VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

OR

Object Oriented Java

Programming (23CS3PCOOJ)

Submitted by

G M Kusuma
24BECS402

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

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 **G M Kusuma** 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, Bul aum. The Lab report has been approved as it satisfies the academic requirements in respect of object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

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

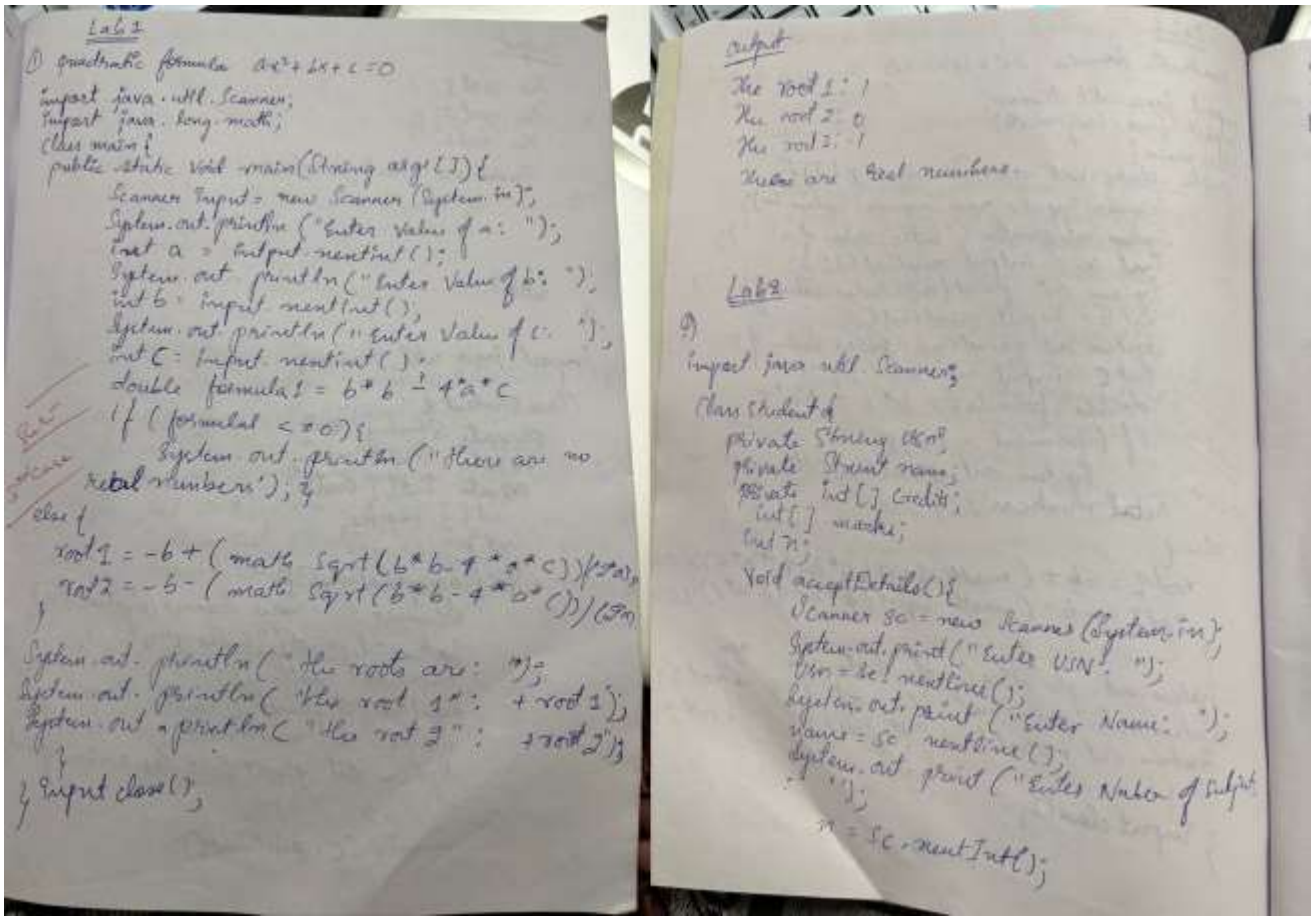
Index

Github Link:
https://github.com/Kusumagm07/OOJ.git

Program 1

Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions.

Algorithm:



Code:

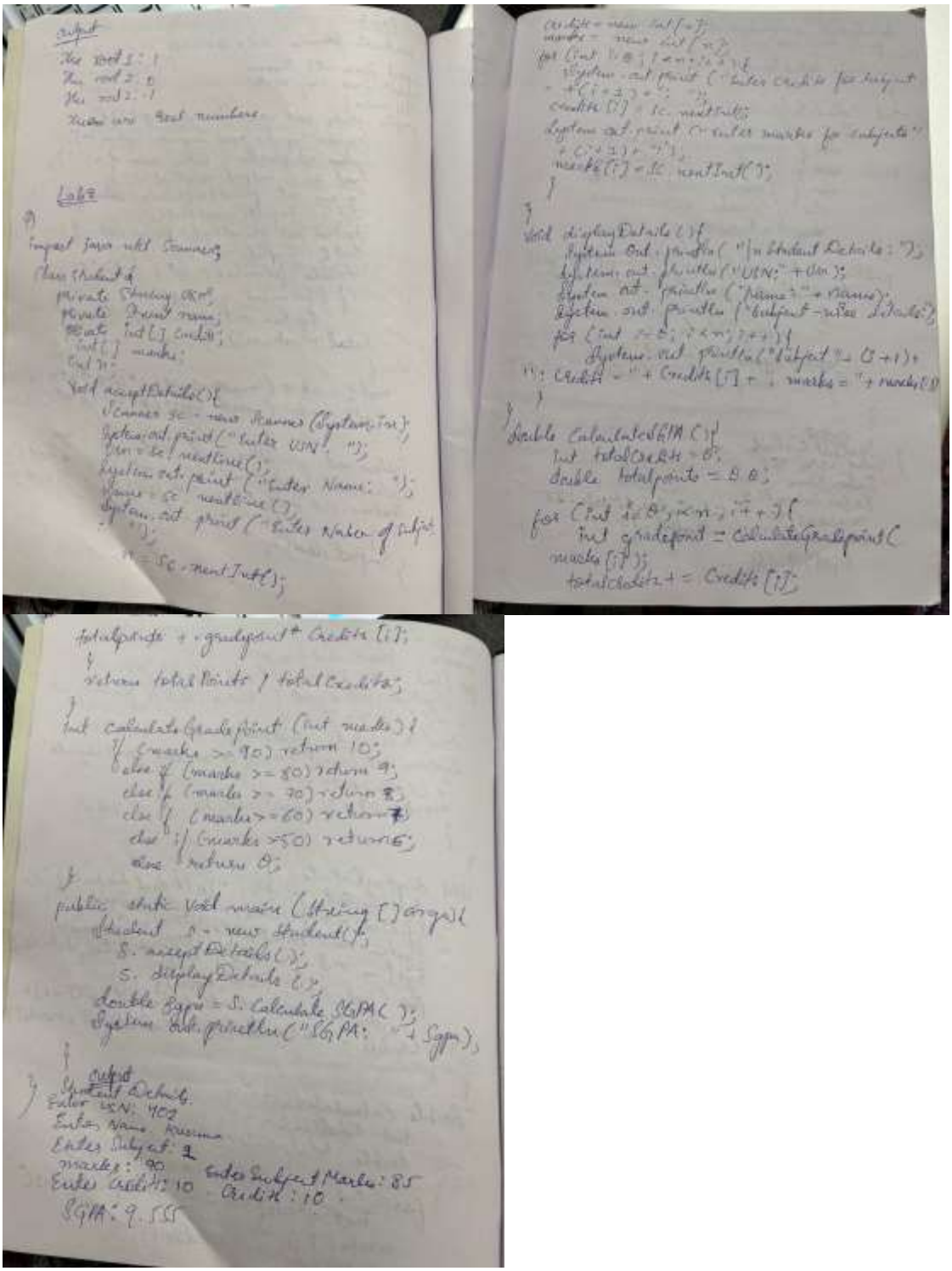
```
import java.util.Scanner;

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

      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 (discriminant > 0) {
         double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
         double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
         System.out.printf("There are two real solutions: x1 = %.2f, x2 = %.2f%n", root1, root2);
      } else if (discriminant == 0) {
         )  double root = -b / (2 * a);
         System.out.printf("There is one real solution: x = %.2f%n", root);
      } else {
         System.out.println("There are no real solutions.");
      }
scanner.close();
   }
}
```

Output:

```
E:\ooj>javac QuadraticEquationSolver.java

E:\ooj>
E:\ooj>java QuadraticEquationSolver
Enter coefficient a: 1
Enter coefficient b: -2
Enter coefficient c: 1
There is one real solution: x = 1.00

E:\ooj>java QuadraticEquationSolver
Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 2
There are two real solutions: x1 = 2.00, x2 = 1.00

E:\ooj>java QuadraticEquationSolver
Enter coefficient a: 2
Enter coefficient b: 5
Enter coefficient c: 2
There are two real solutions: x1 = -0.50, x2 = -2.00
```

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.

Algorithm:



Code:

```
import java.util.Scanner;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n;

    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.print("Enter number of subjects: ");
        n = sc.nextInt();

        credits = new int[n];
        marks = new int[n];
         for (int i = 0; i < n; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = sc.nextInt();

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        } }
 void displayDetails() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subject-wise details:");

        for (int i = 0; i < n; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + credits[i] + ", Marks = " + marks[i]);
        }
    }
 double calculateSGPA() {
        int totalCredits = 0;
        double totalPoints = 0.0;

        for (int i = 0; i < n; i++) {
            int gradePoint = calculateGradePoint(marks[i]);
            totalCredits += credits[i];
            totalPoints += gradePoint * credits[i];
        }
```

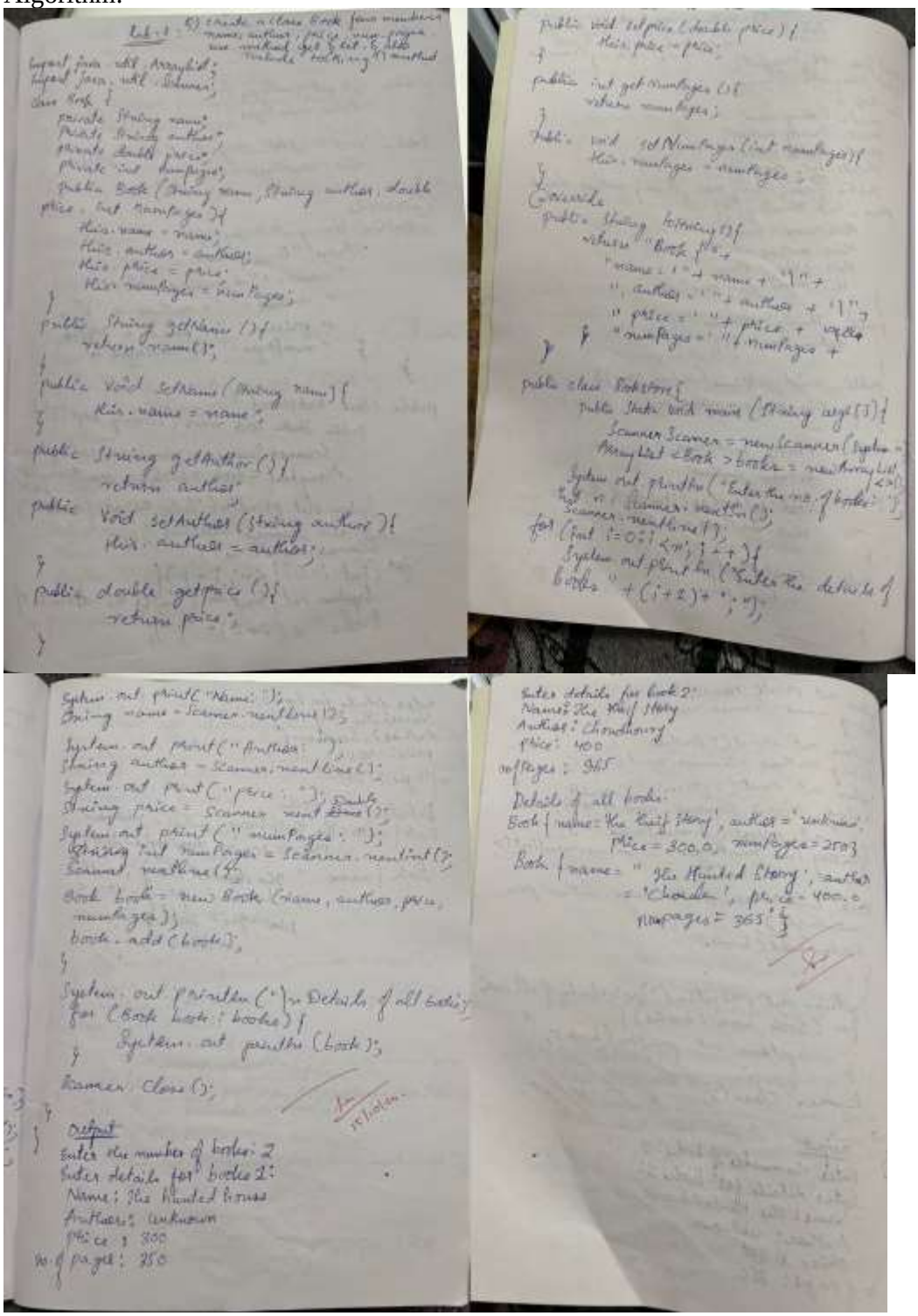
```
        return totalPoints / totalCredits;
    } int calculateGradePoint(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }

    public static void main(String[] args) {
        Student s = new Student();
        s.acceptDetails();
        s.displayDetails();

        double sgpa = s.calculateSGPA();
        System.out.println("SGPA: " + sgpa);
    }
}
```

