VISVESVARAYA TECHNOLOGICAL UNIVERSITY, JNANASANGAMA, BELGAUM - 590014, KARNATAKA

LABORATORY RECORD

ON

Object Oriented Java Programming

(23CS3PCOOJ) ***Submitted by***

Mohammed Shuraih (1BM22CS157)

In partial fulfilment 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
December-2022 to April-2023

INDEX

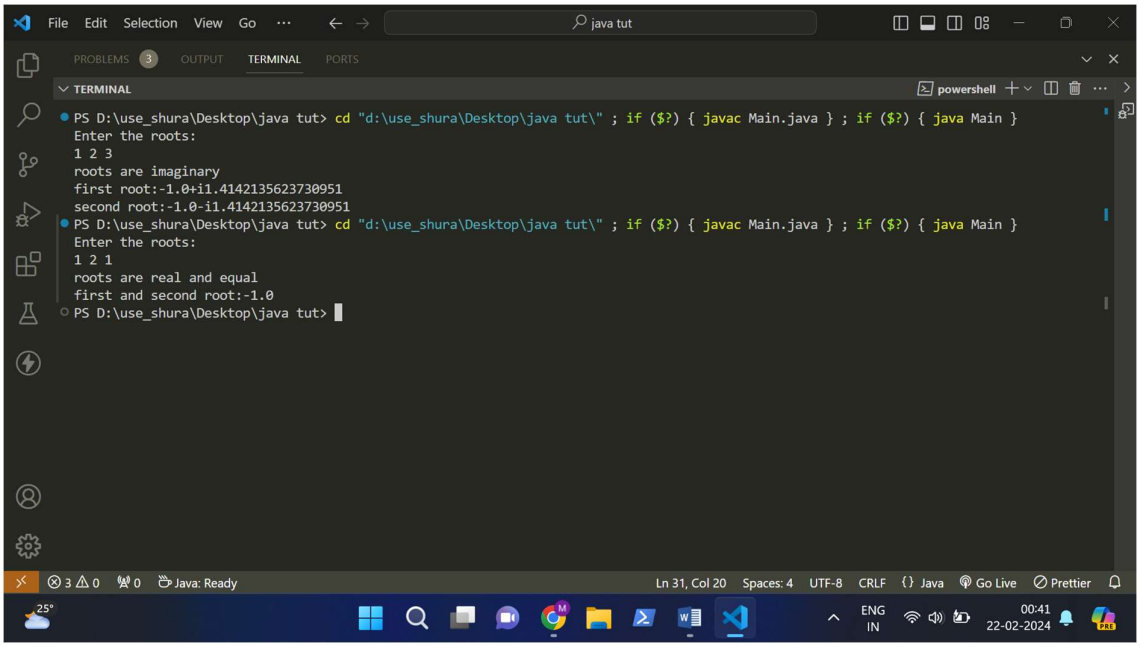
LABORATORY PROGRAM - 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

class Main{
    public static void main(String args[])
    {
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter the roots:");
        Double a = s1.nextDouble();
        Double b = s1.nextDouble();
        Double c = s1.nextDouble();

        Double r1,r2,d = b*b - 4*a*c;
        if(d>0){
            System.out.println("Roots are real and distinct");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2 =(-b-Math.sqrt(d))/(2*a);
            System.out.println("first root:"+r1+"\nSecond root:"+r2);
        }
        else if(d==0){
            System.out.println("roots are real and equal");
            r1 = (-b)/(2*a);
            System.out.println("first and second root:"+r1);
        }
        else{
            System.out.println("roots are imaginary");
            r1=(-b)/(2*a);
            r2 = Math.sqrt(Math.abs(d))/(2*a);
            System.out.println("first root:"+r1+"+i"+r2);
```

```
            System.out.println("second root:"+r1+"-i"+r2);
        }
        s1.close();
    }
}
```

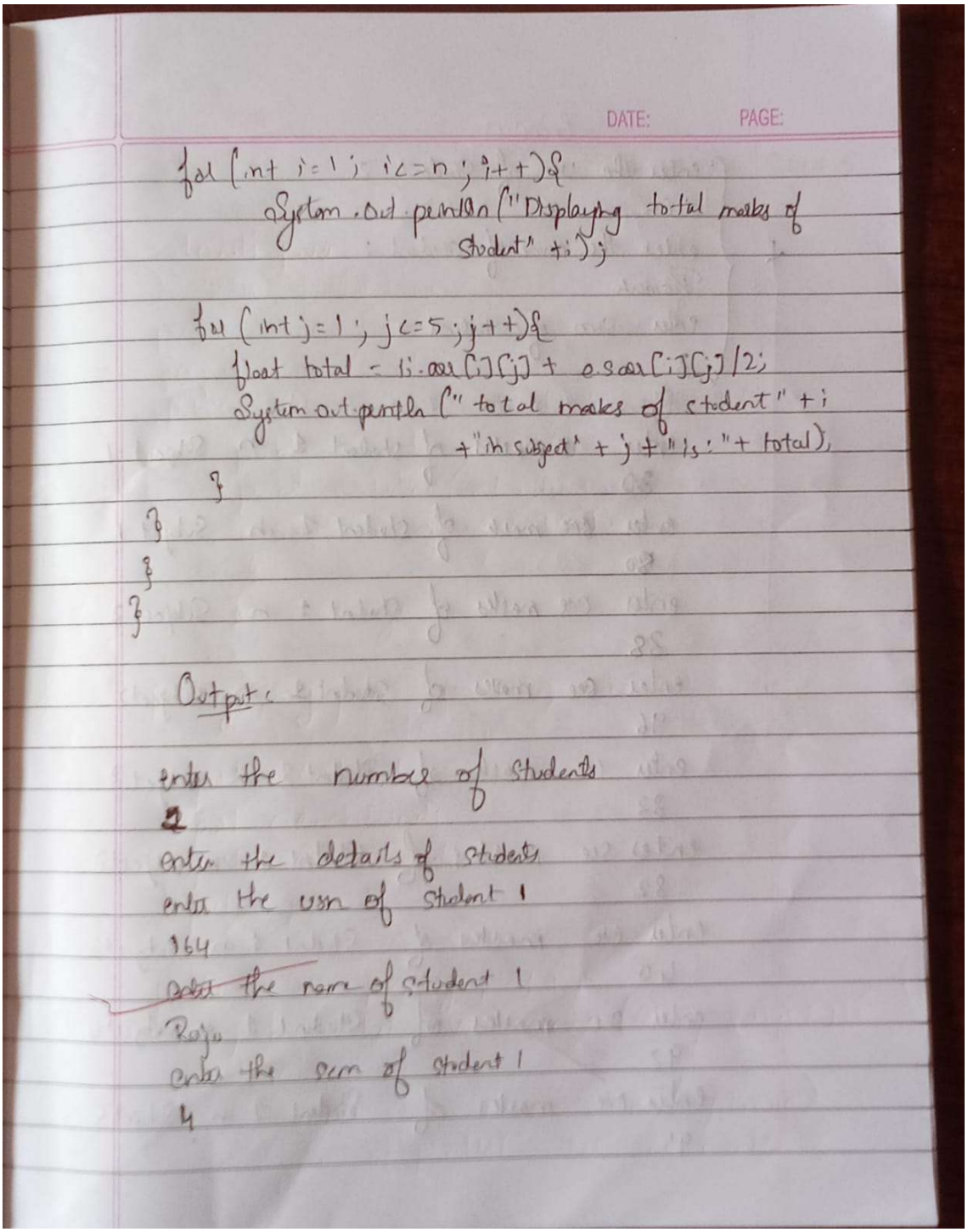
OUTPUT



LAB-1 DATE: PAGE:

Program to find roots of quadratic equation.

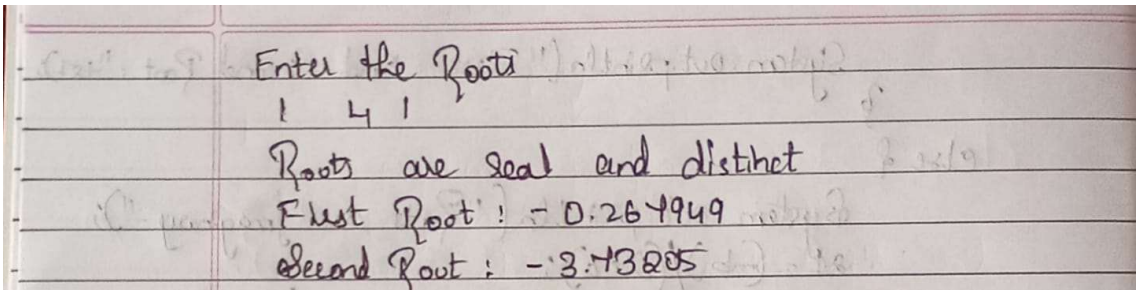
```
import java.util.Scanner;
class Quadratic{
    public static void main(String Args[]){
    Scanner s1 = new Scanner(System.in);
    System.out.println("Enter the Roots :");
    Double a = s1.nextDouble();
    Double b = s1.nextDouble();
    Double c = s1.nextDouble();

    Double x1,x2, d = b*b - 4*a*c;
    if(d>0){
        System.out.println("Roots are read
                            and distinct");
    x1 = (-b+ Math.sqrt(d)) / (2*a);
    x2 = (-b * - Math.sqrt(d)) / (2*a);
    System.out.println("First Root : " +
                x1 + "\n Second Root : " + x2);
    }
else if(d==0){
    System.out.println("Roots are real
                and Equal.");
    x1 = (-b)/(2*a);
```



```
for (int i=1; i<=n; i++){
    System.out.println("Displaying total marks of
                        student" +i);

for (int j=1; j<=5; j++){
    float total = s1.scor[i][j] + s.scor[i][j]/2;
    System.out.println("total marks of student" +i
                       +"in subject" +j+"is:"+ total);
    }
}
}
}
```

Output:

```
enter the number of students
2
enter the details of students
enter the usn of student 1
164
enter the name of student 1
Raju
enter the scm of student 1
4
```


Enter the Roots
1 4 1
Roots are real and distinct
First Root : -0.264949
Second Root : -3.73505

LABORATORY PROGRAM - 2

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

```
import java.util.Scanner;

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

   public void acceptDetails() {
      Scanner sc = new Scanner(System.in);
System.out.println("Enter USN: ");         usn
= sc.nextLine();
      System.out.println("Enter Name: ");
      name = sc.nextLine();
      System.out.println("Enter number of subjects: ");
int n = sc.nextInt();         credits = new int[n];
marks = new int[n];         for (int i = 0; i < n; i++) {
         System.out.println("Enter credits for subject " + (i + 1) + ": ");
credits[i] = sc.nextInt();
         System.out.println("Enter marks for subject " + (i + 1) + ": ");
marks[i] = sc.nextInt();
      }
   }

   public void displayDetails() {
System.out.println("USN: " + usn);
      System.out.println("Name: " + name);
System.out.println("Marks: ");         for (int
i = 0; i < marks.length; i++) {
         System.out.println("Subject " + (i + 1) + ": " + marks[i]);
      }
      System.out.println("Credits: ");
for (int i = 0; i < credits.length; i++) {
         System.out.println("Subject " + (i + 1) + ": " + credits[i]);
```

```
        }
    }

    public double calculateSGPA() {
        double totalGrade = 0;          int totalCredit = 0;
for (int i = 0; i < credits.length; i++) {
totalGrade += getGrade(marks[i]) * credits[i];
            totalCredit += credits[i];
        }
        return totalGrade / totalCredit;
    }

    private double getGrade(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 >= 40) {
            return 5;
} else {
return 0;
        }
    }

    public static void main(String[] args) {
Student student = new Student();
student.acceptDetails();
student.displayDetails();
        System.out.println("SGPA: " + student.calculateSGPA());
    }
}
```

OUTPUT

```
D:\NotePad++\Java>javac Student.java

D:\NotePad++\Java>java Student
Enter USN:
L24
Enter Name:
Royce
Enter number of subjects:
3
Enter credits for subject 1:
4
Enter marks for subject 1:
97
Enter credits for subject 2:
3
Enter marks for subject 2:
98
Enter credits for subject 3:
2
Enter marks for subject 3:
99
USN: L24
Name: Royce
Marks:
Subject 1: 97
Subject 2: 98
Subject 3: 99
Credits:
Subject 1: 4
Subject 2: 3
Subject 3: 2
SGPA: 10.0
```

DATE: PAGE:

LAB-2

```
import java.util.Scanner;
class Student
{
    String usn;
    String name;
    int [] marks;
    int [] credits;
    public void AcceptDetails(int n)
    {
        Scanner s1 = new Scanner(System.in);
        System.out.println("enter usn");
        usn = s1.next();
        System.out.println("enter name");
        name = s1.next();
        credits = new int[n];
        marks = new int[n];
        for(int i=1; i<=credits.length; i++)
        {
            System.out.println("enter credits for
                                Subjects" + i + ":");
            credits[i-1] = s1.nextInt();
            System.out.println("enter marks for
                                Subjects" + i + ":");
            marks[i-1] = s1.nextInt();
        }
    }
```

DATE: PAGE:

