Rectangle();

r.PrintArea();

r.Calc();

Triangle t=new Triangle();

t.PrintArea();

t.Calc();

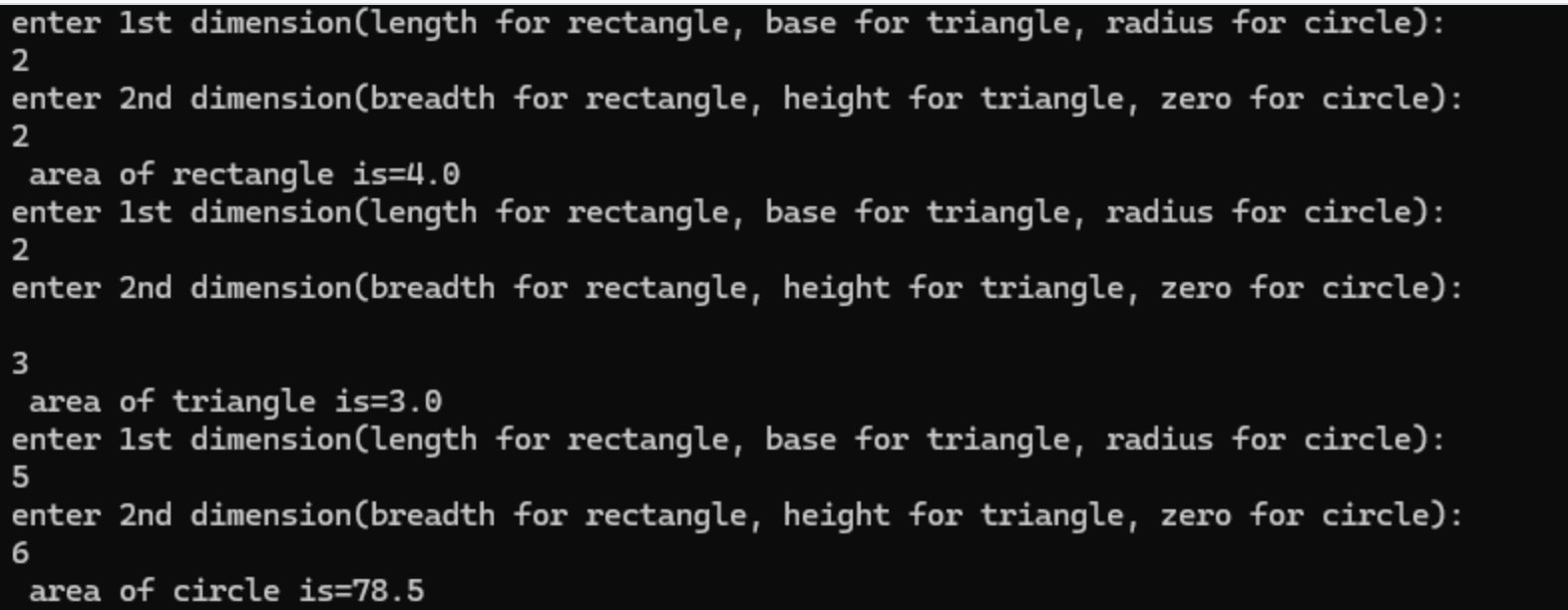
Circle c=new Circle();

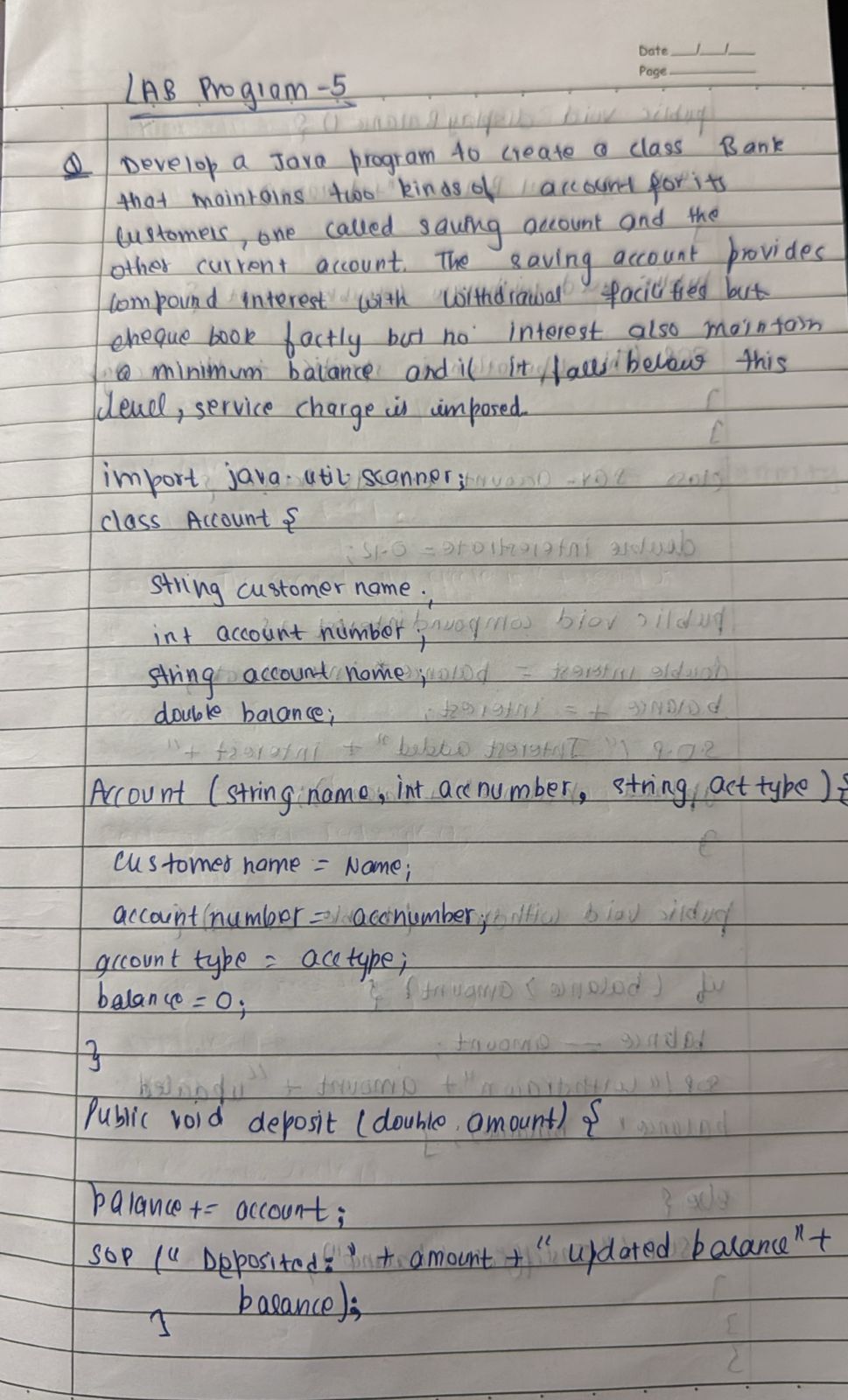
c.PrintArea();