Output:

```
E:\ooj>java Student
Enter USN: 24BECS402
Enter Name: G M kusuma
Enter number of subjects: 6
Enter credits for subject 1: 10
Enter marks for subject 1: 60
Enter credits for subject 2: 10
Enter marks for subject 2: 60
Enter credits for subject 3: 10
Enter marks for subject 3: 78
Enter credits for subject 4: 10
Enter marks for subject 4: 89
Enter credits for subject 5: 10
Enter marks for subject 5: 87
Enter credits for subject 6: 10
Enter marks for subject 6: 90

Student Details:
USN: 24BECS402
Name: G M kusuma
Subject-wise details:
Subject 1: Credits = 10, Marks = 60
Subject 2: Credits = 10, Marks = 60
Subject 3: Credits = 10, Marks = 78
Subject 4: Credits = 10, Marks = 89
Subject 5: Credits = 10, Marks = 87
Subject 6: Credits = 10, Marks = 90
SGPA: 8.333333333333334
```

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.

Algorithm:



Code:

```
import java.util.ArrayList;
import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;
  public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

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

    public String getAuthor() {
        return author;
    }

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

    public double getPrice() {
        return price;
    }

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

    public int getNumPages() {
        return numPages;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }
 @Override
```

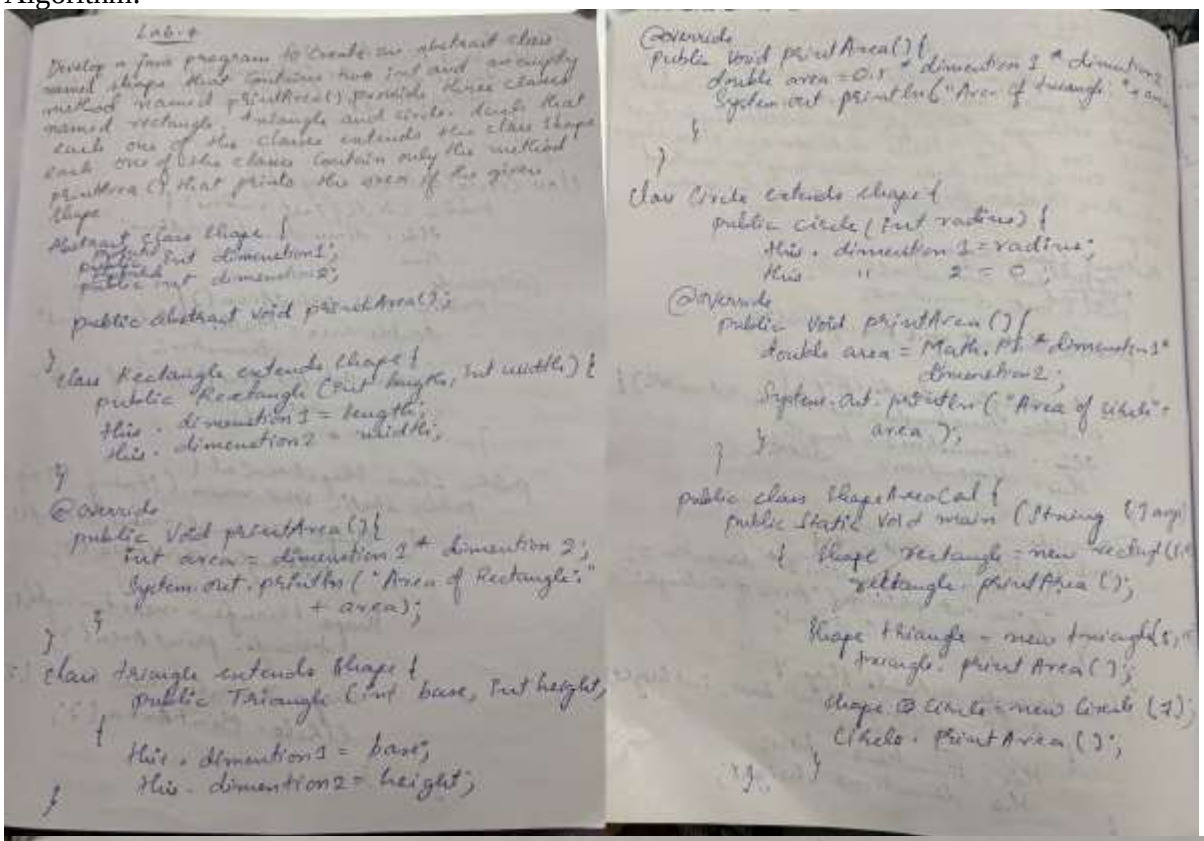
```
    public String toString() {
        return "Book{" +
                "name='" + name + '\'' +
                ", author='" + author + '\'' +
                ", price=" + price +
                ", numPages=" + numPages +
                '}';
    }
}
public class BookStore {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Book> books = new ArrayList<>();

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

        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();

            Book book = new Book(name, author, price, numPages);
            books.add(book);
        }

        System.out.println("\nDetails of all books:");
 for (Book book : books) {
            System.out.println(book);
        }

        scanner.close();
    }
}
```