```
public void DisplayDetails ()
{
    System.out.println (usn);
    System.out.println (name);
    for (int i=1; i<=marks.length; i++)
    {
        System.out.println ("Subject-"+i+":"+
                            marks[i-1]);
    }
}

Public float SgpaCal()
{
    int totalCredits = 0;
    int gradePoints = 0;

    for (int i=1; i<marks.length; i++)
    {
        totalCredits += credits[i-1];
        int temp = gradePoint (marks[i-1]);
        gradePoints += credits[i-1] * temp;
    }
    return gradePoints / totalCredits;
}
```

DATE: PAGE:

```
public int gradePoint(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>40)
        return 5;
    else
        return 0;
}

Class Studentdetail
{
    public static void main(String args[])
    {
        Student s = new Student();
        s.AcceptDetails();
        s.DisplayDetails();
        System.out.println("SGPA:" + s.Sgpa());
    }
}
```

LABORATORY PROGRAM - 3

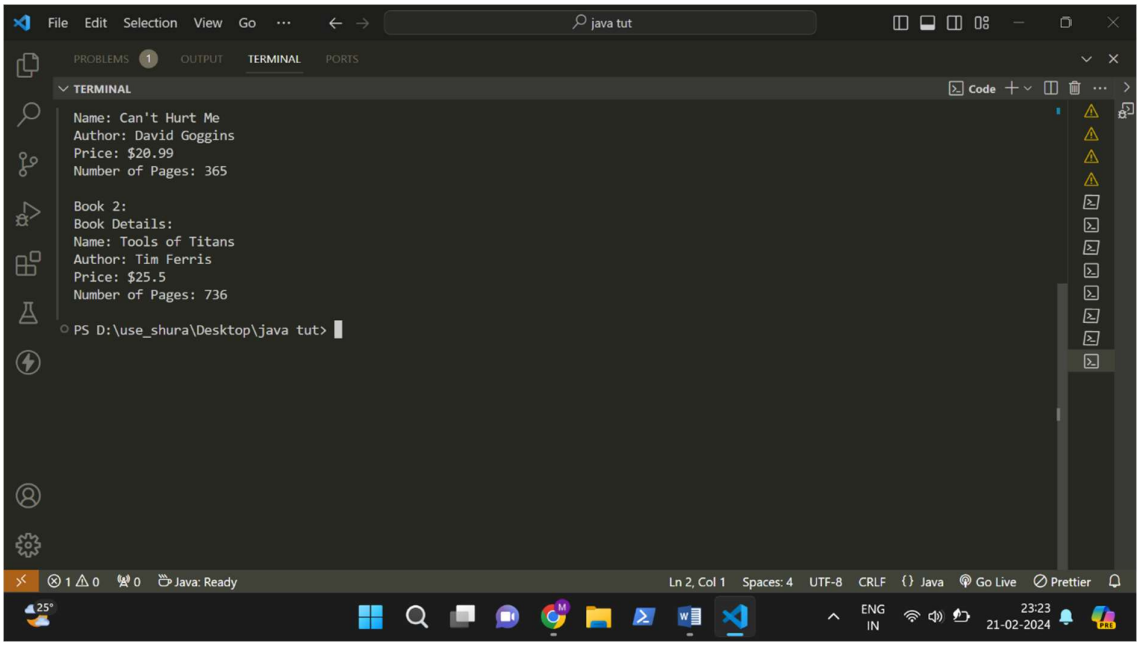
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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

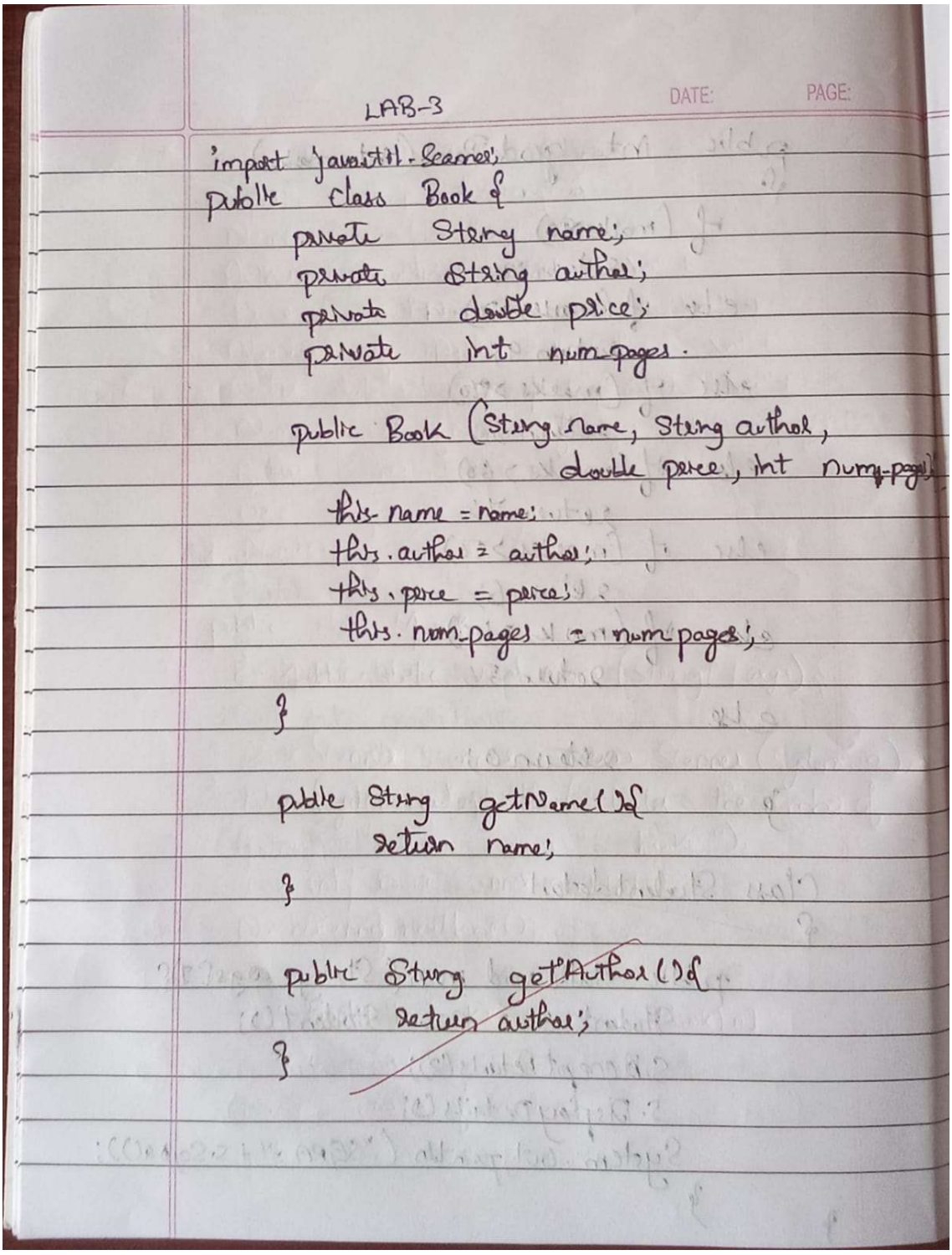
```
import java.util.Scanner;

public class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;

    // Constructor to set the values for the members
    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    // Getter methods
    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return num_pages;
    }

    // Setter methods
    public void setName(String name) {
        this.name = name;
    }

    public void setAuthor(String author) {
```

```
        this.author = author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setNumPages(int num_pages) {
        this.num_pages = num_pages;
    }

    // toString method to display the complete details of the book
    @Override
    public String toString() {
        return "Book Details:\n" +
                "Name: " + name + "\n" +
                "Author: " + author + "\n" +
                "Price: $" + price + "\n" +
                "Number of Pages: " + num_pages;
    }
}}
import java.util.Scanner;

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Author: ");
            String author = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            System.out.print("Number of Pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine(); // Consume newline

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

```
        // Displaying book details
        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":");
            System.out.println(books[i]);
            System.out.println();
        }

        scanner.close();
    }
}
```

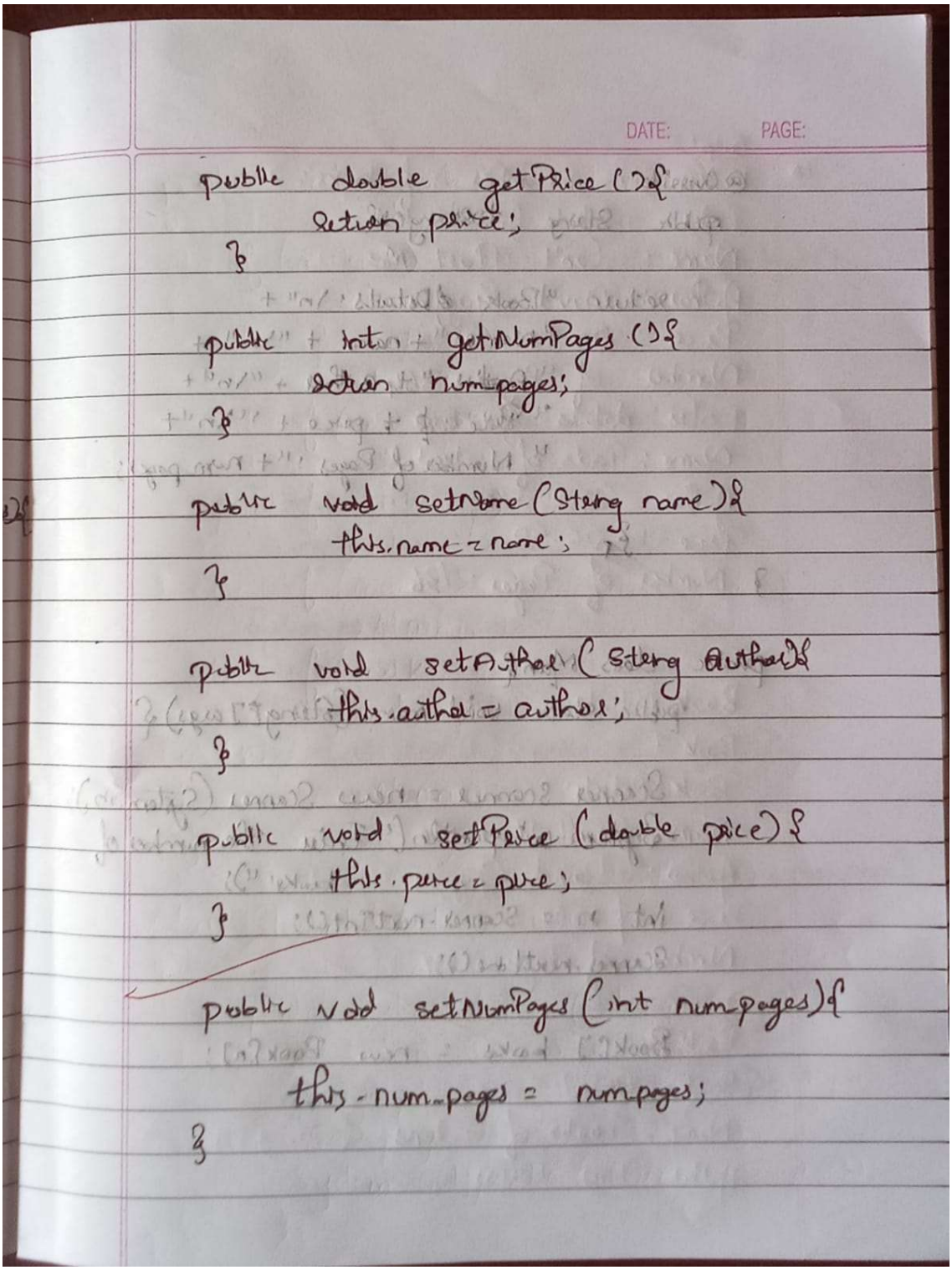
OUTPUT


File Edit Selection View Go
java tut
PROBLEMS 1 OUTPUT TERMINAL PORTS
TERMINAL
Code
Name: Can't Hurt Me
Author: David Goggins
Price: $20.99
Number of Pages: 365
Book 2:
Book Details:
Name: Tools of Titans
Author: Tim Ferris
Price: $25.5
Number of Pages: 736
PS D:\use_shura\Desktop\java tut>
Java: Ready
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Java Go Live Prettier
ENG IN
23:23
21-02-2024