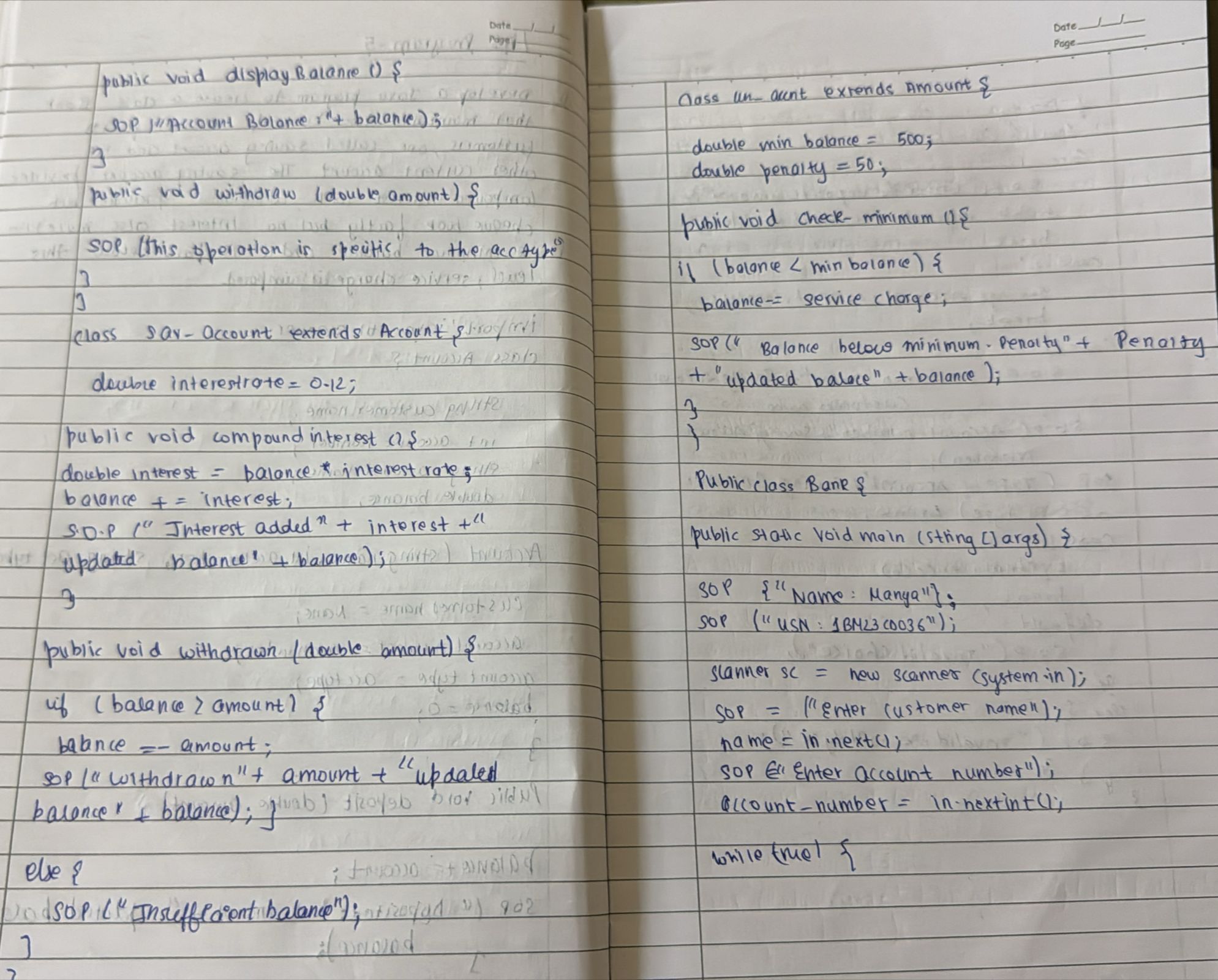
c.Calc();