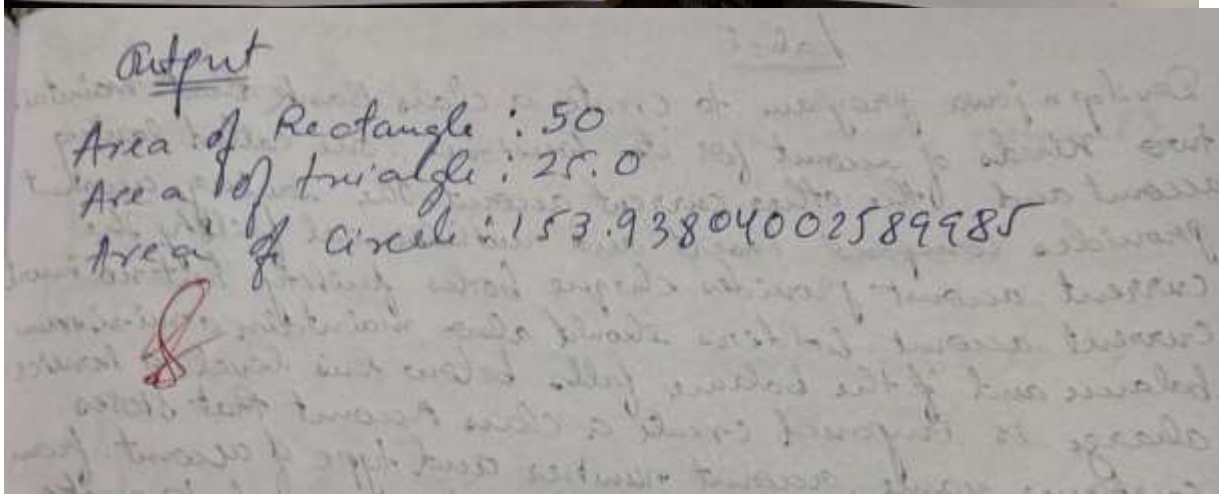
Output:

```
D:\OOJ>java BookStore
Enter the number of books: 5
Enter details for book 1:
Name: Exception in thread "main" ^C
D:\OOJ>java BookStore
Enter the number of books: 2
Enter details for book 1:
Name: the hunted house
Author: unknow
Price: 300
Number of pages: 400
Enter details for book 2:
Name: the theif story
Author: choudhoury
Price: 320
Number of pages: 278

Details of all books:
Book{name='the hunted house', author='unknow ', price=300.0, numPages=400}
Book{name='the theif story', author='choudhoury', price=320.0, numPages=278}
```

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.

Algorithm:





Code:

```
abstract class shape {
   protected int d1;
   protected int d2;public abstract void printarea();
}class rectangle extends shape {
```

```
    public rectangle(int length, int width) {
        this.d1 = length;
        this.d2 = width;
    } @Override
    public void printarea() {
        int area = d1 * d2;
        System.out.println("Area of rectangle: " + area); }
}class triangle extends shape {
    public triangle(int base, int height) {
        this.d1 = base;
        this.d2 = height;
    } @Override
    public void printarea() {
        double area = 0.5 * d1 * d2;
        System.out.println("Area of triangle: " + area);}}
class circle extends shape {
    public circle(int radius) {
        this.d1 = radius;
        this.d2 = 0; }
 @Override
    public void printarea() {
        double area = Math.PI * d1 * d1;
        System.out.println("Area of circle: " + area);}
}public class shapearea {
  public static void main(String[] args) {
        shape rectangleShape = new rectangle(5, 10);
        rectangleShape.printarea();shape triangleShape = new triangle(5, 10);
        triangleShape.printarea();
shape circleShape = new circle(7);
        circleShape.printarea();}}}
```

Output:

```
E:\ooj>javac shapearea.java

E:\ooj>java shapearea
Area of rectangle: 50
Area of triangle: 25.0
Area of circle: 153.93804002589985
```

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:

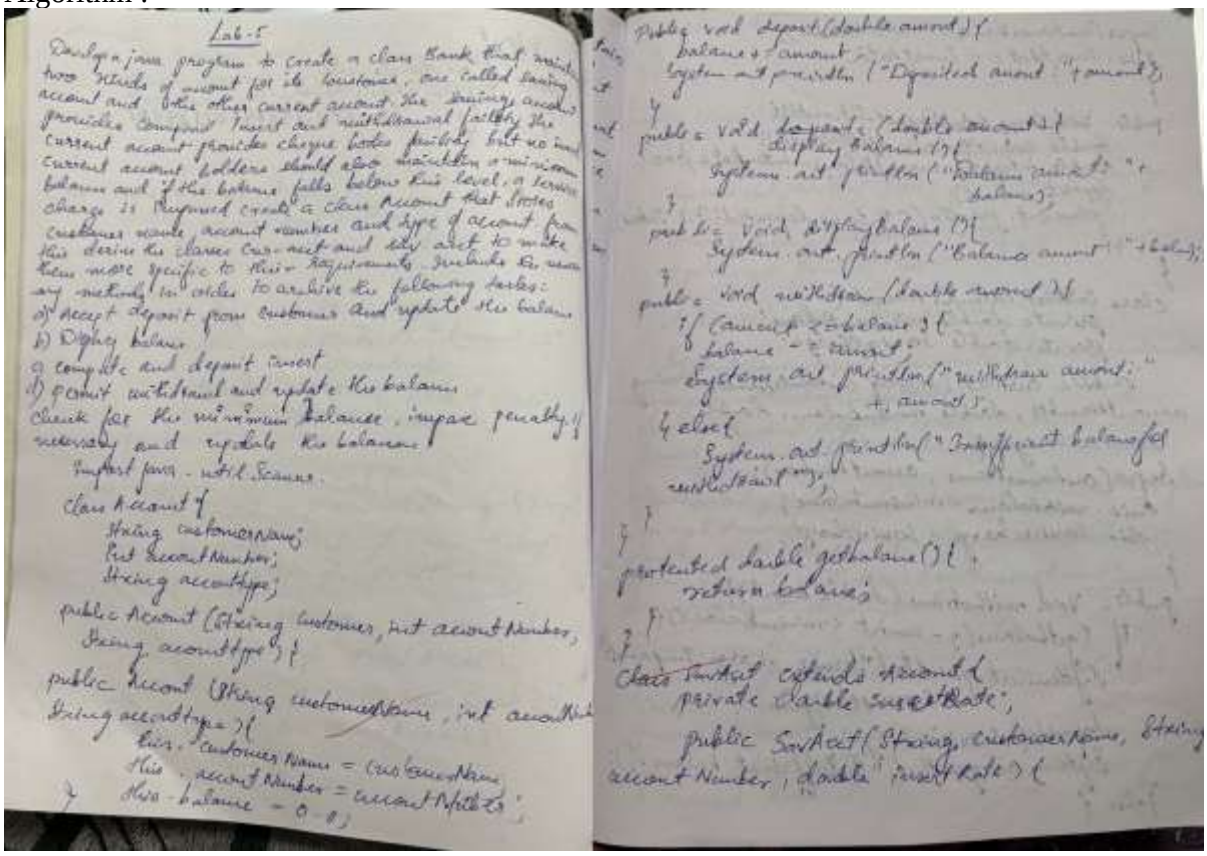
a) Accept deposit from customer and update the balance.