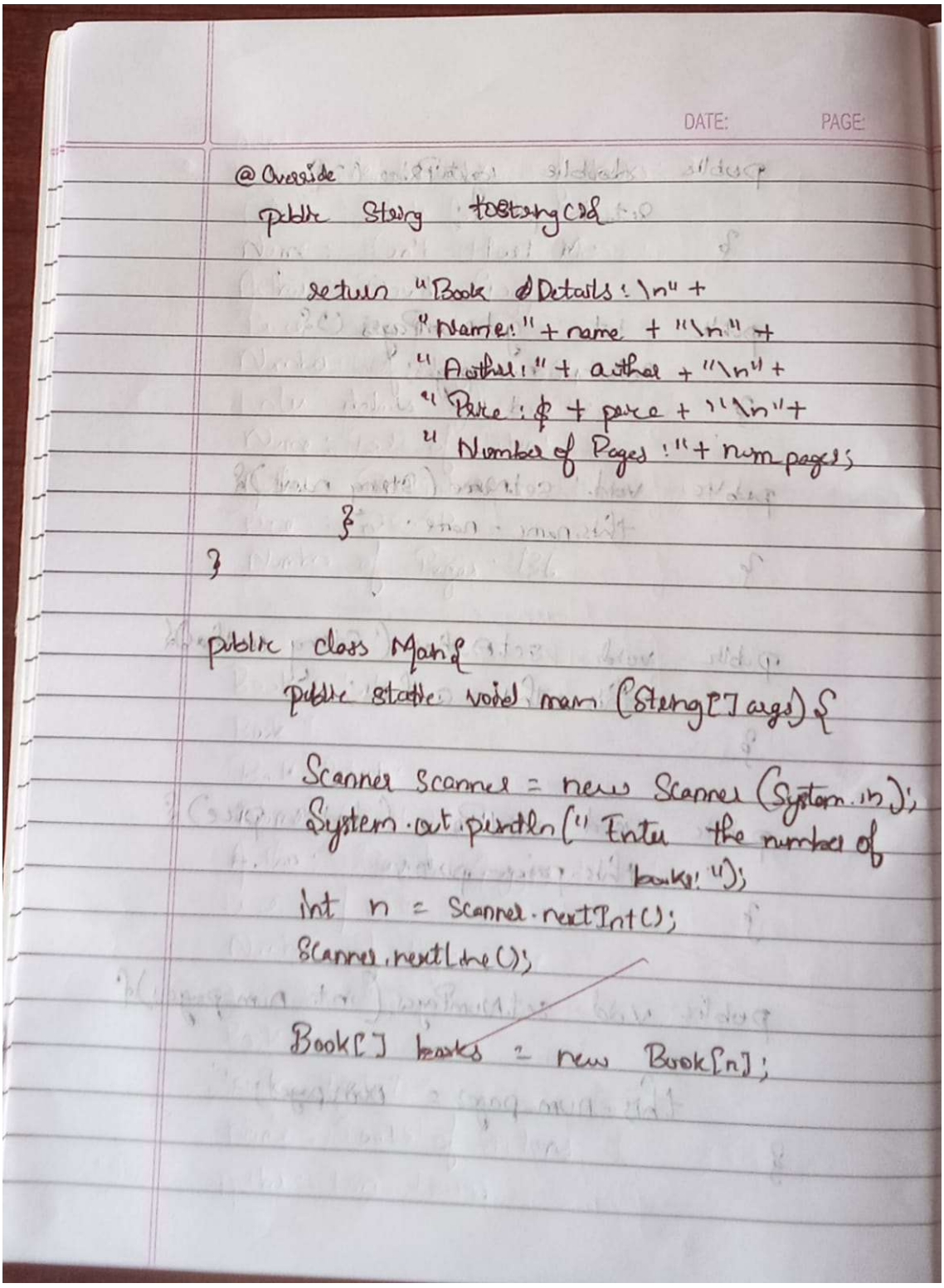


LAB-3 DATE: PAGE:

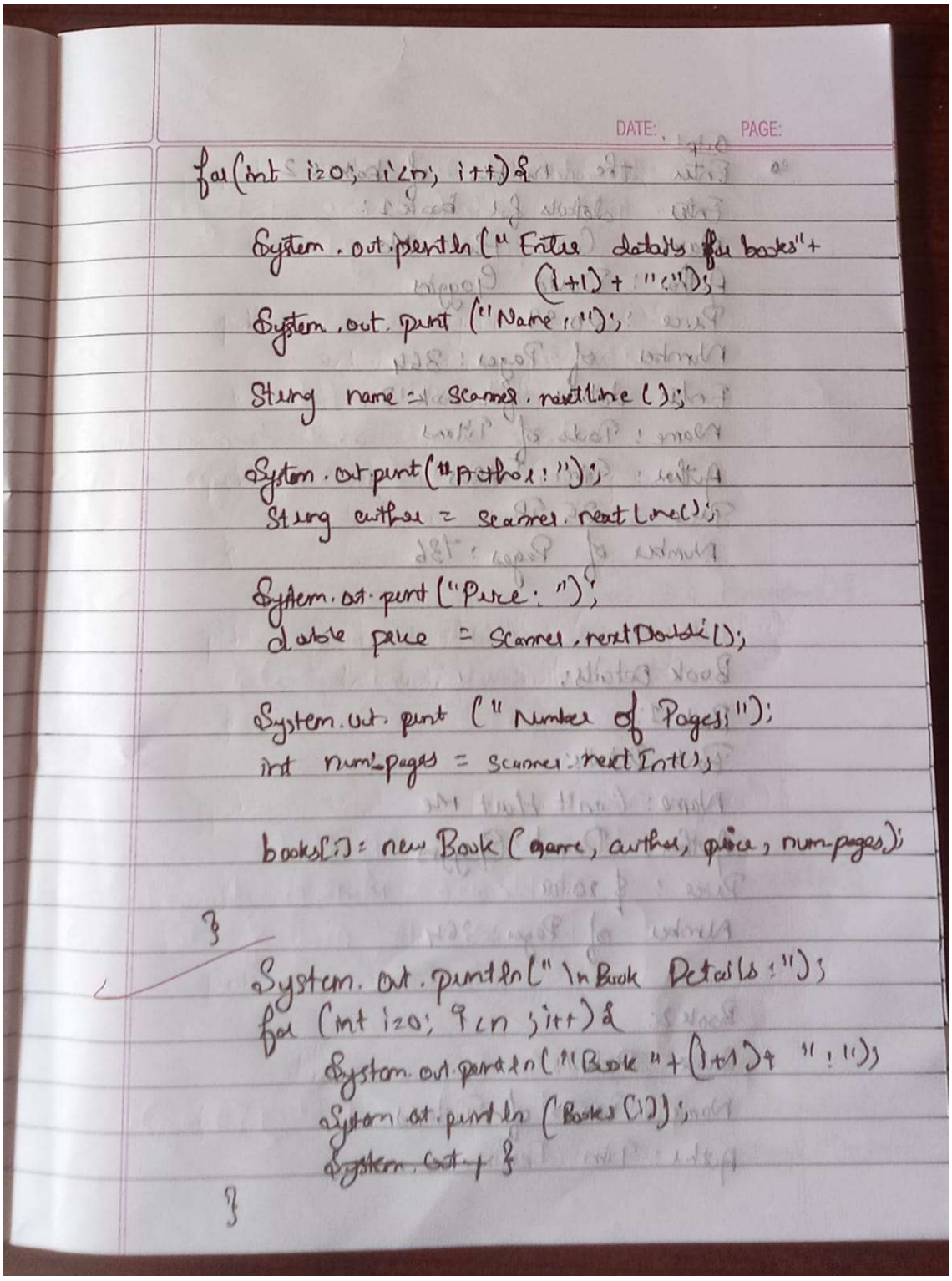
```
import java.util.Scanner;
public class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages.

    public Book (String name, String author,
                 double price, int num_pages)
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }
```



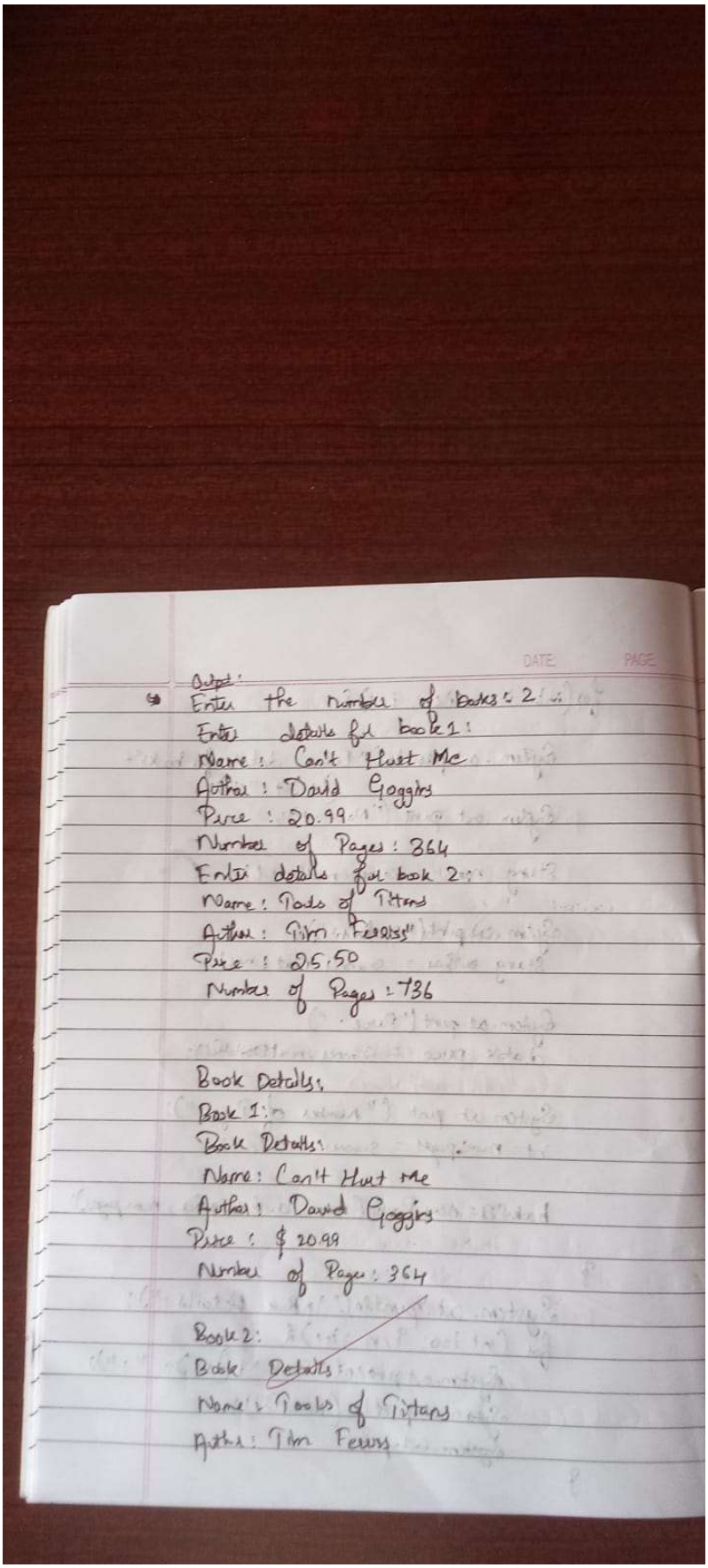
```
DATE:  PAGE:

public double getPrice(){
    return price;
}

public int getNumPages(){
    return num_pages;
}

public void setName(String name){
    this.name = name;
}

public void setAuthor(String author){
    this.author = author;
}

public void setPrice(double price){
    this.price = price;
}

public void setNumPages(int num_pages){
    this.num_pages = num_pages;
}
```



```
@Override
public String toString() {
    return "Book Details:\n" +
           "Name: " + name + "\n" +
           "Author: " + author + "\n" +
           "Price: $" + price + "\n" +
           "Number of Pages: " + num_pages;
    }
}

public class Main {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of
                            books: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Book[] books = new Book[n];
```



```
for(int i=0; i<n; i++){
    System.out.println("Enter details for books"+
                       (i+1)+":");
    System.out.print("Name:");
    String name = scanner.nextLine();

    System.out.print("Author:");
    String author = scanner.nextLine();

    System.out.print("Price:");
    double price = scanner.nextDouble();

    System.out.print("Number of Pages:");
    int num_pages = scanner.nextInt();

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

System.out.println("\nBook Details:");
for(int i=0; i<n; i++){
    System.out.println("Book "+(i+1)+":");
    System.out.println(books[i]);
}
}
```



Output:

Enter the number of books: 2

Enter details for book 1:

Name: Can't Hurt Me

Author: David Goggins

Price: 20.99

Number of Pages: 364

Enter details for book 2:

Name: Tools of Titans

Author: Tim Ferriss

Price: 25.50

Number of Pages: 736

Book Details:

Book 1:

Book Details:

Name: Can't Hurt Me

Author: David Goggins

Price: $ 20.99

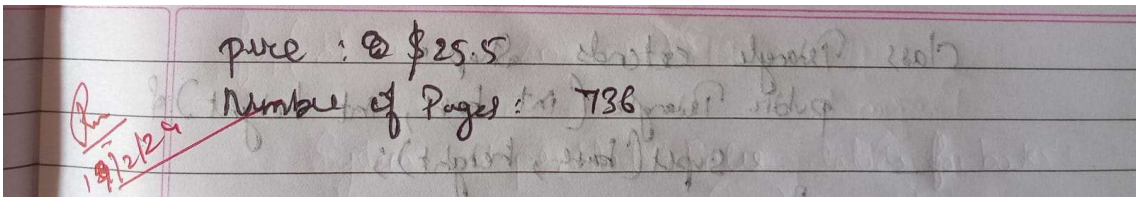
Number of Pages: 364

Book 2:

Book Details:

Name: Tools of Titans

Author: Tim Ferriss


price : @ $25.5
Number of Pages : 736

LABORATORY PROGRAM - 4

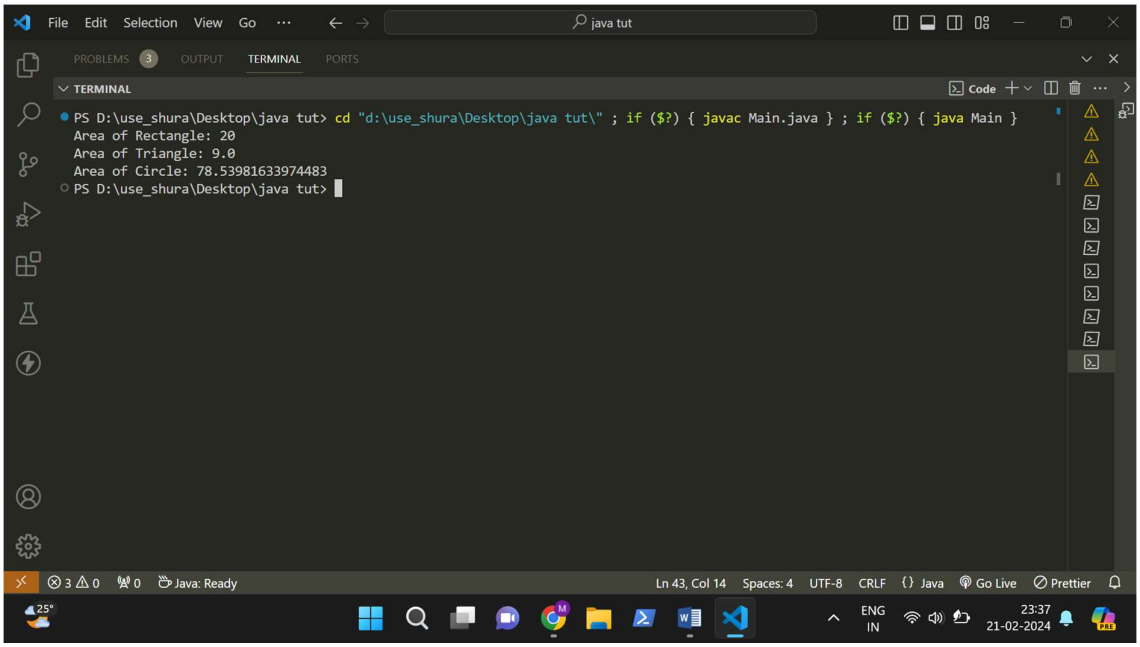
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

```
abstract class Shape {
protected int dimension1;
protected int dimension2;

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

// Abstract method to be implemented by subclasses
public abstract void printArea();
}

class Rectangle extends Shape {
public Rectangle(int length, int width) {
super(length, width);
}

@Override
public void printArea() {
int area = dimension1 * dimension2;
System.out.println("Area of Rectangle: " + area);
}
}

class Triangle extends Shape {
```

```
public Triangle(int base, int height) {
super(base, height);
}

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

class Circle extends Shape {
public Circle(int radius) {
super(radius, 0); // dimension2 is not needed for circle
}

@Override
public void printArea() {
double area = Math.PI * dimension1 * dimension1;
System.out.println("Area of Circle: " + area);
}
}

public class Main {
public static void main(String[] args) {
Rectangle rectangle = new Rectangle(5, 4);
```

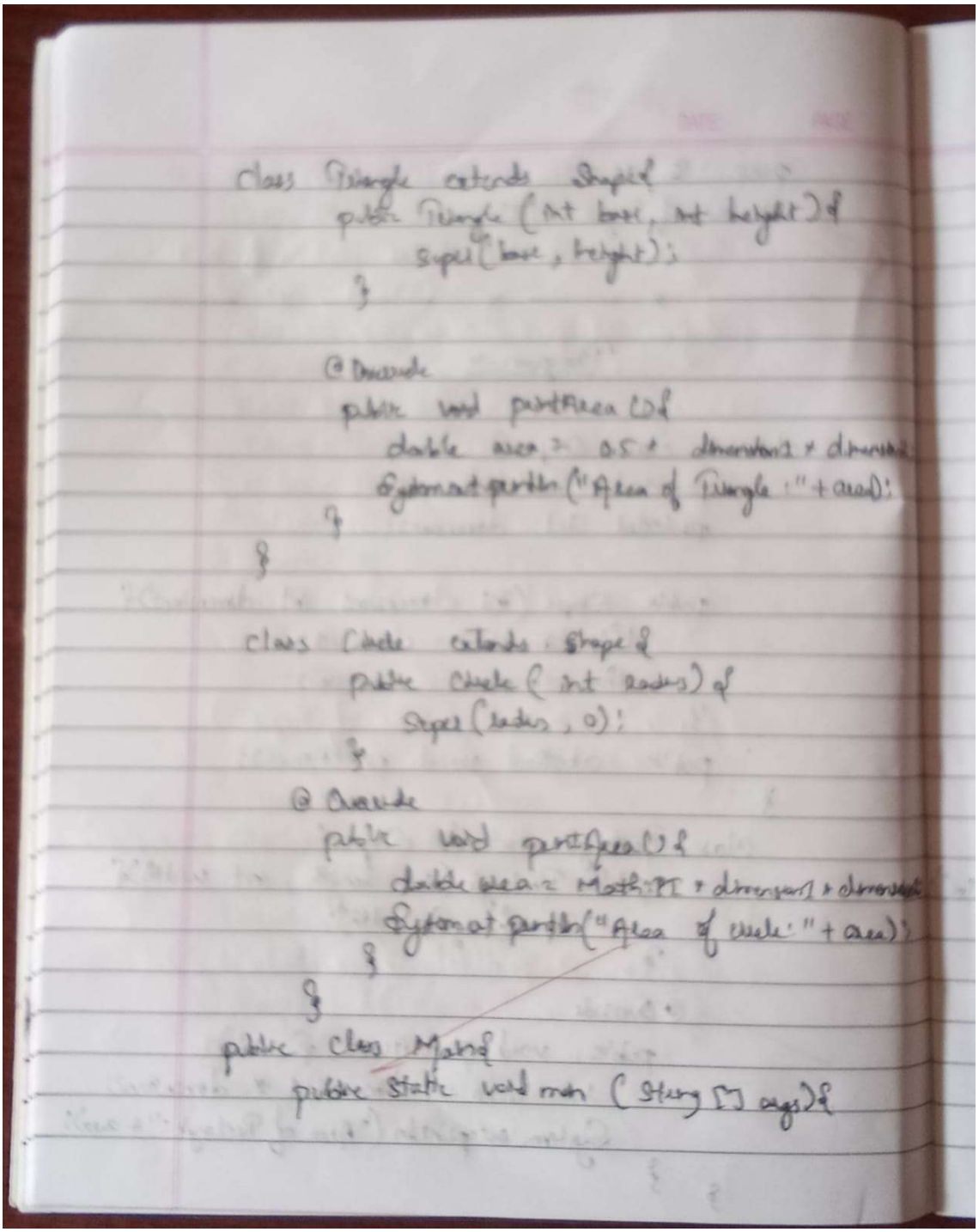
```
rectangle.printArea();

Triangle triangle = new Triangle(3, 6);
triangle.printArea();

Circle circle = new Circle(5);
circle.printArea();
    }
}
```

OUTPUT

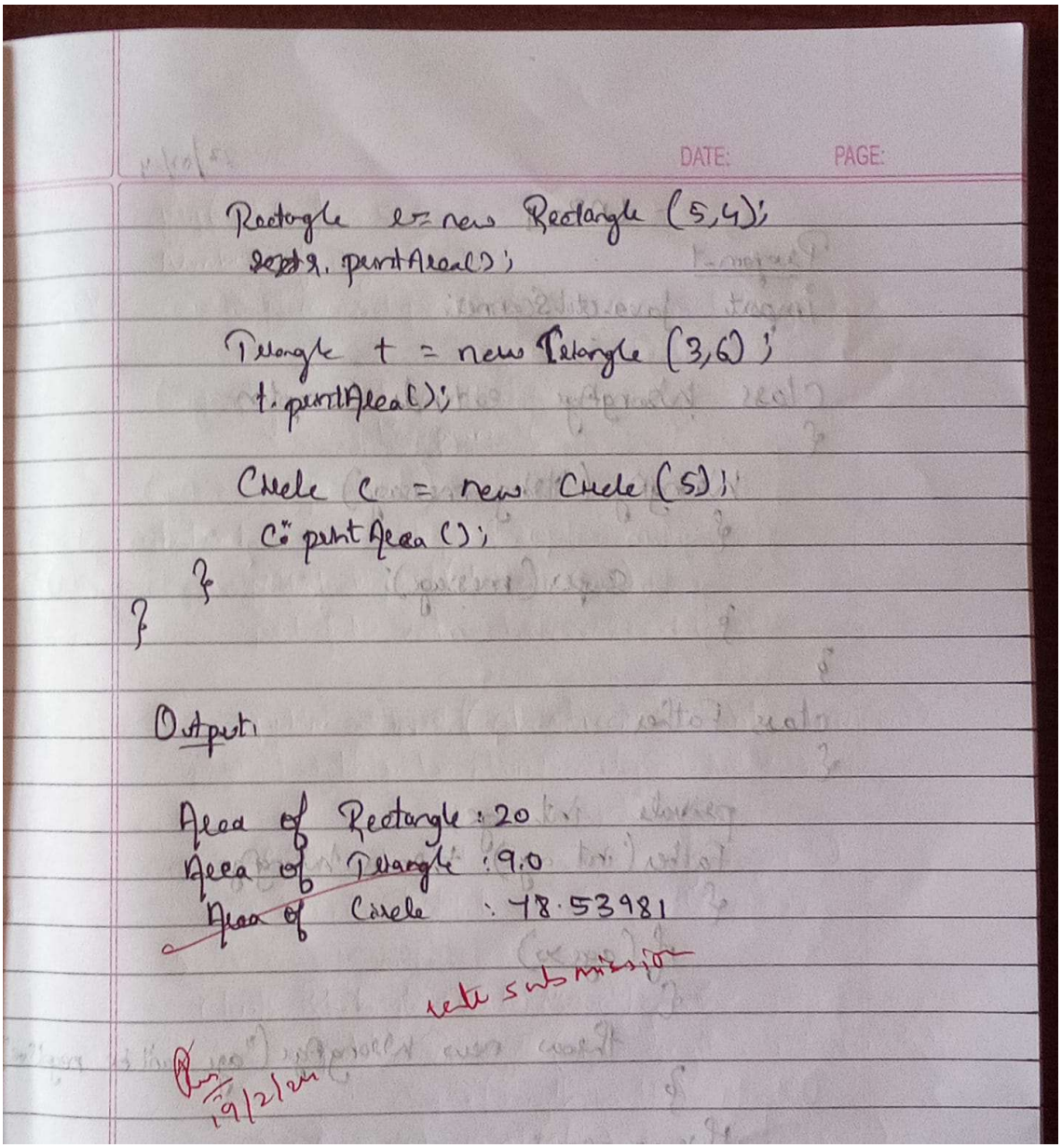


Program-4

```
abstract class Shape {
    protected int dimension1;
    protected int dimension2;

    public Shape (int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle (int length, int width) {
        super(length, width);
    }
    @Override
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle:" + area);
    }
}
```



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

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

class Circle extends Shape{
    public Circle ( int radius){
        super(radius, 0);
    }
    @Override
    public void printArea(){
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle :" + area);
    }
}
public class Main{
    public static void main (String[] args){
```



```
Rectangle r = new Rectangle (5,4);
r.printArea();

Triangle t = new Triangle (3,6);
t.printArea();