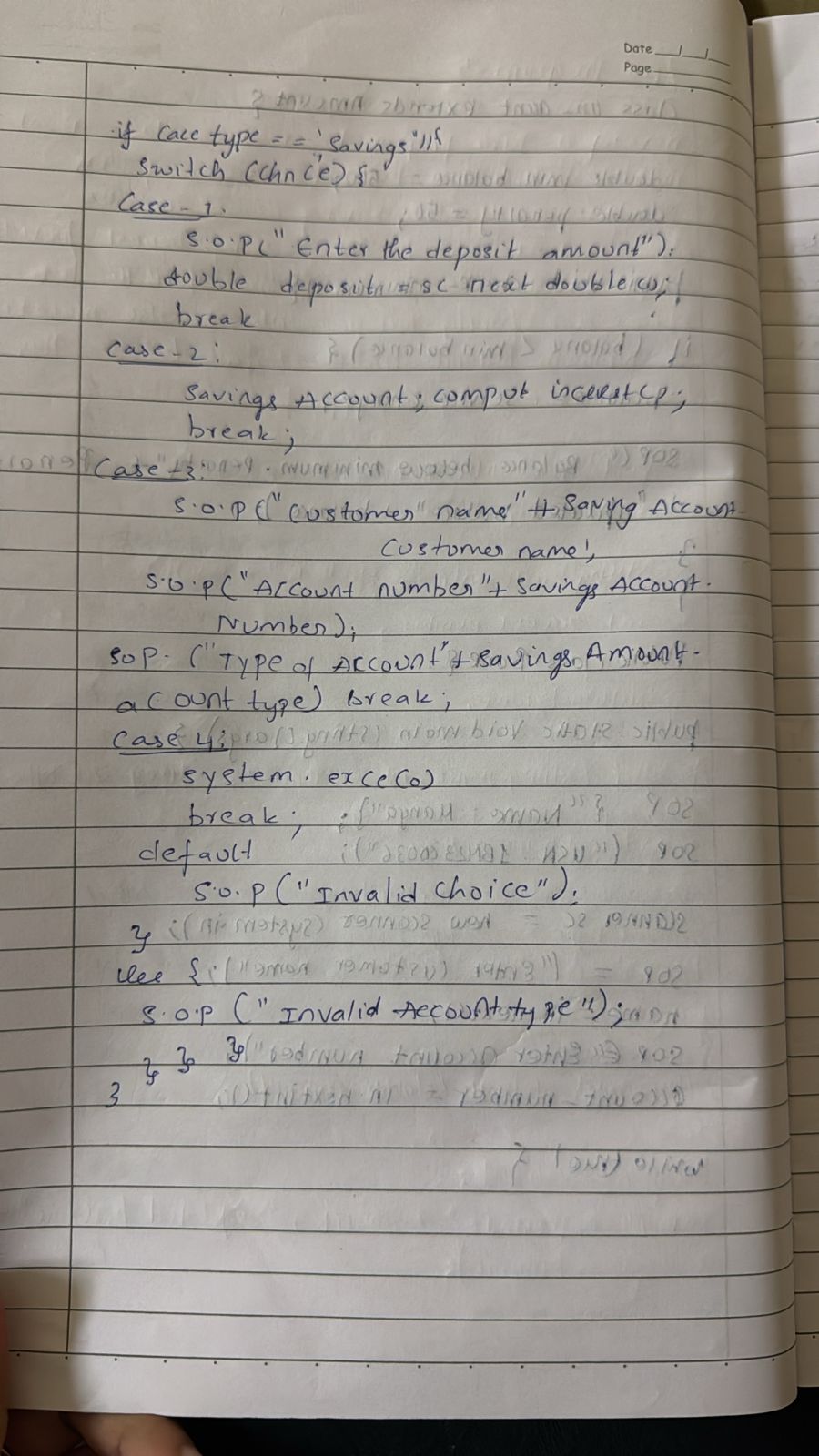
}

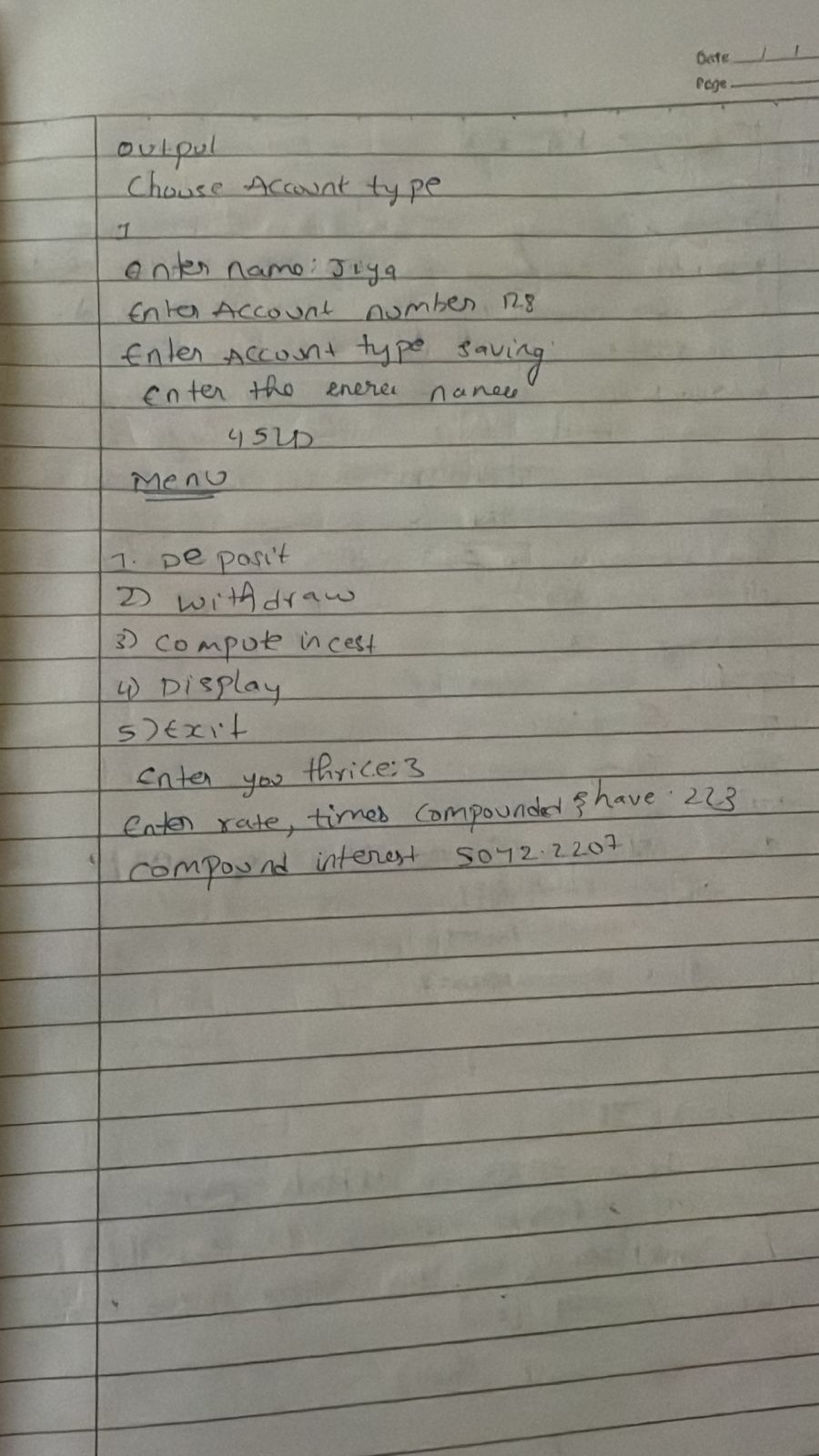
}











import java.util.Scanner;

class Account {

String customerName;

int accountNumber;

String accountType;

double balance;

Account(String name, int accNumber, String accType) {

customerName = name;

accountNumber = accNumber;

accountType = accType;

balance = 0;

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount + ". Updated balance: " + balance);

}

public void displayBalance() {

System.out.println("Account Balance: " + balance);

}

public void withdraw(double amount) {

System.out.println("This operation is specific to account type.");

}

}

class SavAccount extends Account {

double interestRate = 0.04; // 4% annual interest rate

SavAccount(String name, int accNumber) {

super(name, accNumber, "Savings");

}

public void computeInterest() {

double interest = balance \* interestRate;

balance += interest;

System.out.println("Interest added: " + interest + ". Updated balance: " + balance);

}

@Override

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);

} else {

System.out.println("Insufficient balance.");

}

}

}

class CurAccount extends Account {

double minBalance = 500.0;

double serviceCharge = 50.0;

CurAccount(String name, int accNumber) {

super(name, accNumber, "Current");

}

public void checkMinBalance() {

if (balance < minBalance) {

balance -= serviceCharge;

System.out.println("Balance below minimum. Service charge imposed: " + serviceCharge + ". Updated balance: " + balance);

}

}

@Override

public void withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);

checkMinBalance();

} else {

System.out.println("Insufficient balance.");

}

}

}

public class Bank {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter customer name:");

String name=sc.next();

System.out.println("Enter account number:");

int accountnumber=sc.nextInt();

SavAccount savingsAccount = new SavAccount(name, accountnumber);

System.out.println("Enter customer name:");

String name1=sc.next();

System.out.println("Enter account number:");

int accountnumber1=sc.nextInt();

CurAccount currentAccount = new CurAccount(name1, accountnumber1);

while (true) {

System.out.println("manya jain");

System.out.println("1BM23CS371");

System.out.println("\n-----MENU-----");

System.out.println("1. Deposit\n2. Withdraw\n3. Compute Interest for Savings Account\n4. Display Account Details\n5. Exit");

System.out.print("Enter your choice: ");

int choice = sc.nextInt();

System.out.print("Enter the type of account (saving/current): ");

String accType = sc.next();

if (accType.equals("saving")) {

switch (choice) {

case 1:

System.out.print("Enter the deposit amount: ");

double depositAmount = sc.nextDouble();

savingsAccount.deposit(depositAmount);

break;

case 2:

System.out.print("Enter the withdrawal amount: ");

double withdrawalAmount = sc.nextDouble();

savingsAccount.withdraw(withdrawalAmount);

break;

case 3:

savingsAccount.computeInterest();

break;

case 4:

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

System.out.println("Account number: " + savingsAccount.accountNumber);

System.out.println("Type of Account: " + savingsAccount.accountType);

savingsAccount.displayBalance();

break;

case 5:

System.exit(0);

break;

default:

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

}

} else if (accType.equals("current")) {

switch (choice) {

case 1:

System.out.print("Enter the deposit amount: ");

double depositAmount = sc.nextDouble();

currentAccount.deposit(depositAmount);

break;

case 2:

System.out.print("Enter the withdrawal amount: ");

double withdrawalAmount = sc.nextDouble();

currentAccount.withdraw(withdrawalAmount);

break;

case 3:

System.out.println("Current accounts do not earn interest.");

break;

case 4:

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

System.out.println("Account number: " + currentAccount.accountNumber);

System.out.println("Type of Account: " + currentAccount.accountType);

currentAccount.displayBalance();

break;

case 5:

System.exit(0);

break;

default:

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

}

} else {

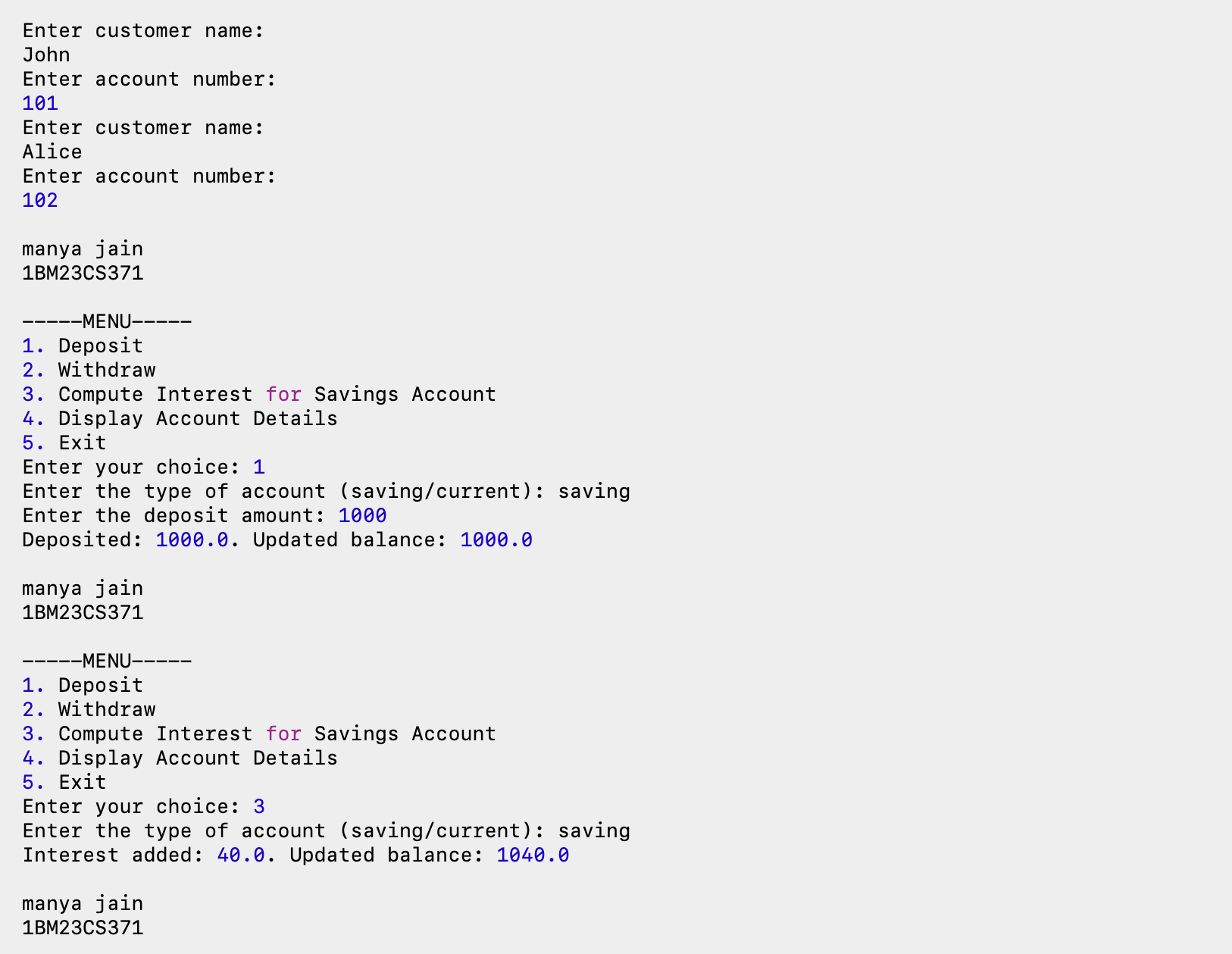
System.out.println("Invalid account type.");