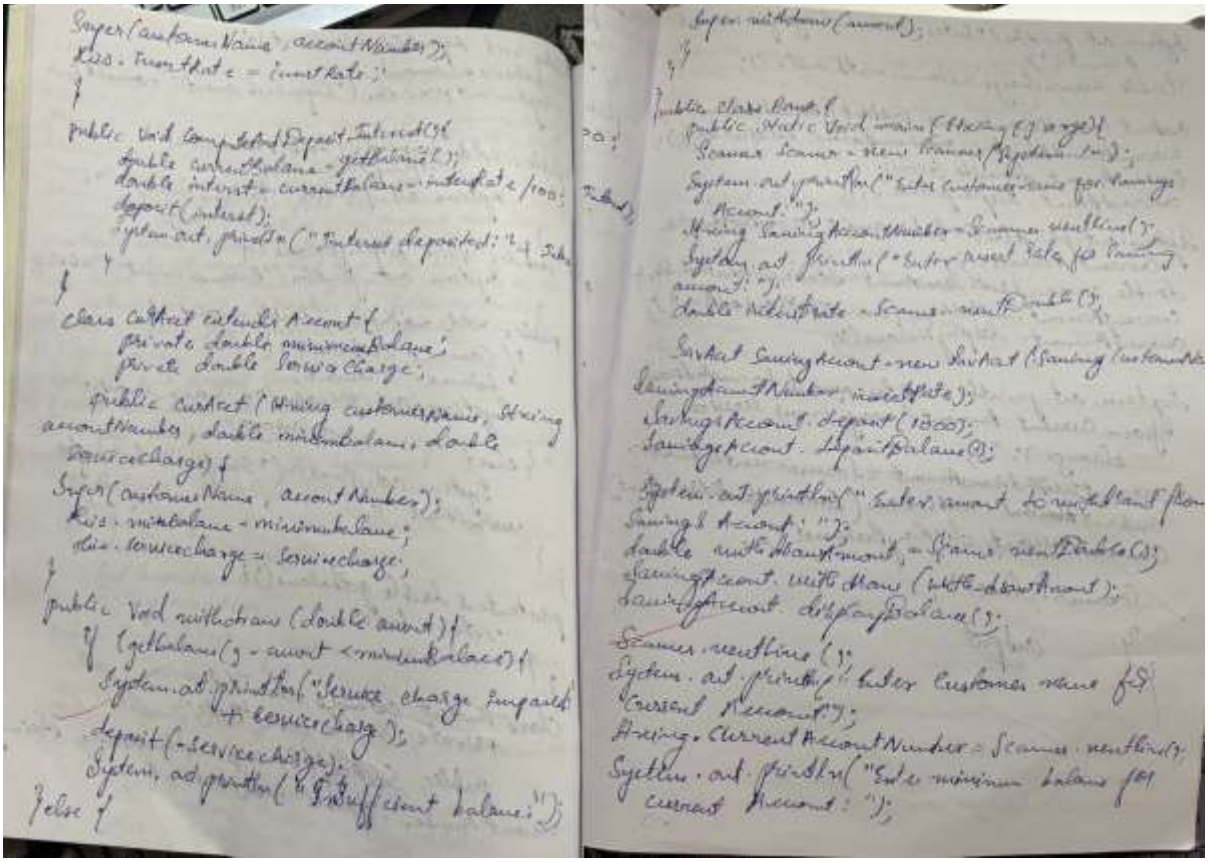
b) Display the balance.

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.

Algorithm :





```
super(customerName, accountNumber);
this.interestRate = interestRate;
}

public void computeAndDepositInterest(){
    double currentBalance = getBalance();
    double interest = currentBalance * interestRate / 100;
    deposit(interest);
    System.out.println("Interest deposited: " + interest);
}

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

    public CurrentAccount(String customerName, String
accountNumber, double minBalance, double
serviceCharge) {
super(customerName, accountNumber);
this.minimumBalance = minBalance;
this.serviceCharge = serviceCharge;
}

public void withdraw(double amount){
    if (getBalance() - amount < minimumBalance){
        System.out.println("Service charge imposed
            + serviceCharge");
        deposit(-serviceCharge);
        System.out.println("Insufficient balance");
}else {
super.withdraw(amount);
}
}

public class Bank {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter customer name for savings
            Account: ");
        String savingAccountNumber = scanner.nextLine();
        System.out.println("Enter interest rate for savings
            account: ");
        double interestRate = scanner.nextDouble();

        SavingAccount savingAccount = new SavingAccount(customerName,
savingAccountNumber, interestRate);
        savingAccount.deposit(1000);
        savingAccount.displayBalance();

        System.out.println("Enter amount to withdraw from
savings Account: ");
        double withdrawAmount = scanner.nextDouble();
        savingAccount.withdraw(withdrawAmount);
        savingAccount.displayBalance();

        scanner.nextLine();
        System.out.println("Enter customer name for
Current Account:");
        String currentCustomerName = scanner.nextLine();
        System.out.println("Enter minimum balance for
            current Account: ");
```

```
System.out.println("Enter service charge for current
    Account: ");
double serviceCharge = scanner.nextDouble();

CurrentAccount currentAccount = new CurrentAccount(currentCustomerName,
currentAccountNumber, minBalance, serviceCharge);
currentAccount.deposit(5000);
currentAccount.displayBalance();
System.out.println("Enter amount to withdraw from
    Current Account:");
double currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw
from Current Account (may incur service
    charge):");
currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();
scanner.close();
}
}
```