Circle c = new Circle (5);
c.printArea();
    }
}
```

Output:

```
Area of Rectangle : 20
Area of Triangle : 9.0
Area of Circle : 78.53981
```

Test submission

19/2/24

LABORATORY PROGRAM - 5

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

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

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

```
import java.util.Scanner;

class Account
{
   String customerName;
long accountNumber;
String accountType;
   double balance;

   public Account(String customerName, long accountNumber, String accountType, double balance)
         {
      this.customerName = customerName;
this.accountNumber = accountNumber;        this.accountType
= accountType;
      this.balance = balance;
   }

   public void deposit(double amount)
         {
      balance += amount;
      System.out.println("Deposit successful. Updated balance: " + balance);
   }
```

```
    public void displayBalance()
            {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: " + balance);
    }
}

class SavAcct extends Account
{
    public SavAcct(String customerName, long accountNumber, double balance)
            {
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeAndDepositInterest(double rate)
            {
        double interest = balance * rate / 100;
balance += interest;
        System.out.println("Interest computed and deposited. Updated balance: " + balance);
    }

    public void withdraw(double amount)
            {
        if (amount <= balance)
                    {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        }
                    else
                    {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class CurrAcct extends Account
{
    double minimumBalance;
    double serviceCharge;

    public CurrAcct(String customerName, long accountNumber,
double balance, double minimumBalance, double serviceCharge)
            {
```

```
        super(customerName, accountNumber, "Current", balance);
this.minimumBalance = minimumBalance;          this.serviceCharge
= serviceCharge;
    }

    private void checkMinimumBalance()
            {
        if (balance < minimumBalance)
                        {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained. Service charge imposed. Updated
balance: " + balance);
        }
    }

    public void withdraw(double amount)
            {
        if (amount <= balance)
                        {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
checkMinimumBalance();
        }
                    else
                    {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

public class Bank
{
    public static void main(String[] args)
            {
        Scanner s1 = new Scanner(System.in);

        System.out.print("Enter customer name for Savings Account: ");
        String SCN = s1.nextLine();
        System.out.print("Enter account number for Savings Account: ");
long SAN = s1.nextLong();
        System.out.print("Enter initial balance for Savings Account: ");
double SIB = s1.nextDouble();
        SavAcct SA = new SavAcct(SCN, SAN, SIB);

        System.out.print("Enter customer name for Current Account: ");
        String CCN = s1.next();
        System.out.print("Enter account number for Current Account: ");
long CAN = s1.nextLong();
```

```
        System.out.print("Enter initial balance for Current Account: ");
double CIB = s1.nextDouble();
        System.out.print("Enter minimum balance for Current Account: ");
double MB = s1.nextDouble();
        System.out.print("Enter service charge for Current Account: ");
double SC = s1.nextDouble();
        CurrAcct CA = new CurrAcct(CCN, CAN, CIB, MB, SC);



        System.out.print("Enter deposit amount for Savings Account: ");
        double SDA = s1.nextDouble();
        SA.deposit(SDA);

        System.out.print("Enter interest rate for Savings Account: ");
        double SIR = s1.nextDouble();
SA.computeAndDepositInterest(SIR);

        System.out.print("Enter withdrawal amount for Savings Account: ");
        double SWA = s1.nextDouble();
        SA.withdraw(SWA);

        System.out.print("Enter deposit amount for Current Account: ");
double CDA = s1.nextDouble();
        CA.deposit(CDA);

        System.out.print("Enter withdrawal amount for Current Account: ");
        double CWA = s1.nextDouble();
        CA.withdraw(CWA);



        System.out.println("\nFinal Balances:");
        System.out.println("Savings Account:");
        SA.displayBalance();

        System.out.println("\nCurrent Account:");
        CA.displayBalance();

    }
}
```

OUTPUT

```
D:\NotePad++\Java>javac Bank.java

D:\NotePad++\Java>java Bank
Enter customer name for Savings Account: Ram
Enter account number for Savings Account: 2324
Enter initial balance for Savings Account: 5000
Enter customer name for Current Account: Ram
Enter account number for Current Account: 2324
Enter initial balance for Current Account: 6000
Enter minimum balance for Current Account: 1000
Enter service charge for Current Account: 100
Enter deposit amount for Savings Account: 2000
Deposit successful. Updated balance: 7000.0
Enter interest rate for Savings Account: 2
Interest computed and deposited. Updated balance: 7140.0
Enter withdrawal amount for Savings Account: 500
Withdrawal successful. Updated balance: 6640.0
Enter deposit amount for Current Account: 1000
Deposit successful. Updated balance: 7000.0
Enter withdrawal amount for Current Account: 750
Withdrawal successful. Updated balance: 6250.0

Final Balances:
Savings Account:
Account Number: 2324
Customer Name: Ram
Account Type: Savings
Balance: 6640.0

Current Account:
Account Number: 2324
Customer Name: Ram
Account Type: Current
Balance: 6250.0
```

19/02/24

Program 5

```
class Account {

    String  cust_name;
    int     acc_no;
    int     acc_type; // 0 for savings, 1 for current
    double  balance = 0.00;

    Account (String cust_name, int acc_no, int acc_type,
                double balance) {

        this.cust_name = cust_name;
        this.acc_no    = acc_no;
        this.acc_type  = acc_type;
        this.balance   = balance;
    }

    public void deposit (int dep_bal) {
        this.balance += dep_bal;
    }

    public void display_balance() {
        System.out.println("Balance:" + this.balance);
    }
```

```
public void withdraw (double withdrawal_amt){
    if (withdrawal_amt > this.balance){
        System.out.println("balance not sufficient");
    }
    else {
        this.balance -= withdrawal_amt;
    }
}
}

class Savings extends Account {

    double interestRate = 0.08;

    Savings (String cust_name, int acc_no, double balance){
        super(cust_name, acc_no, 0, balance);
    }

    public void compute_interest(){
        super.balance += super.balance * interestRate;
    }

    @Override
    public void deposit (int dep_bal){
        compute_interest();
        super.deposit(dep_bal);
    }
}
```

DATE: PAGE:

```
class Current extends Account{
    double min_bal = 10000;
    double penalty = 500;

    public void impose_penalty (double bal){
        if (bal < min_bal){
            super.balance -= penalty;
        }
    }
    Current (String cust_name, int acc_no, double
                balance){
        super (cust_name, acc_no, 1, balance);
        this.impose_penalty (balance);
    }

    @Override
    public void withdraw (double withdrawal_amt){
        if ((super.balance - withdrawal_amt) < min_bal){
            impose_penalty (C.super.balance - withdrawal_amt
        }
        super.withdraw (withdrawal_amt);
    }
}
```

DATE: PAGE:

```
Class bankacc {
    public static void main ( String[] args) {

        Saving a1 = new saving ("Rajesh", 0001, 1200);
        Current a2 = new Current ("Bharath", 0002, 12000);


        a1.deposit(2500);
        a1. display-balance();
        a1. withdraw(1000);
        a1. display-balance();
        a1. withdraw(5000);

        a2. deposit(3000);
        a2. withdraw(7000);
        a2. display-balance();
    }
}
```

Output

19/2/24

Balance: 3796.0
Balance: 2796.0
balance not sufficient

Balance: 7500.0

LABORATORY PROGRAM - 6

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

```
package CIE;

public class Internals extends Student
{
   public int[] internalMarks;

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

package CIE;

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 SEE; import
CIE.Student;

public class External extends Student
{
   public int[] seeMarks;
```

```
    public External(String usn, String name, int sem, int[] seeMarks)
            {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}

import java.util.Scanner; import
CIE.*;
import SEE.*;

public class CalculateFinalMarks
{
    public static void main(String[] args)
            {
        Scanner s1 = new Scanner(System.in);

        System.out.println("Enter the number of students:");
        int n = s1.nextInt();

        Internals[] CS = new Internals[n];
        for (int i = 0; i < n; i++)
                    {
            System.out.println("Enter details for CIE student " + (i + 1));
            System.out.print("USN: ");
            String usn = s1.next();
            System.out.print("Name: ");
            String name = s1.next();
System.out.print("Semester: ");            int
sem = s1.nextInt();
            System.out.println("Enter internal marks for 5 courses:");
int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++)
                                {
                System.out.print("Course " + (j + 1) + ": ");
internalMarks[j] = s1.nextInt();
            }

            CS[i] = new Internals(usn, name, sem, internalMarks);
        }

        External[] SS = new External[n];
        for (int i = 0; i < n; i++)
                    {
            System.out.println("Enter details for SEE student " + (i + 1));
            System.out.print("USN: ");
            String usn = s1.next();
```

```
            System.out.print("Name: ");
            String name = s1.next();
System.out.print("Semester: ");
            int sem = s1.nextInt();
            System.out.println("Enter SEE marks for 5 courses:");
            int[] seeMarks = new int[5];
            for (int j = 0; j < 5; j++)
                                    {
                System.out.print("Course " + (j + 1) + ": ");
seeMarks[j] = s1.nextInt();
            }

            SS[i] = new External(usn, name, sem, seeMarks);
        }

        int[][] finalMarks = new int[n][5];
        for (int i = 0; i < n; i++)
                        {
            for (int j = 0; j < 5; j++)
                                    {
                finalMarks[i][j] = CS[i].internalMarks[j] + SS[i].seeMarks[j];
            }
        }

        System.out.println("\nFinal Marks:");
        for (int i = 0; i < n; i++)
                        {
            System.out.print("USN: " + CS[i].usn +", Name: " + CS[i].name +", Semester: " + CS[i].sem
+", Final Marks: ");
            for (int j = 0; j < 5; j++)
                                    {
                System.out.print(finalMarks[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT

```
D:\NotePad++\Java\Packages>javac CalculateFinalMarks.java

D:\NotePad++\Java\Packages>java CalculateFinalMarks
Enter the number of students:
1
Enter details for CIE student 1
USN: 1
Name: Ram
Semester: 3
Enter internal marks for 5 courses:
Course 1: 47
Course 2: 48
Course 3: 49
Course 4: 50
Course 5: 49
Enter details for SEE student 1
USN: 1
Name: Ram
Semester: 3
Enter SEE marks for 5 courses:
Course 1: 48
Course 2: 49
Course 3: 47
Course 4: 50
Course 5: 50

Final Marks:
USN: 1, Name: Ram, Semester: 3, Final Marks: 95 97 96 100 99
```

DATE: PAGE:

Program-6

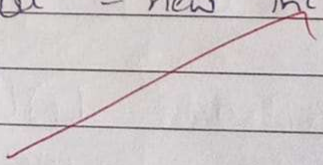
cie

Student.java

```
package cie;
public class Student{
    public char name[10];
    public int usn;
    public int sem;
}
```

internals.java

```
package cie;
public class internals{
    public int[][] ass;
    public internals(int n){
        ass = new int[n][5];
    }
}
```



DATE: PAGE:

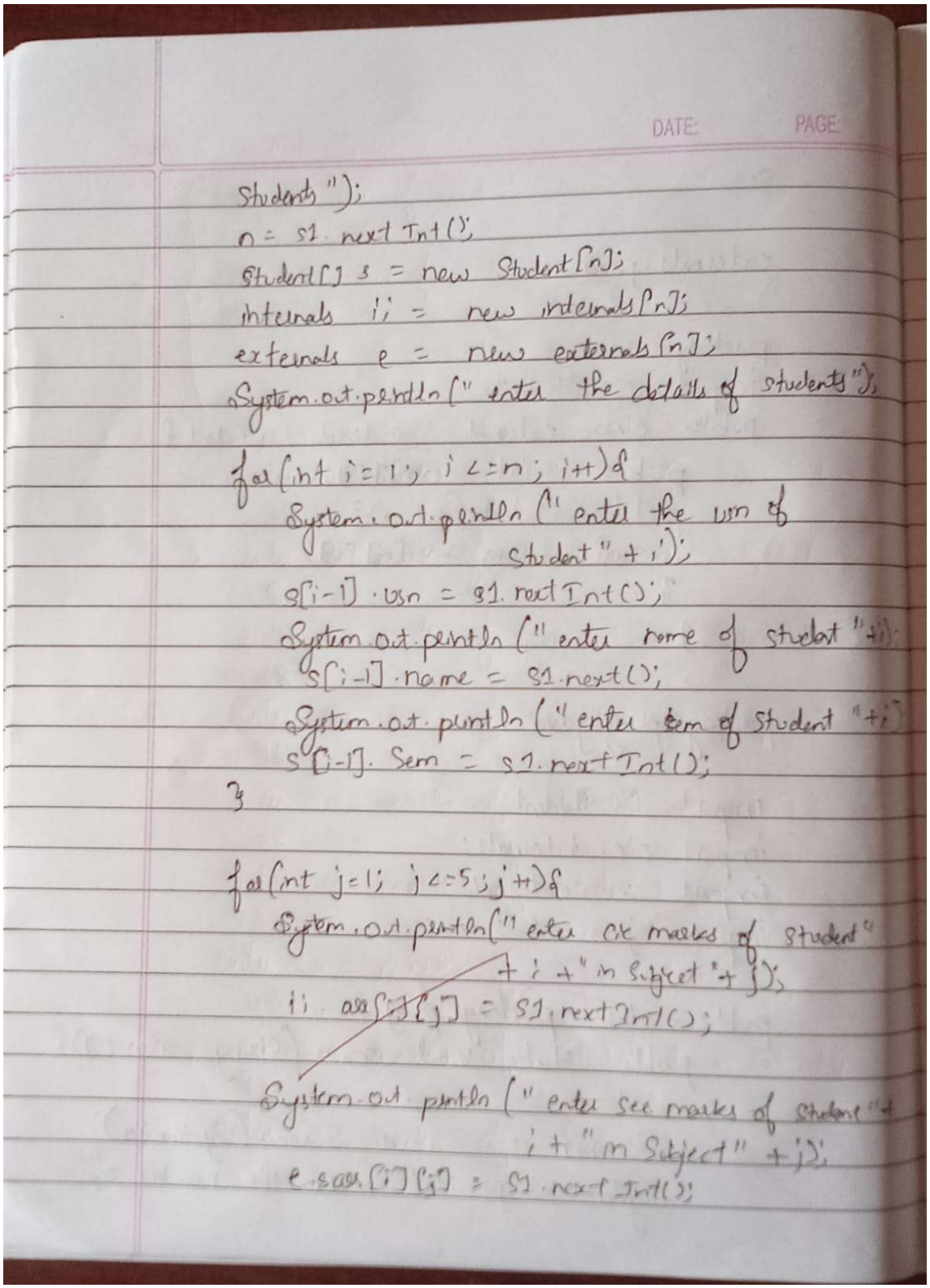
See

externals.java

```
package see;
import cie.Student;
public class external extends Student {
    public int sass[][];
    public student(int n){
      sass = new int[n][5];
    }
}
```

Package plog.java

```
import cie.Student;
import cie.internals;
import see.externals;
import java.util.Scanner;

public class package plog {
    public static void main(String args[]){
      int n;
      Scanner s1 = new Scanner(System.in);
      System.out.println("enter the number of
```



```
Students");
n = s1.nextInt();
Student[] s = new Student[n];
internals ii = new internals[n];
externals e = new externals[n];
System.out.println(" enter the details of students");

for(int i=1; i<=n; i++){
    System.out.println(" enter the usn of
                        student" + i);
    s[i-1].usn = s1.nextInt();
    System.out.println(" enter name of student "+i);
    s[i-1].name = s1.next();
    System.out.println(" enter sem of student "+i);
    s[i-1].Sem = s1.nextInt();
}

for(int j=1; j<=5; j++){
    System.out.println(" enter CIE marks of student"
                        + i + " in Subject " + j);
    ii.aa[i][j] = s1.nextInt();

System.out.println(" enter see marks of student"+
                    i + " in Subject" + j);
e.saa[i][j] = s1.nextInt();
```

DATE: PAGE:

```
for (int i=1; i<=n; i++){
    System.out.println("Displaying total marks of
                        student" +i);

for (int j=1; j<=5; j++){
    float total = ii.marks[i][j] + ss.marks[i][j]/2;
    System.out.println("total marks of student" +i
                       +"in subject" +j+"is:"+ total);
    }
}
}
}
```

Output:

enter the number of students
2
enter the details of students
enter the usn of student 1
164
enter the name of student 1
Raju
enter the sem of student 1
4

LABORATORY PROGRAM - 7

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

```
import java.util.Scanner;

class WrongAge extends Exception
{
   public WrongAge()
          {
      super("Invalid age! Age cannot be negative nor zero.");
   }

   public WrongAge(String message)
          {
      super(message);
   }
}


class Father
{
   private int age;

   public Father(int age) throws WrongAge
          {
      if (age <= 0)
                    {
         throw new WrongAge();
      }
      this.age = age;
   }

   public int getAge()
          {
      return age;
   }
}

class Son extends Father
{
   private int sonAge;
```

```
    public Son(int fatherAge, int sonAge) throws WrongAge
            {
        super(fatherAge);

        if (sonAge >= fatherAge)
                    {
            throw new WrongAge("Son's age should be less than Father's age.");
        }

        this.sonAge = sonAge;
    }

    public int getSonAge()
            {
        return sonAge;
    }
}

public class InheritanceException
{
    public static void main(String[] args)
            {
        try
                    {
            Scanner s1 = new Scanner(System.in);

            System.out.print("Enter Father's age: ");
int fatherAge = s1.nextInt();            Father f =
new Father(fatherAge);

            System.out.print("Enter Son's age: ");
int sonAge = s1.nextInt();
            Son s = new Son(f.getAge(), sonAge);

            System.out.println("Father's age: " + f.getAge());
            System.out.println("Son's age: " + s.getSonAge());

        } catch (WrongAge e)
                    {
            System.out.println("Exception: " + e);
        } catch (Exception e)
                    {
            System.out.println("Exception: Invalid input. Please enter valid integer values.");
        }
    }
}
```

OUTPUT

```
D:\NotePad++\Java>javac InheritanceException.java

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 46
Enter Son's age: 56
Exception: WrongAge: Son's age should be less than Father's age.

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 56
Enter Son's age: 24
Father's age: 56
Son's age: 24
```

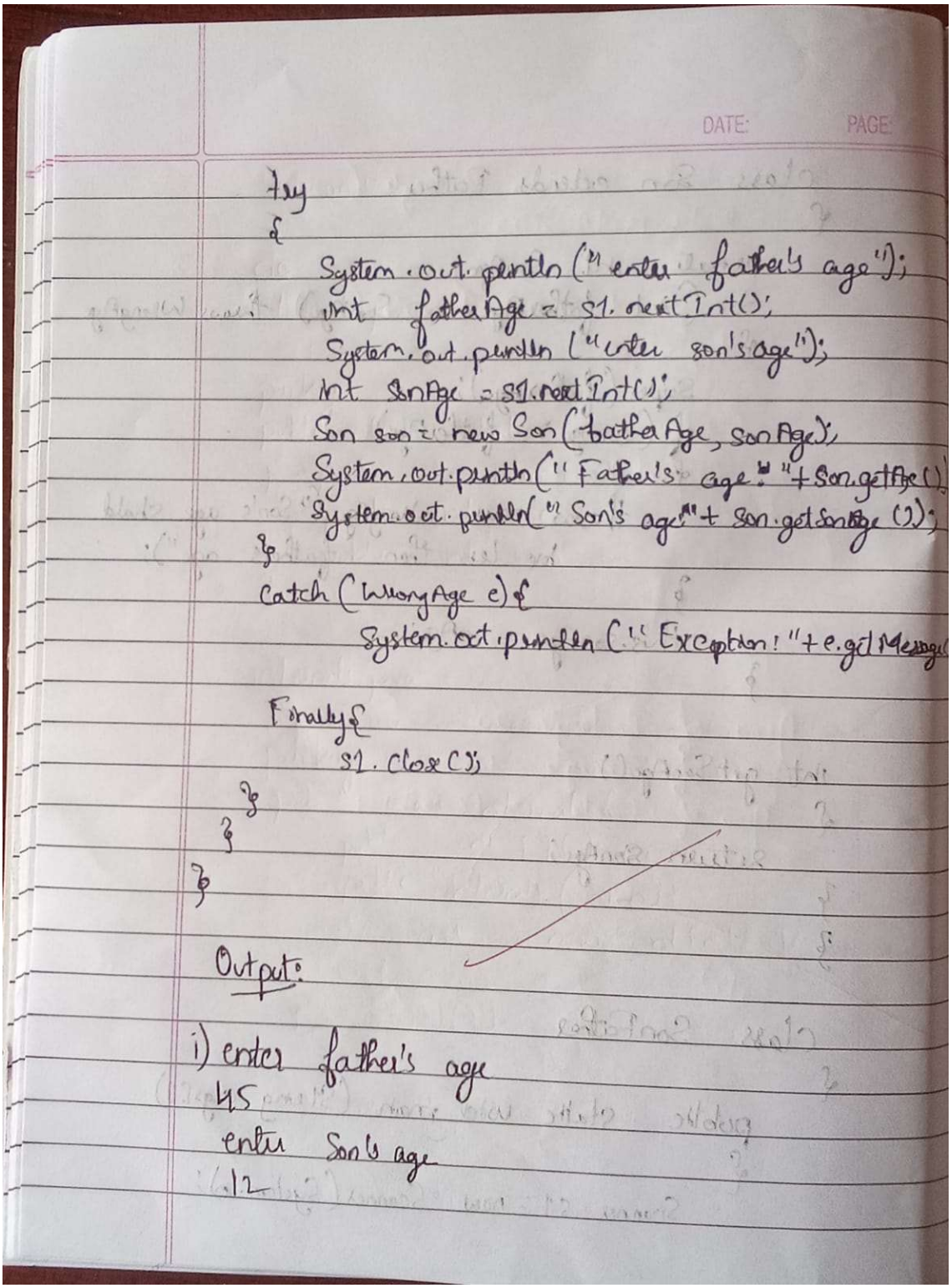
DATE: 22/07/24 PAGE:

LAB-4

Program-1

```
import java.util.Scanner;

class WrongAge extends Exception
{
    WrongAge(String message)
    {
        super(message);
    }
}
class Father
{
    private int age;
    Father(int age) throws WrongAge
    {
        if(age<0)
        {
            throw new WrongAge("age can't be negat
        }
    this.age=age;
    }
    int getAge()
    {
        return Age;
    }
```

```
class Son extends Father
{
    private int SonAge;
    Son(int fatherAge, int SonAge) throws WrongAge
    {
        super(fatherAge);
        if (SonAge >= fatherAge)
        {
            throw new WrongAge("Son's age should
                be less than father's age");
        }
        this.SonAge = SonAge;
    }

int getSonAge()
{
    return SonAge;
}
}

class SonFather
{
    public static void main (String args[])
    {
        Scanner s1 = new Scanner(System.in);
```



```
try
{
    System.out.println("enter father's age");
    int fatherAge = s1.nextInt();
    System.out.println("enter son's age");
    int SonAge = s1.nextInt();
    Son son = new Son(fatherAge, sonAge);
    System.out.println("Father's age:" + son.getFage());
    System.out.println("Son's age:" + son.getSonAge());
}
catch (WrongAge e){
    System.out.println("Exception!" + e.getMessage
Finally{
    s1.close();
}
}
}
```

Output:

i) enter father's age
45
enter Son's age
12

DATE: PAGE:

Father's age : 45
Son's age : 12

ii) enter father's age
-40
enter Son's age
-20
Exception: Age can't be negative

iii) enter father's age
12
enter Son's age
30
Exception: Son's age should be less than father's age.