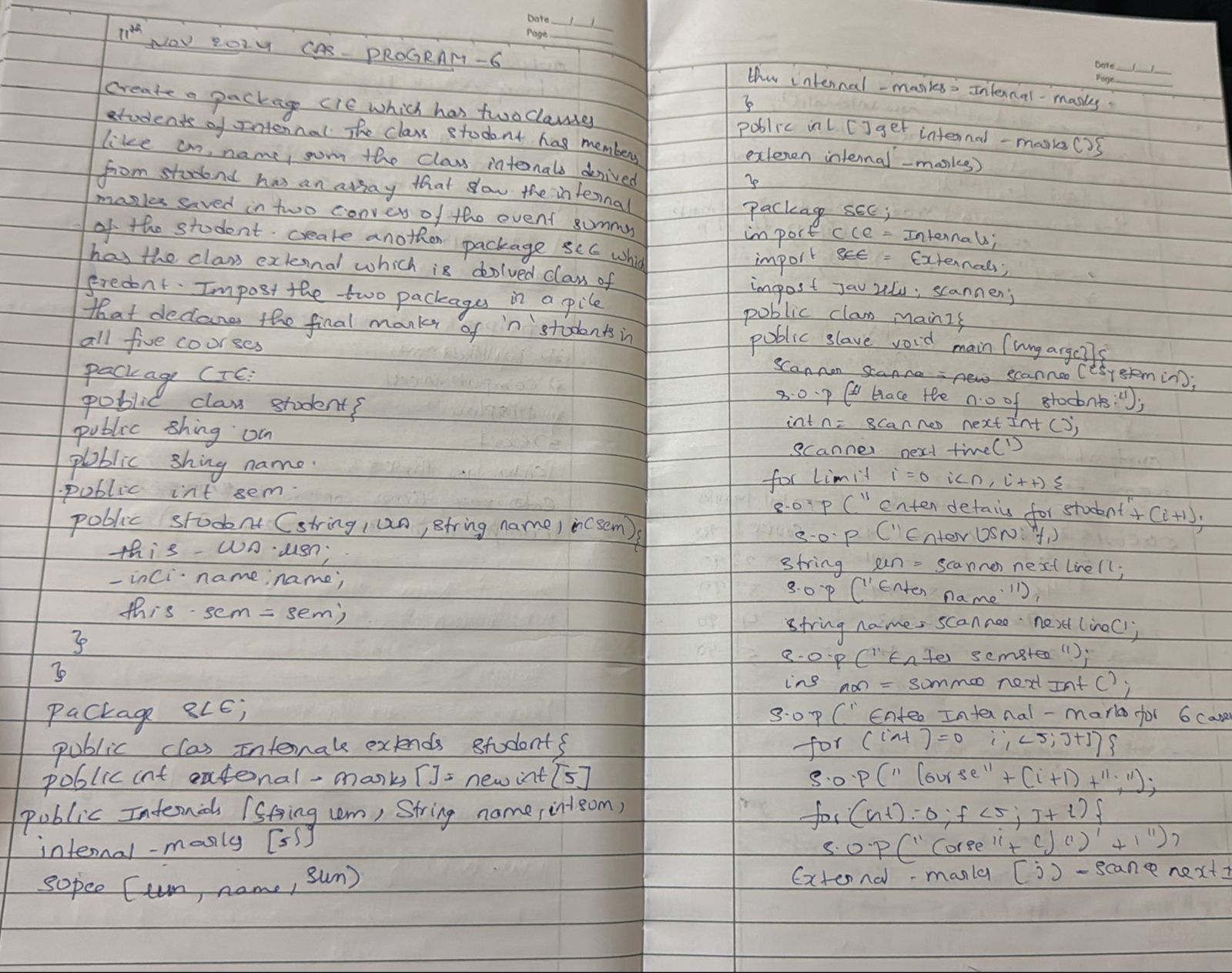
}

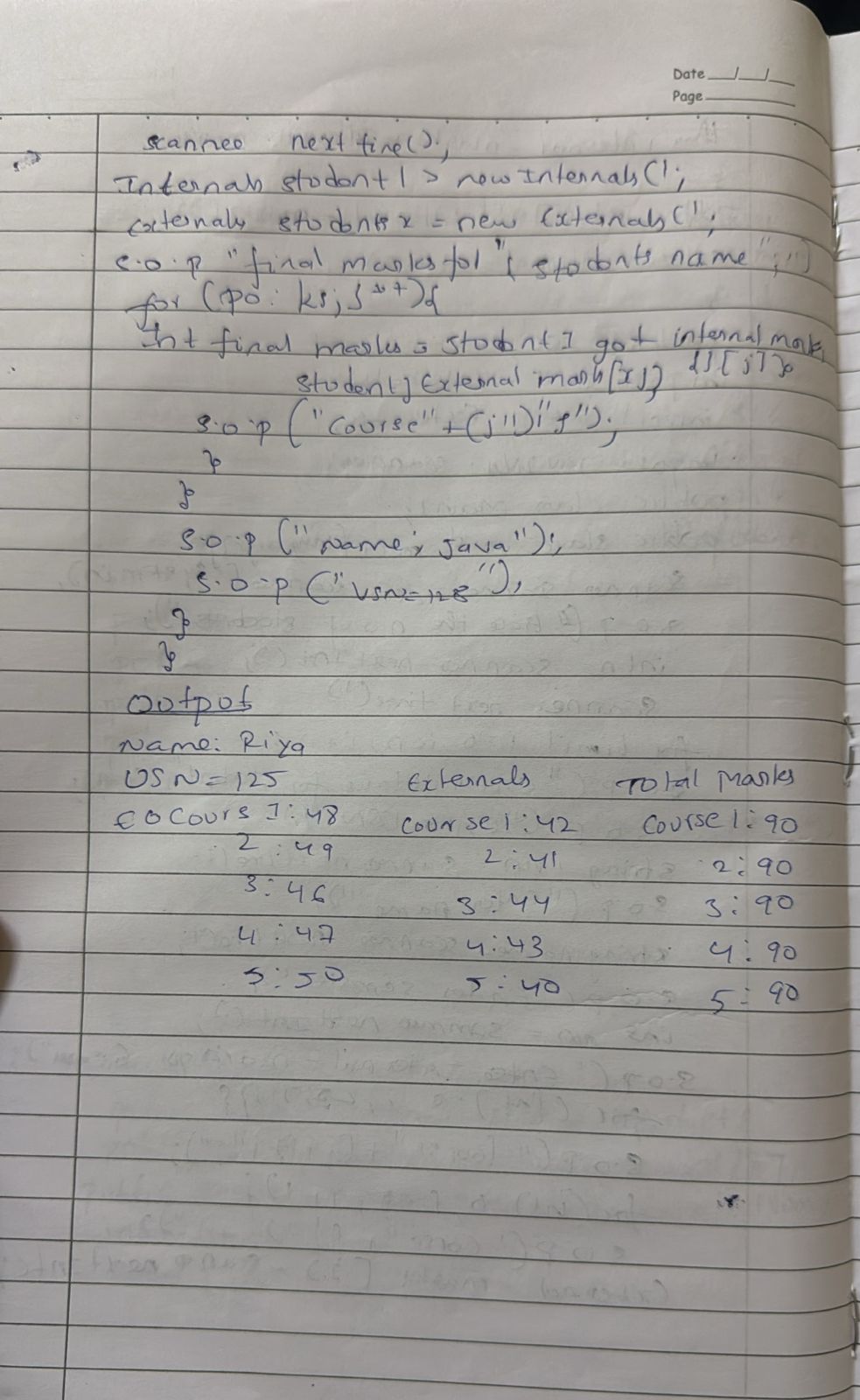
}

}

}







import java.util.Scanner;

public class Student

{

public String usn;

public String name;

public int sem;

public Student( String usn,

String name, int sem)

{

this.usn=usn;

this.name=name;

this.sem=sem;

}

}

package CIE;

public class Internals extends Student

{

int internalmarks[]=new int[5];

public Internals( String usn,String name,int sem,int internalmarks[]){

super(usn,name,sem);

this.internalmarks=internalmarks;

}

public int[] getInternalmarks(){

return internalmarks;}

}

package SEE;

import CIE.Student;

public class External extends Student

{

int[] externalmarks=new int[5];

public External ( String usn,

String name,

int sem,int externalmarks[])

{

super(usn,name,sem);

this.externalmarks=externalmarks;

}

public int[] getExternalMarks(){

return externalmarks;

}

}

import CIE.Internals;

import SEE.External;

import java.util.Scanner;

public class Main1{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

scanner.nextLine();

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

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

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

String usn = scanner.nextLine();

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

String name = scanner.nextLine();

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

int sem = scanner.nextInt();

scanner.nextLine();

int[] internalMarks = new int[5];

System.out.println("Enter internal marks for 5 courses: ");

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

System.out.print("Course " + (j + 1) + ": ");

internalMarks[j] = scanner.nextInt();

}

scanner.nextLine();

int[] externalMarks = new int[5];

System.out.println("Enter external marks for 5 courses: ");

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

System.out.print("Course " + (j + 1) + ": ");

externalMarks[j] = scanner.nextInt();

}

scanner.nextLine();

Internals internStudent = new Internals(usn, name, sem, internalMarks);

External externalStudent = new External(usn, name, sem, externalMarks);

System.out.println("\nFinal marks for " + internStudent.name + ":");

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

int finalMark = internStudent.getInternalmarks()[j] + externalMarks[j];

System.out.println("Course " + (j + 1) + ": " + finalMark);

}

System.out.println("\nFinal marks for " + externalStudent.name + ":");

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

int finalMark = externalStudent.getExternalMarks()[j] + internalMarks[j];

System.out.println("Course " + (j + 1) + ": " + finalMark);

}

}

scanner.close();

}

}







package Lab;

import java.util.Scanner;

import java.util.concurrent.ExecutionException;

class AgeException extends java.lang.Exception{

public AgeException(String str){

super(str);

}

}

class Father{

Scanner sc = new Scanner(System.in);

int fatherAge;

public Father(){

try {

System.out.println("Enter the age of Father here: ");

fatherAge = sc.nextInt();

if (fatherAge < 0) throw new AgeException("Father Age Can't be negative: ");

}

catch(AgeException e){

System.out.println(e);

System.out.println("Re-Enter Father's age here: ");

fatherAge = sc.nextInt();