output

```
Enter Customer name for saving account
Rob
Enter acc number for saving account
23741976
Enter interest rate
2
Deposited Amount: 1000
Interest deposited 20.0
Balance amount 1020
Enter amount to withdraw
100
withdraw amount 100
Balance amount 920.
```

Code:

```
import java.util.Scanner;

class Account {
    private String customerName;
    private String accountNumber;
    protected double balance;
 public Account(String customerName, String accountNumber) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited amount: " + amount);
    }

    public void displayBalance() {
        System.out.println("Balance amount: " + balance);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw amount: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal!");
        }
    }protected double getBalance() {
        return balance;
    }
}
class SavAcct extends Account {
    private double interestRate;

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

    public void computeAndDepositInterest() {
        double currentBalance = getBalance();
        double interest = currentBalance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest deposited: " + interest);
    }
}class CurAcct extends Account {
    private double minimumBalance;
```

```
    private double serviceCharge;

    public CurAcct(String customerName, String accountNumber, double minimumBalance, double
serviceCharge) {
        super(customerName, accountNumber);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }public void withdraw(double amount) {
        if (getBalance() - amount < minimumBalance) {
            System.out.println("Service charge imposed: " + serviceCharge);
            deposit(-serviceCharge);
            System.out.println("Insufficient balance.");
        } else {
            super.withdraw(amount);
        }
    }
}public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
System.out.println("Enter customer name for Savings Account:");
        String savingsCustomerName = scanner.nextLine();
        System.out.println("Enter account number for Savings Account:");
        String savingsAccountNumber = scanner.nextLine();
        System.out.println("Enter interest rate for Savings Account:");
        double interestRate = scanner.nextDouble();
 SavAcct savingsAccount = new SavAcct(savingsCustomerName, savingsAccountNumber, interestRate);
        savingsAccount.deposit(1000);
        savingsAccount.computeAndDepositInterest();
        savingsAccount.displayBalance();
System.out.println("Enter amount to withdraw from Savings Account:");
        double withdrawAmount = scanner.nextDouble();
        savingsAccount.withdraw(withdrawAmount);
        savingsAccount.displayBalance();
 scanner.nextLine();
        System.out.println("Enter customer name for Current Account:");
        String currentCustomerName = scanner.nextLine();
        System.out.println("Enter account number for Current Account:");
        String currentAccountNumber = scanner.nextLine();
        System.out.println("Enter minimum balance for Current Account:");
        double minimumBalance = scanner.nextDouble();
        System.out.println("Enter service charge for Current Account:");
        double serviceCharge = scanner.nextDouble();
  CurAcct currentAccount = new CurAcct(currentCustomerName, currentAccountNumber,
minimumBalance, serviceCharge);
        currentAccount.deposit(2000);
        currentAccount.displayBalance();

        System.out.println("Enter amount to withdraw from Current Account:");
        double currentWithdrawAmount = scanner.nextDouble();
```

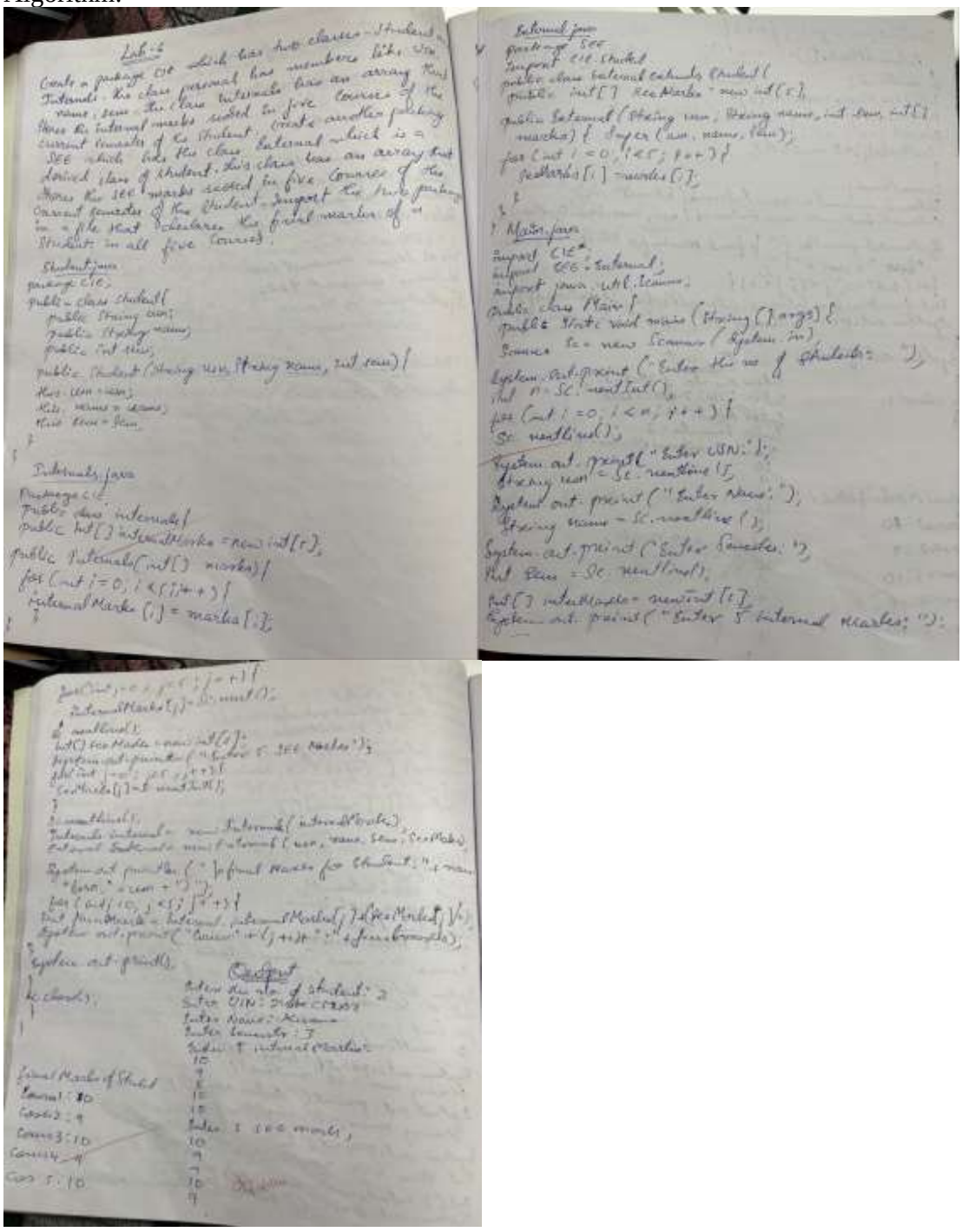
```
        currentAccount.withdraw(currentWithdrawAmount);
        currentAccount.displayBalance();
System.out.println("Enter amount to withdraw from Current Account (may incur service charge):");
        currentWithdrawAmount = scanner.nextDouble();
        currentAccount.withdraw(currentWithdrawAmount);
        currentAccount.displayBalance();

        scanner.close();
    }
}
```