Ry
22/1/2024

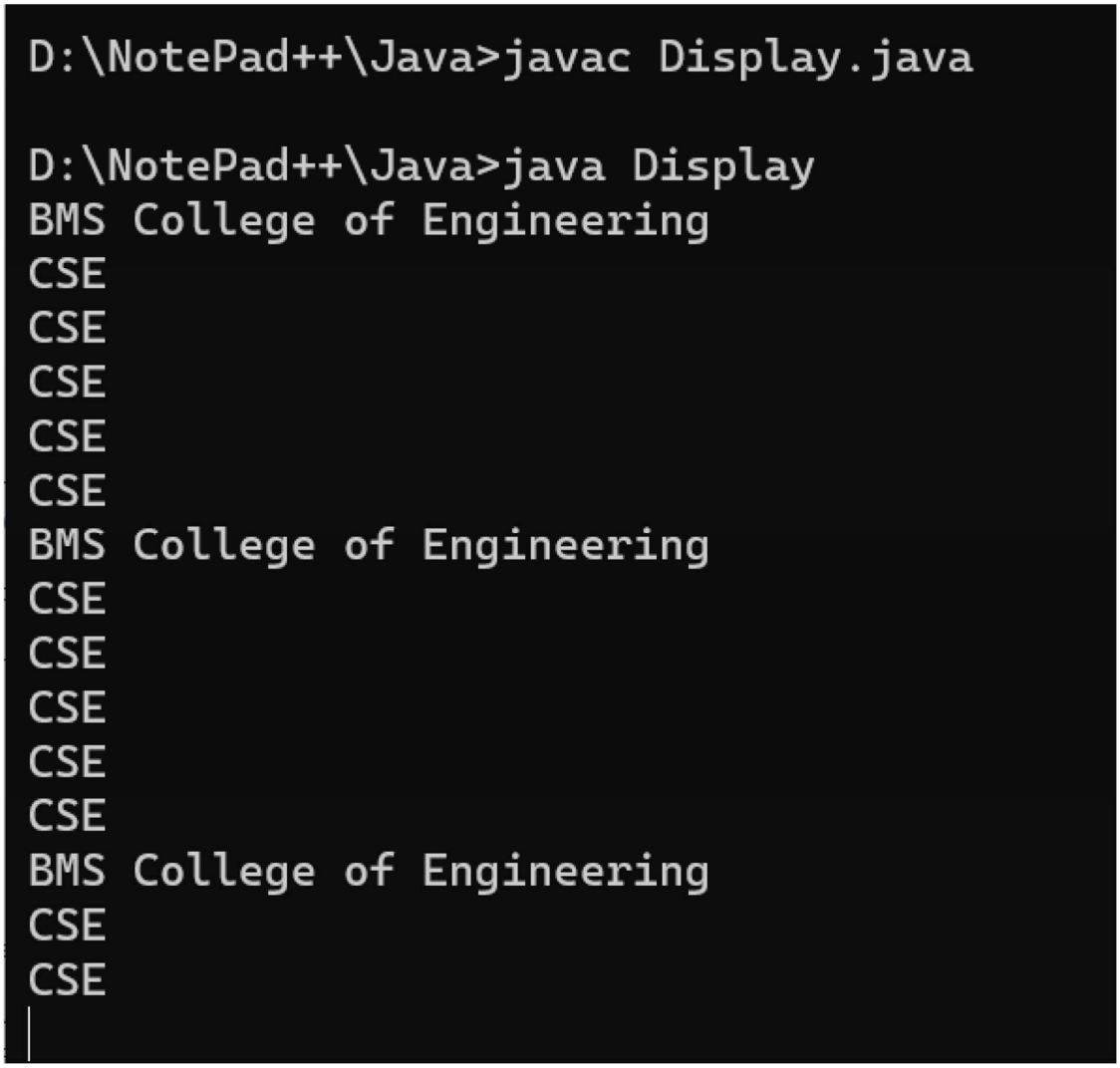
LABORATORY PROGRAM - 8

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 BMSThread implements Runnable
{
   public void run()
        {
            while (true)
                {
        try
                                {
                                        System.out.println("BMS College of Engineering");
                                        Thread.sleep(10000);
                                }
                        catch (InterruptedException ie)
                                {
                                        System.out.println("BMSThread is Interrupted");
                                }
                }
        }
}

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

public class Display
{
   public static void main(String[] args)
```

```
        {
                Thread bms = new Thread(new BMSThread());
                Thread cse = new Thread(new CSEThread());
                bms.start();
                cse.start();
        }
}
```

OUTPUT



Program-8

DATE: 5/2/24 PAGE:

Q) Write a program which creates two threads, one displaying "BMSCE" once every ten seconds and another displaying "CSE" once every two seconds.

→

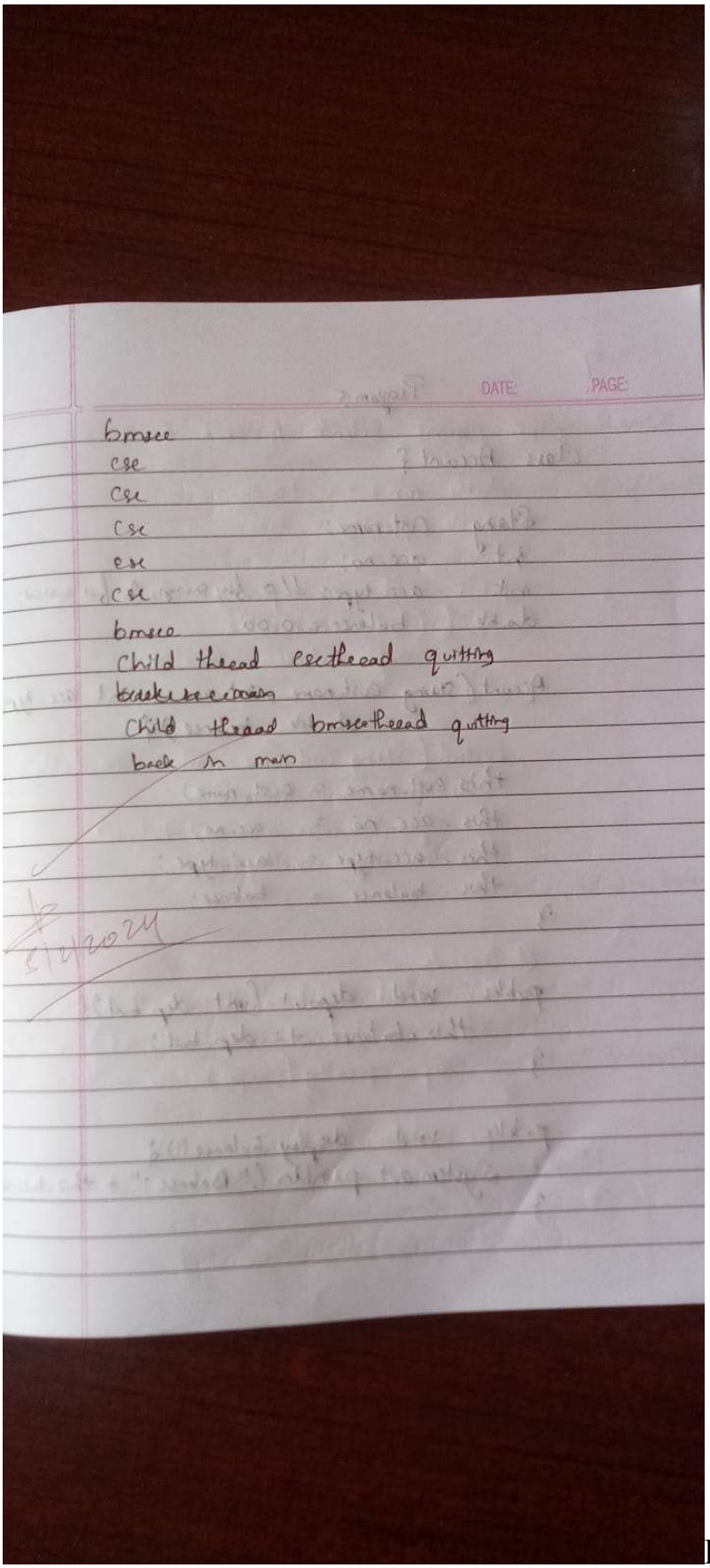
```
class newthread implements Runnable
{
    Thread t;
    String str;
    String name;
    int sec;
    int loop;
    newthread (String name, String str, int sec, int loop)
    {
     this.name = name;
     t = new Thread (this, this.name);
     this.loop = loop;
     this.str = str;
     this.sec = sec;
     t.start();
    }
```

```
public void run()
{
  try
  {
    for (int n=this.loop; n>0; n--)
    {
      System.out.println(sts);
      Thread.sleep(sec);
    }
  }

  catch (InterruptedException ie)
  {
    System.out.println("Child thread" + this.name
                       + " interrupted");

  }
  System.out.println("Child thread" + this.name
                     + "quitting");
  }
}

Class threadprog{
  public static void main (String [] args){
  System.out.println("entered main");
```

```
new newthread ("bmscethread", "bmsce", 16000, 3);
new newthread ("csethread", "cse", 2000, 10);

try{
    Thread.sleep(2000);
}

catch(InterruptedException ie)
{
    System.out.println("main thread interrupted");
}
System.out.println("back in main");

    }
}
```

Output:

entered main
bmsce
cse
cse
csc
cse
cse


DATE: PAGE:
bmsce
cse
cse
cse
ese
cse
bmsce
Child thread csethread quitting
Child thread bmscethread quitting
back in main

LABORATORY

PROGRAM - 9

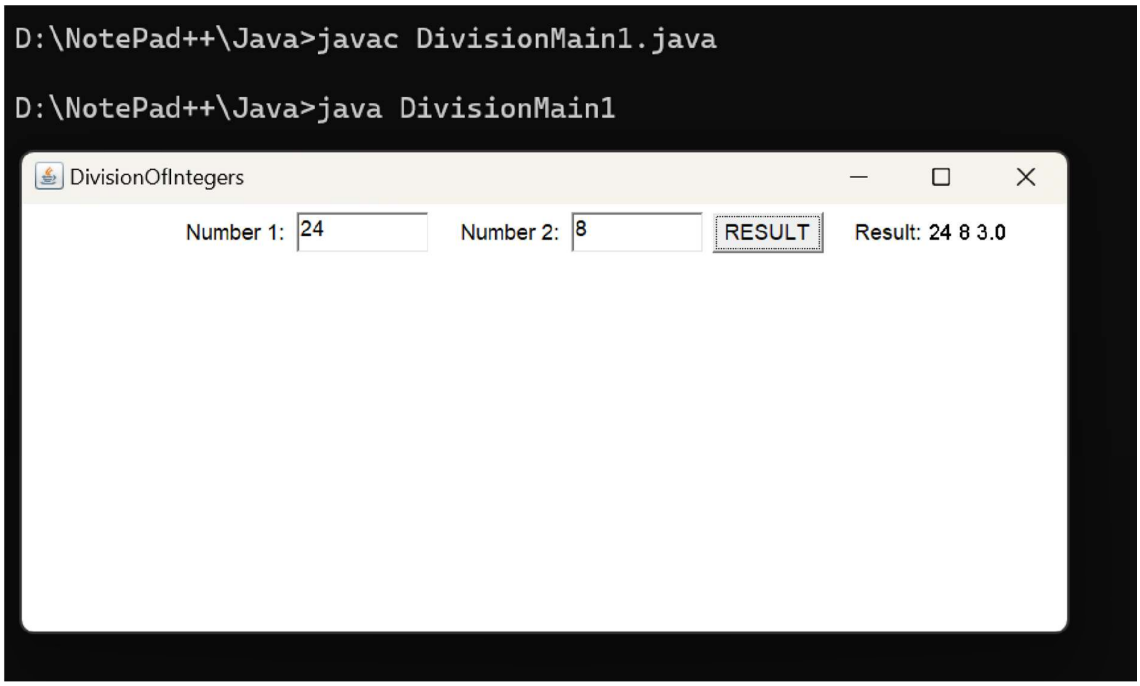
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

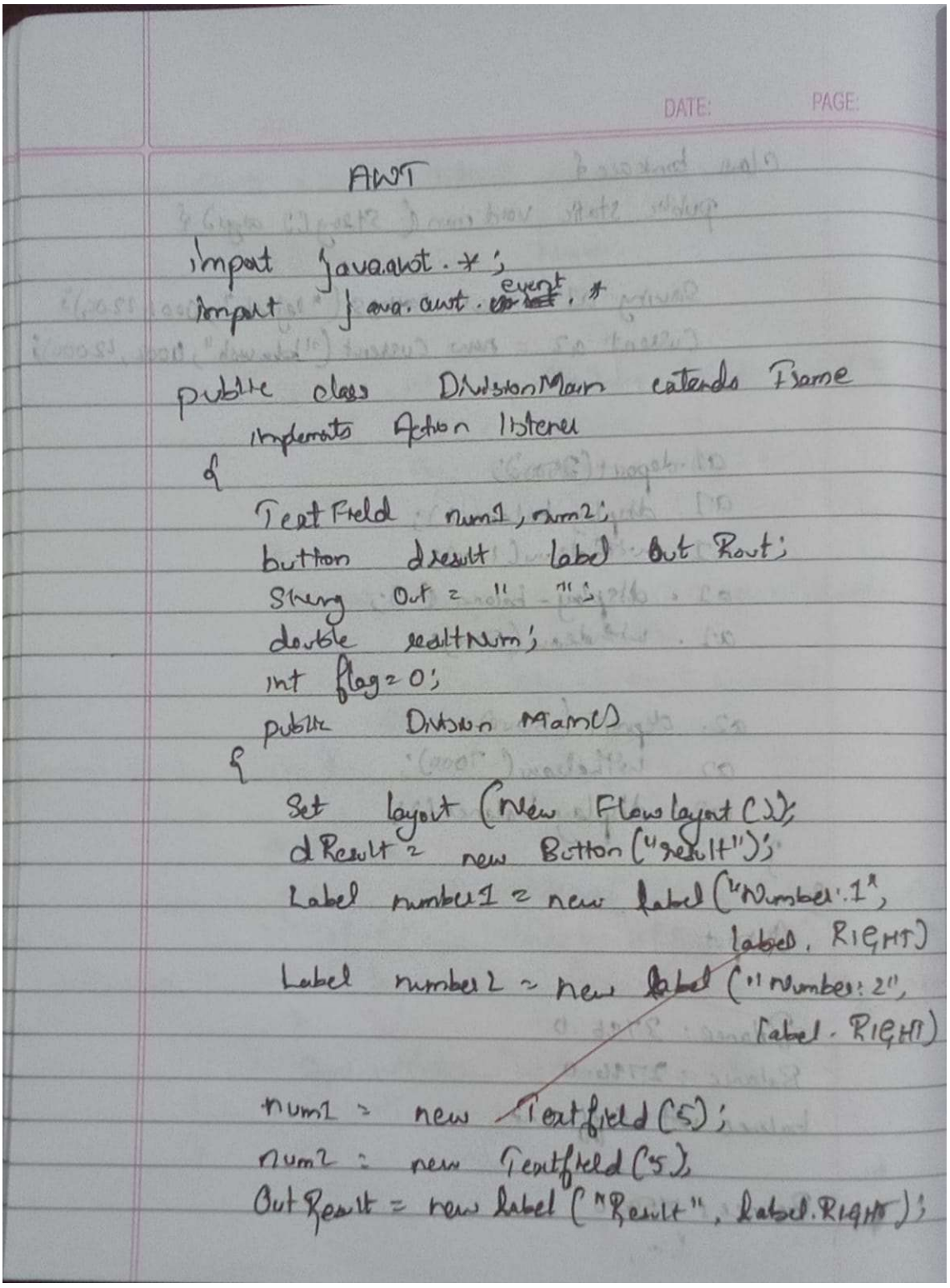
```
import java.awt.*;
import java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener {
        TextField num1,num2;
        Button dResult;
        Label outResult;
String out="";   double
resultNum;
        int flag=0;