}

}

public void fatherDetails(){

System.out.println("Father's age is: " + fatherAge);

}

}

class Son extends Father{

Scanner sc = new Scanner(System.in);

int sonAge;

public Son() {

super();

try{

System.out.println("Enter son age: ");

sonAge = sc.nextInt();

if(sonAge>fatherAge) throw new AgeException("Son age can't be greater than Father's age");

}

catch (AgeException e){

System.out.println(e);

System.out.println("Re-Enter Son's age: ");

sonAge = sc.nextInt();

}

}

public void sonDetails(){

System.out.println("Son's age is: " + sonAge);

}

}

public class Exception {

public static void main(String[] args) {

Son sonObj = new Son();

sonObj.sonDetails();

sonObj.fatherDetails();

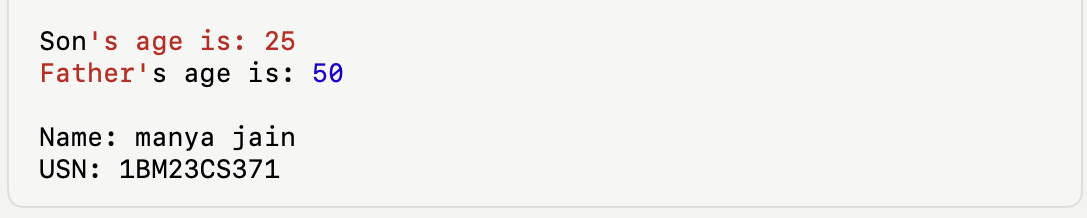
System.out.println();

System.out.println("Name: manya jain");

System.out.println("USN: 1BM23CS371");

}

}



class BMS extends Thread{

public void run(){

while (true){

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

try{

Thread.sleep(10000);

}

catch (InterruptedException e){}

}

}

}

class CSE extends Thread{

public void run(){

while (true){

System.out.println("CSE");

try{

Thread.sleep(2000);

}

catch (InterruptedException e){}

}

}

}

public class ThreadingProgram {

public static void main(String[] args) {

BMS bms = new BMS();

CSE cse = new CSE();

System.out.println("Name:manya jain");

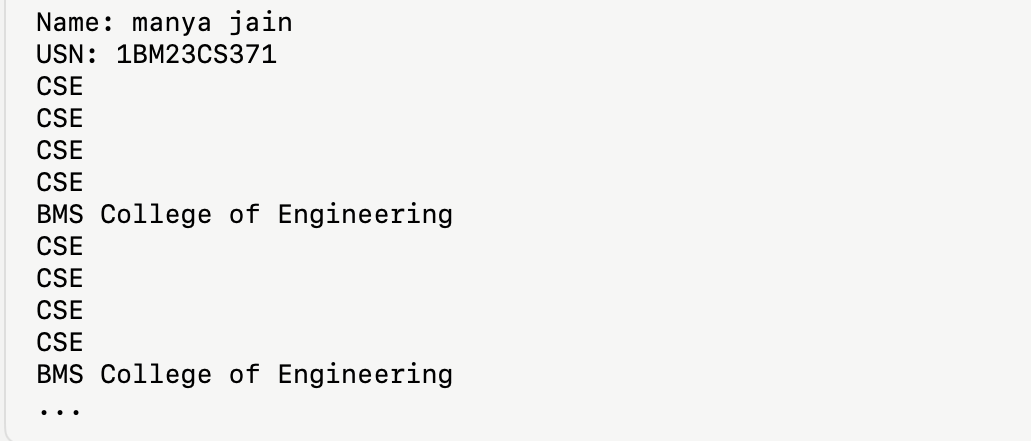
System.out.println("USN: 1BM23CS371");

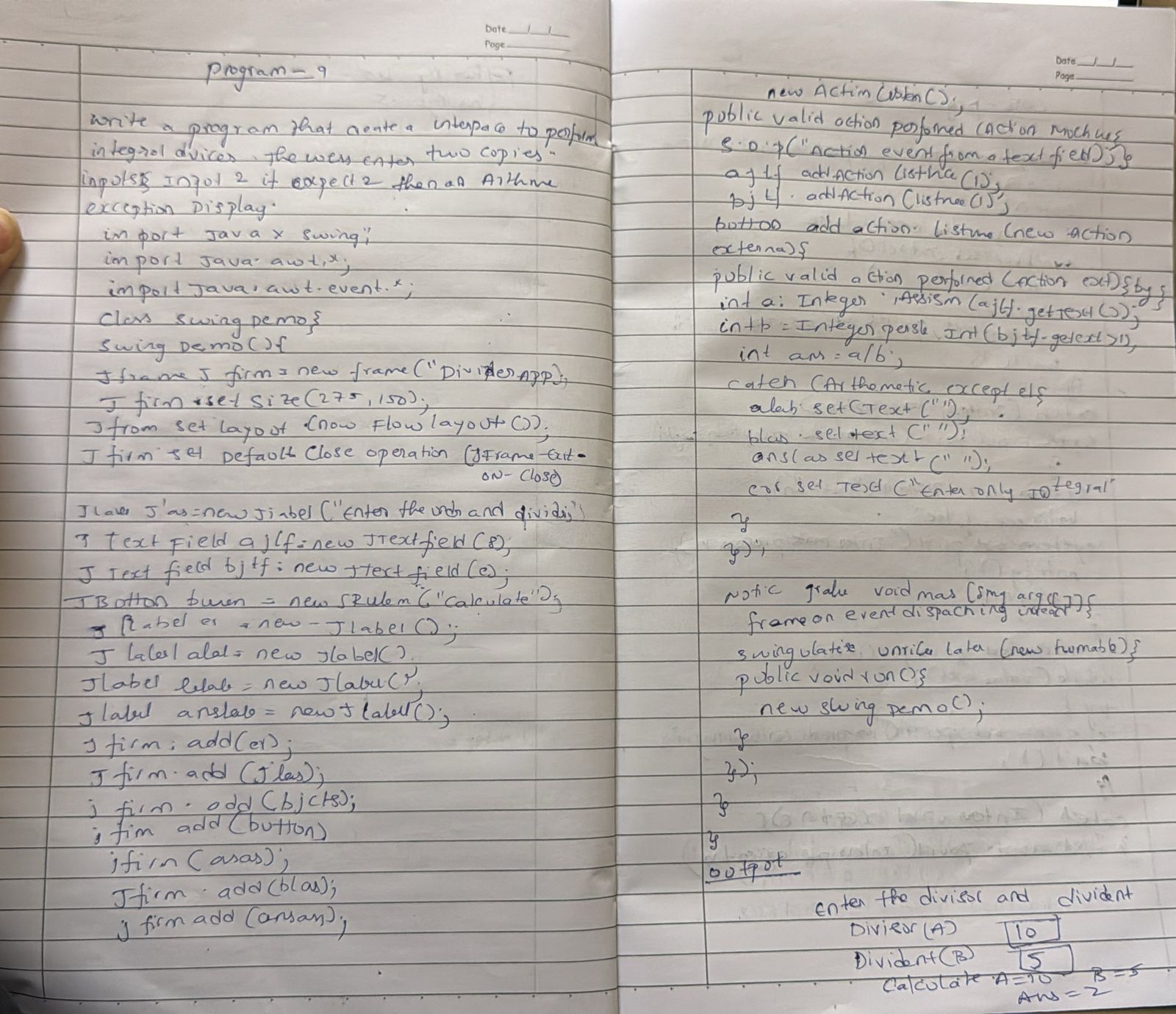
bms.start();

cse.start();