Output:

```
E:\ooj>javac Bank.java

E:\ooj>java Bank
Enter customer name for Savings Account:
kusuma
Enter account number for Savings Account:
567894921
Enter interest rate for Savings Account:
2
Deposited amount: 1000.0
Deposited amount: 20.0
Interest deposited: 20.0
Balance amount: 1020.0
Enter amount to withdraw from Savings Account:
100
Withdraw amount: 100.0
Balance amount: 920.0
Enter customer name for Current Account:
gm kusumaa
Enter account number for Current Account:
5678903284
Enter minimum balance for Current Account:
500
Enter service charge for Current Account:
2
Deposited amount: 2000.0
Balance amount: 2000.0
Enter amount to withdraw from Current Account:
100
Withdraw amount: 100.0
Balance amount: 1900.0
Enter amount to withdraw from Current Account (may incur service charge):
271
Withdraw amount: 271.0
Balance amount: 1629.0
```

Program 6:
Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Algorithm:



Code:

```
// CIE/Student.java
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;
    }
}
// CIE/Internals.java
package CIE;

public class Internals {
public int[] internalMarks = new int[5];

public Internals(int[] marks) {
for (int i = 0; i < 5; i++) {
 internalMarks[i] = marks[i];
        }
    }
}
// SEE/External.java
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] seeMarks = new int[5];

    public External(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        for (int i = 0; i < 5; i++) {
            seeMarks[i] = marks[i];
        }
    }
}

// Main.java
import CIE.*;
import SEE.External;
import java.util.Scanner;

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

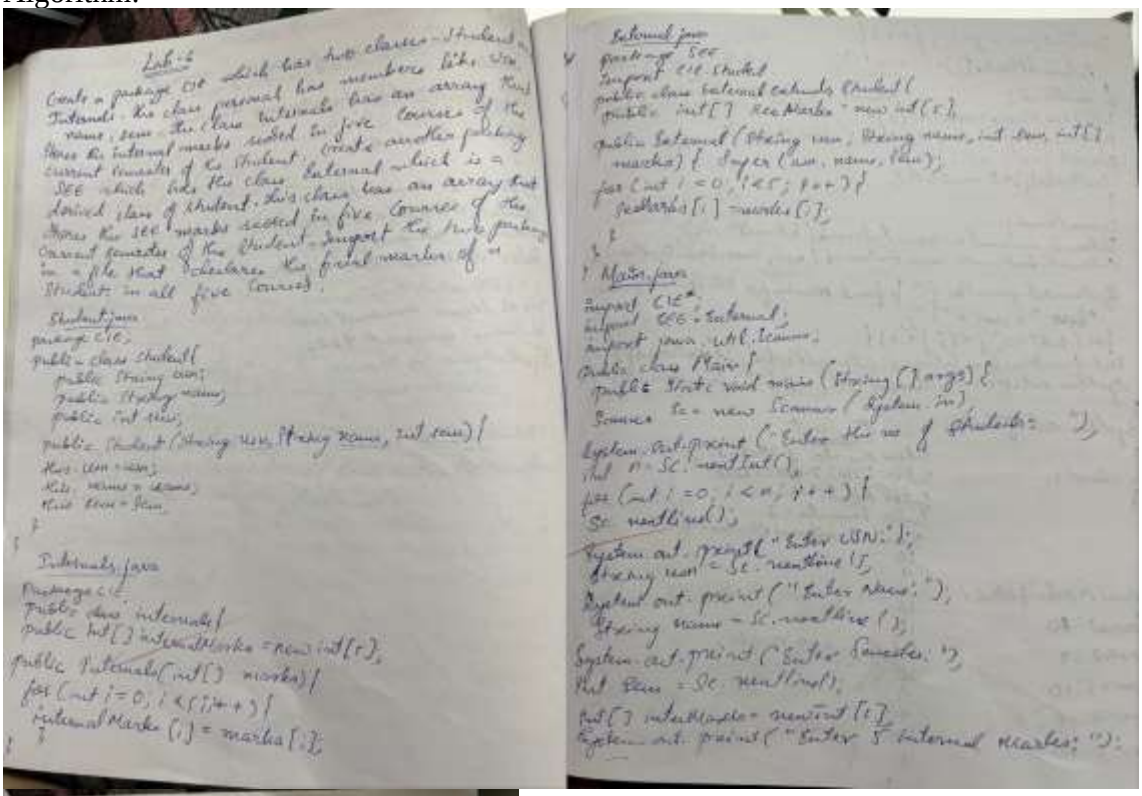
```
Scanner sc = new Scanner(System.in);