        public DivisionMain1()
        {
                setLayout(new FlowLayout());

                dResult = new Button("RESULT");
                Label number1 = new Label("Number 1:",Label.RIGHT);
                Label number2 = new Label("Number 2:",Label.RIGHT);
                num1=new TextField(5);
num2=new TextField(5);
                outResult = new Label("Result:",Label.RIGHT);

                add(number1);
add(num1);              add(number2);
add(num2);              add(dResult);
                add(outResult);

                num1.addActionListener(this);
num2.addActionListener(this);           dResult.addActionListener(this);
                addWindowListener(new WindowAdapter()
                {
                        public void windowClosing(WindowEvent we)
                        {
                                System.exit(0);
                        }
                });
        }
```

```
public void actionPerformed(ActionEvent ae)
{
	int n1,n2;
	try
	{
		if (ae.getSource() == dResult)
		{
			n1=Integer.parseInt(num1.getText());
		      n2=Integer.parseInt(num2.getText());

			/*if(n2==0)
				throw new ArithmeticException();*/
			out=n1+" "+n2+" ";
			resultNum=n1/n2;
			out+=String.valueOf(resultNum);
			repaint();

		}
	}
	catch(NumberFormatException e1)
	{
		flag=1;
		out="Number Format Exception!  "+e1;
		repaint();
	}
	catch(ArithmeticException e2)
	{
		flag=1;
		out="Divide by 0 Exception!  "+e2;
		repaint();
	}

}

public void paint(Graphics g)
{
	if(flag==0)
	g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.
	getHeight()-8);
	else
	g.drawString(out,100,200);
	flag=0;
}

public static void main(String[] args)
{
```

```
                DivisionMain1 dm=new DivisionMain1();
dm.setSize(new Dimension(800,400));
dm.setTitle("DivisionOfIntegers");                dm.setVisible(true);
        }
}
```

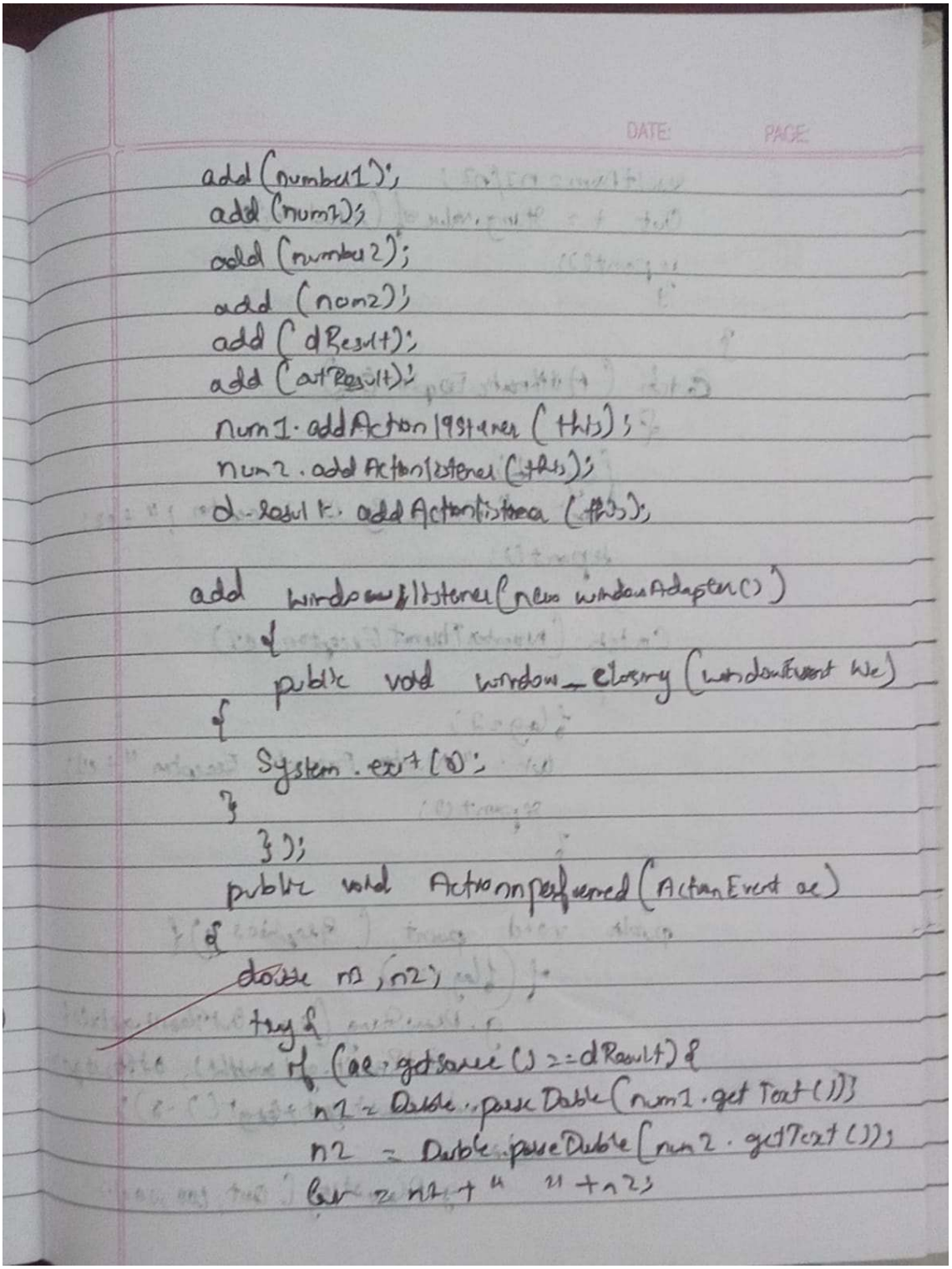
OUTPUT





AWT

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

public class DivisionMain extends Frame
    implements ActionListener
{
    TextField num1, num2;
    button dresult label Out Rout;
    String Out = "";
    double resultNum;
    int flag = 0;
    public DivisionMain()
{
    Set layout (New FlowLayout(2);
    dResult = new Button("result");
    Label number1 = new label("Number:1",
                              label.RIGHT)
    Label number2 = new label("Number:2",
                              label.RIGHT)

num1 = new Textfield(5);
num2 = new Textfield(5);
OutResult = new label("Result", label.RIGHT);
```



```
add(number1);
add(num1);
add(number2);
add(num2);
add(dResult);
add(arResult);
num1.addActionListener(this);
num2.addActionListener(this);
d.result.addActionListener(this);

add windowListener(new windowAdapter()
    {
      public void window_closing(windowEvent we)
{
  System.exit(0);
}
  });
public void ActionPerformed(ActionEvent ae)
{
  double n1, n2);
  try{
    if(ae.getSource() == dResult){
    n1 = Double.parseDouble(num1.getText());
      n2 = Double.parseDouble(num2.getText());
    bar = n1 + " " + n2;
```

DATE: PAGE:

REPORT

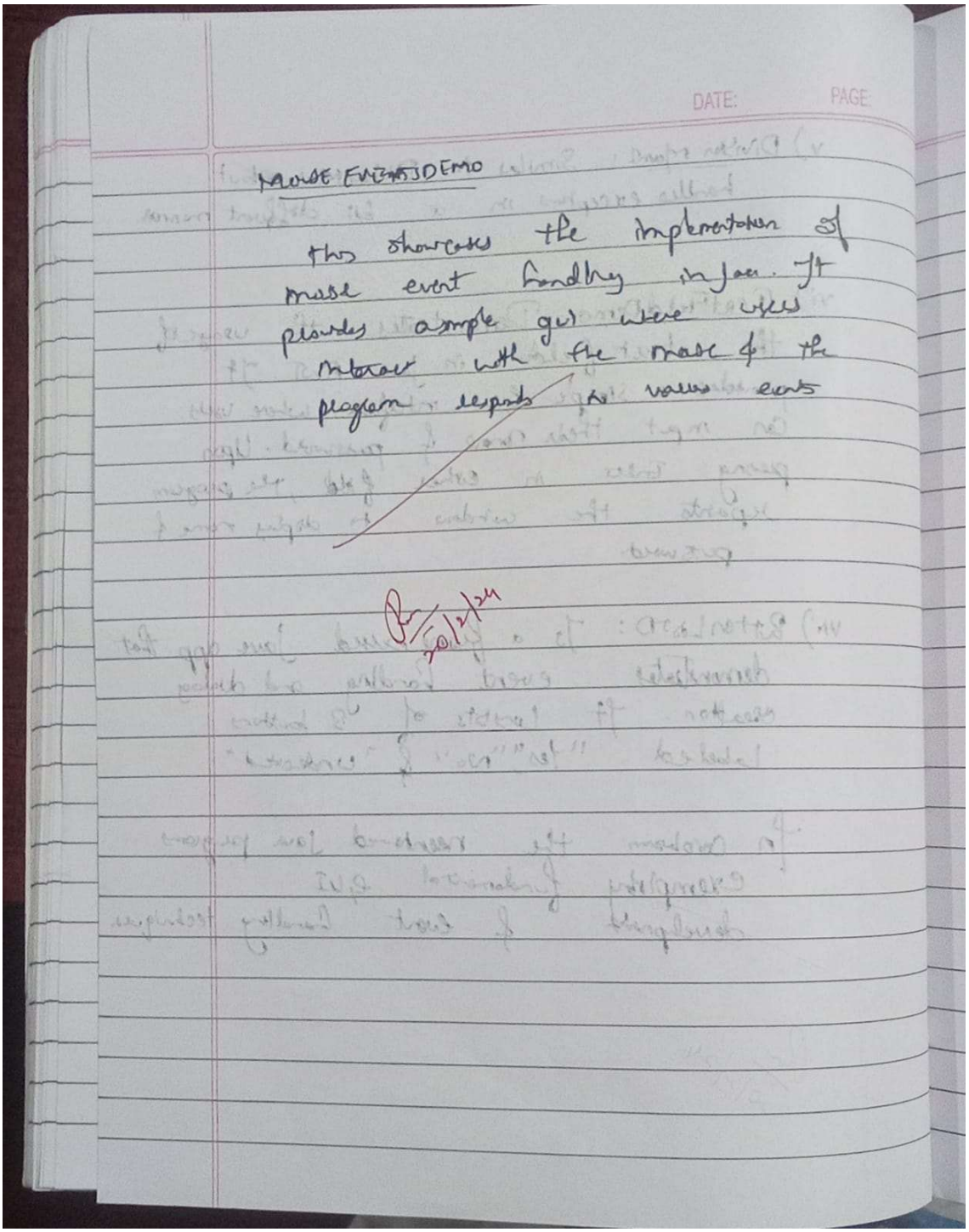
The given program utilize Java's AWT and Swing libraries to create GUI Apps. These programs showcase various event handling in Java.

i) ButtonDemo: Is an applet that demonstrates event handling in Java AWT. It consists of three buttons labeled "Yes", "No" and "Undecided". Clicking on each button triggers an action event and the corresponding message is displayed on the applet.

ii) ButtonDemo: Is another frame-based Java App that demonstrates event handling and consists of three buttons similar to ButtonDemo program.

iii) buttonMagic: Is a frame-based Java app that implements a puzzle game.

iv) DivisionMain: Is a frame-based Java app that allows user to input 2 numbers & calculate their division.


DATE:
PAGE:
MOUSE EVENTS DEMO
This showcases the implementation of mouse event handling in Java. It provides a simple gui where user interact with the mouse & the program responds to various events
20/2/24