}

}





package Lab;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class SwingDemo {

SwingDemo() {

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

jfrm.setSize(275, 200);

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[]) {

System.out.println("Name: manya jain");

System.out.println("USN: 1BM23CS371");

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new SwingDemo();

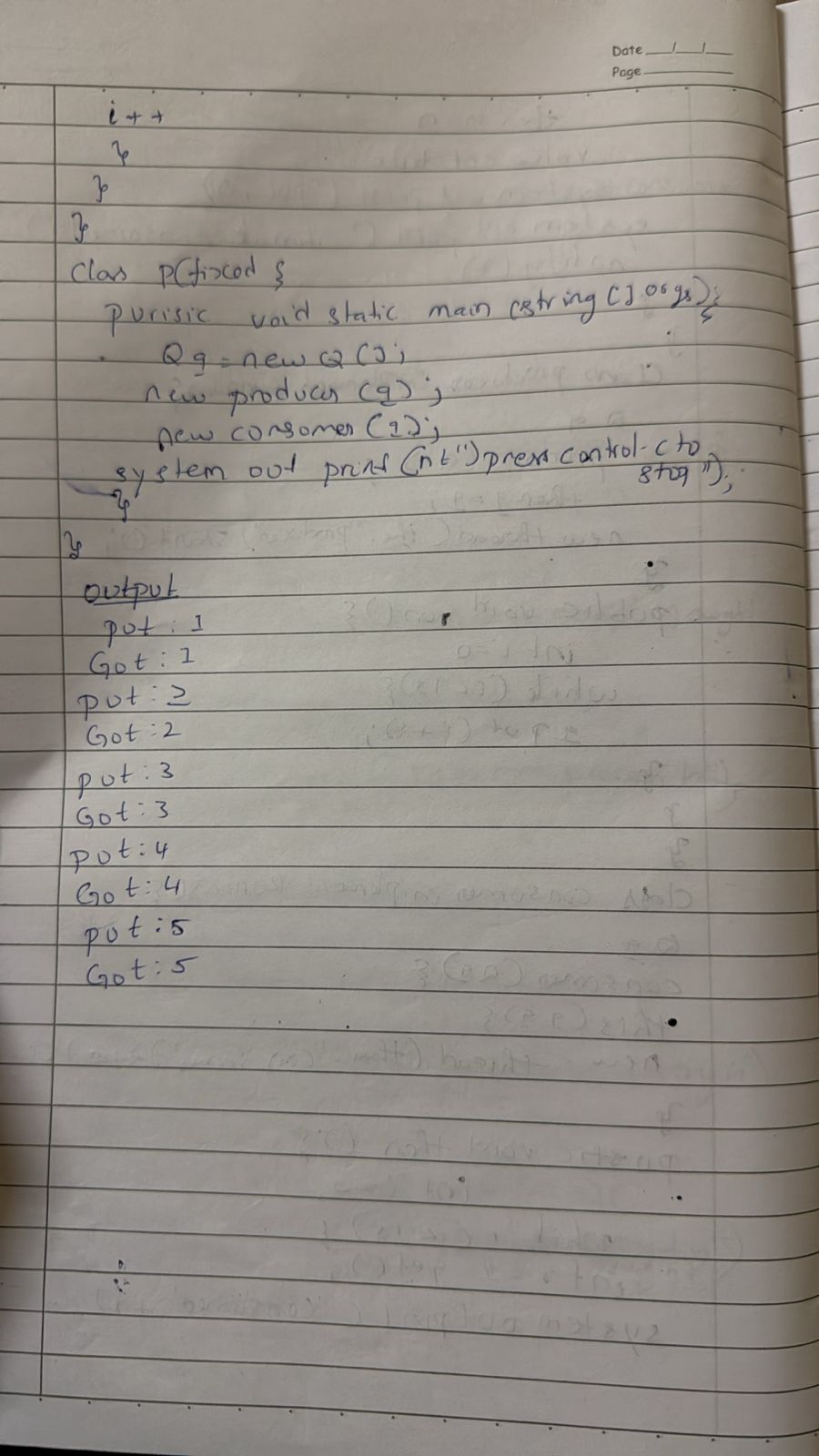
}

});

}

}





class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet) {

try {

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

wait();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

}

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

valueSet = false;

System.out.println("\nIntimate Producer\n");

notify();

return n;

}

synchronized void put(int n) {

while (valueSet) {

try {

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

wait();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");

}

}

this.n = n;

valueSet = true;

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

System.out.println("\nIntimate Consumer\n");

notify();

}

}

class Producer implements Runnable {

Q q;

Producer(Q q) {

this.q = q;

new Thread(this, "Producer").start();

}

public void run() {

int i = 0;

while (i < 15) {

q.put(i++);

}

}

}

class Consumer implements Runnable {

Q q;

Consumer(Q q) {

this.q = q;

new Thread(this, "Consumer").start();

}

public void run() {

int i = 0;

while (i < 15) {

int r = q.get();

System.out.println("Consumed: " + r);

i++;

}

}

}

public class PCFixed {

public static void main(String args[]) {

Q q = new Q();

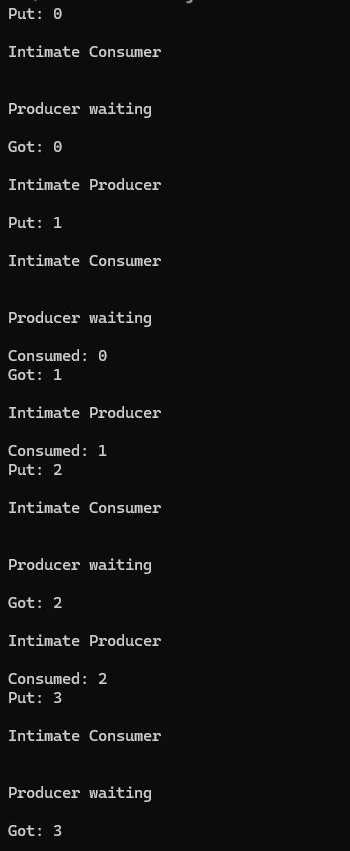
new Producer(q);

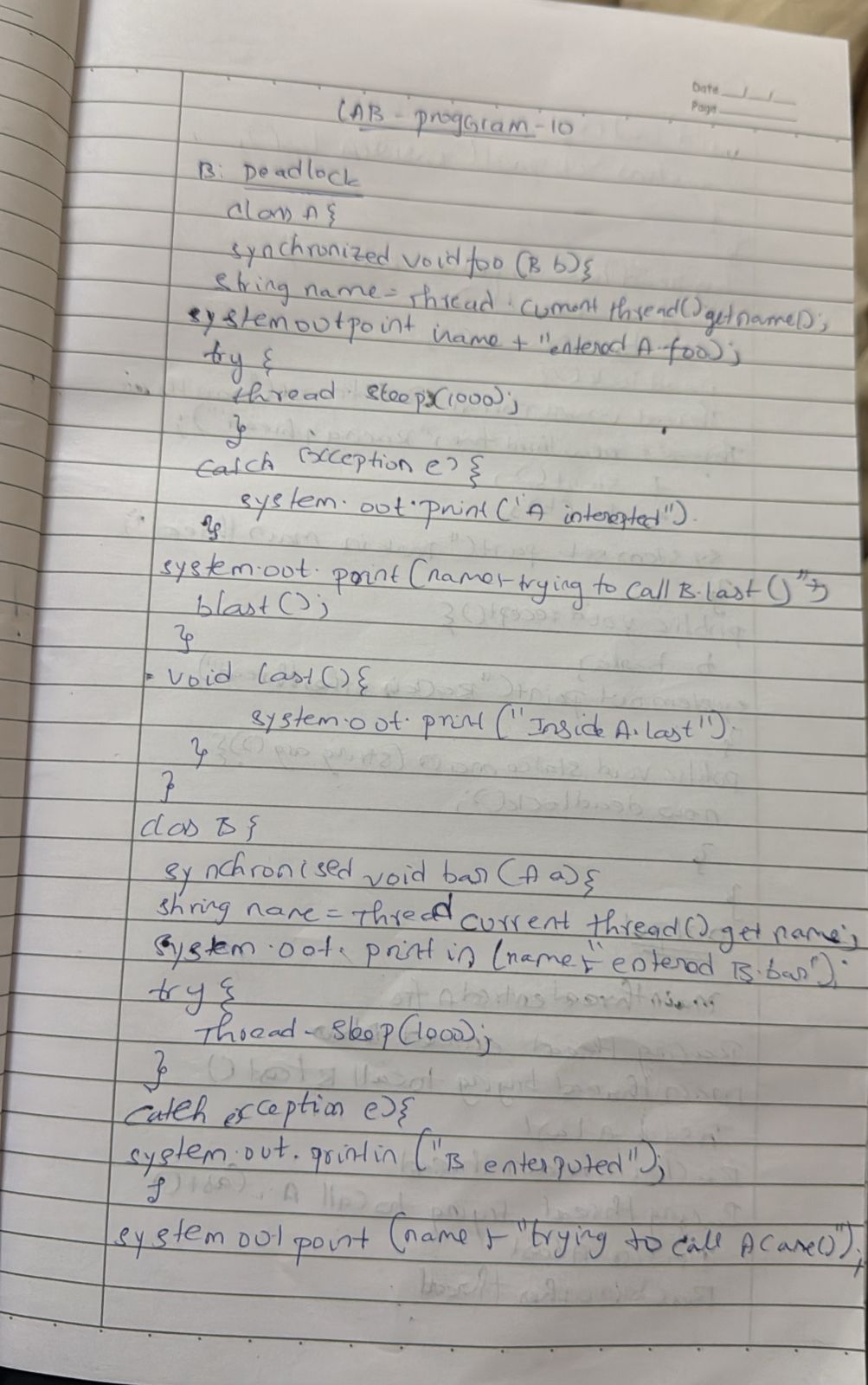
new Consumer(q);

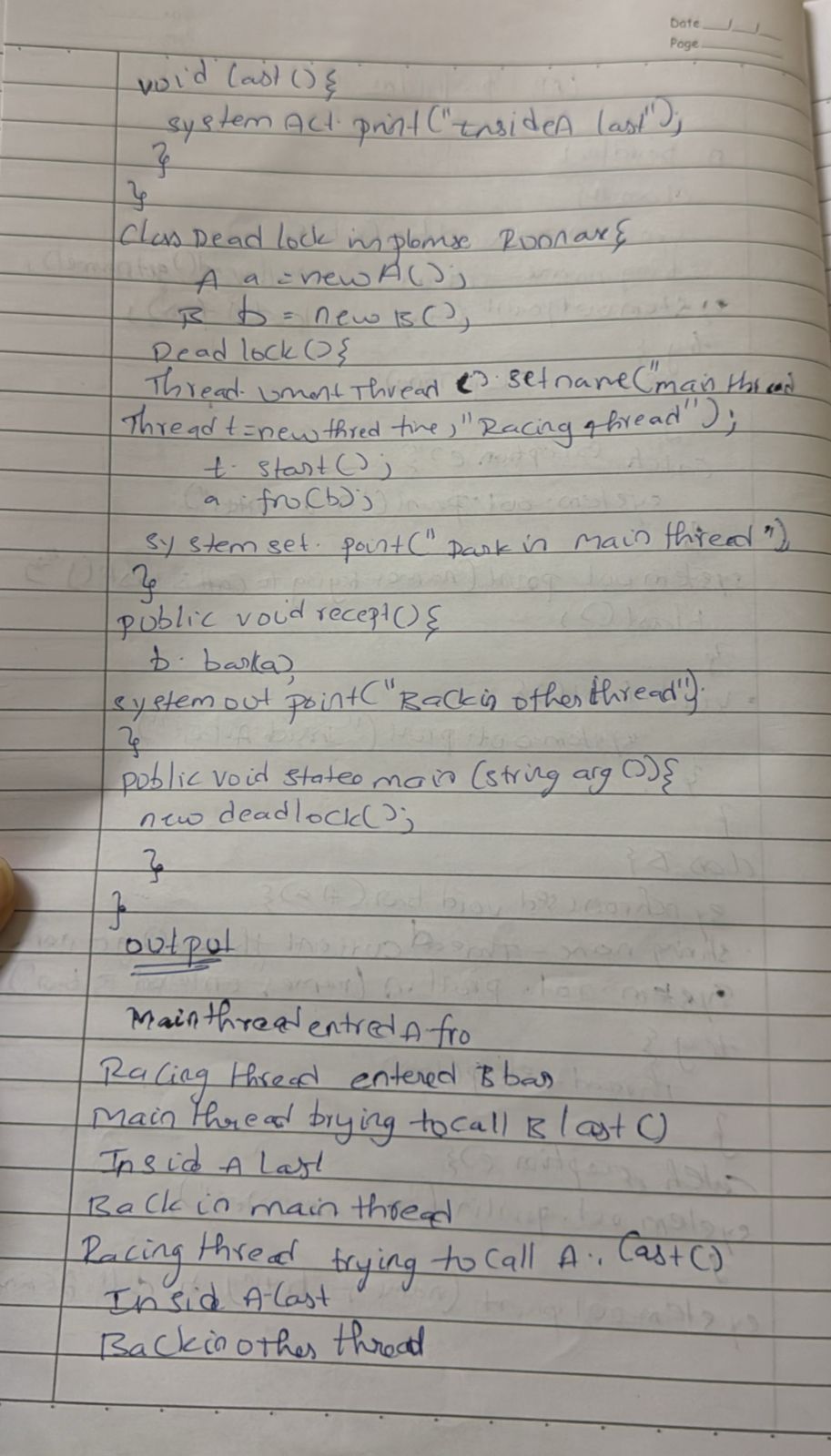
System.out.println("Press Control-C to stop.");

}

}







class A {

synchronized void foo(B b) {

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

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

try {

Thread.sleep(1000);

} catch (Exception e) {

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

}

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

b.last();

}

void last() {

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

}

}

class B {

synchronized void bar(A a) {

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

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

try {

Thread.sleep(1000);

} catch (Exception e) {

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

}

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

a.last();

}

void last() {

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

}

}

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock() {

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

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

t.start();

a.foo(b);

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

}

public void run() {

b.bar(a);

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

}

public static void main(String args[]) {

new Deadlock();

}

}