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

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

 System.out.print("Enter USN: ");
 String usn = sc.nextLine();

System.out.print("Enter Name: ");
String name = sc.nextLine();

 System.out.print("Enter Semester: ");
int sem = sc.nextInt();

int[] internalMarks = new int[5];
System.out.println("Enter 5 internal marks:");
 for (int j = 0; j < 5; j++) {
internalMarks[j] = sc.nextInt();
}
sc.nextLine();
int[] seeMarks = new int[5];
System.out.println("Enter 5 SEE marks:");
for (int j = 0; j < 5; j++) {
seeMarks[j] = sc.nextInt();
            }
sc.nextLine();

Internals internal = new Internals(internalMarks);
External external = new External(usn, name, sem, seeMarks);

 System.out.println("\nFinal Marks for Student: " + name + " (USN: " + usn + ")");
for (int j = 0; j < 5; j++) {
int finalMark = internal.internalMarks[j] + (seeMarks[j] / 2);
 System.out.println("Course " + (j + 1) + ": " + finalMark);
}
System.out.println();
}
 sc.close();
    }
}
```

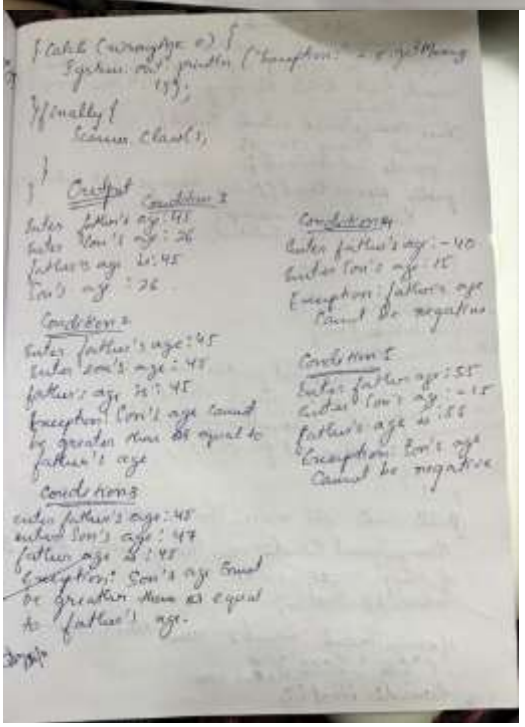
Ouput:

```
E:\javaprograms>java Main1
Enter the number of students: 2
Enter USN: 24becs402
Enter Name: kusuma
Enter Semester: 3
Enter 5 internal marks:
5
6
7
8
9
Enter 5 SEE marks:
4
8
9
0
23

Final Marks for Student: kusuma (USN: 24becs402)
Course 1: 7
Course 2: 10
Course 3: 11
Course 4: 8
Course 5: 20
```

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=father's age.

Algorithm:





Code:

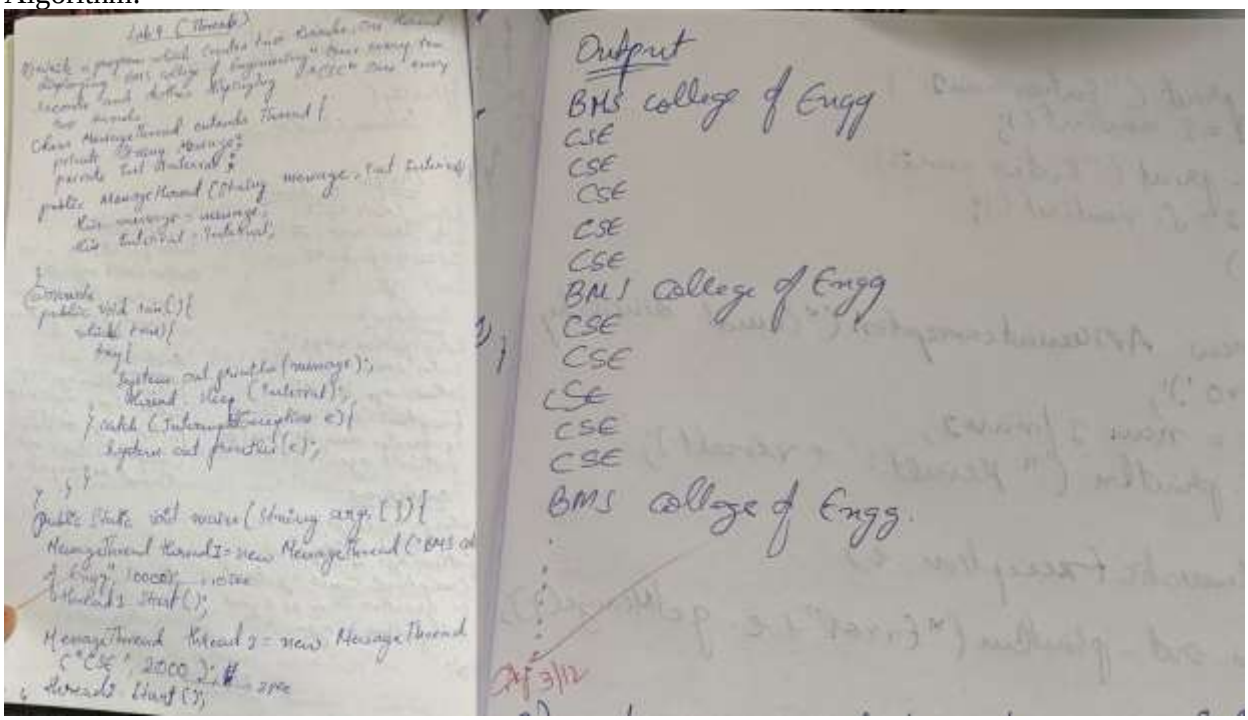
```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}class Father {
    int fatherAge;
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Father's age cannot be negative.");
        }
        this.fatherAge = age;
        System.out.println("Father's age is: " + fatherAge);
    }
}
class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Call the constructor of Father
        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age is: " + sonAge);
    }
}public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

Output:

```
E:\ooj>javac Main.java

E:\ooj>java Main.java
Enter Father's age: 45
Enter Son's age: 26
Father's age is: 45
Son's age is: 26

E:\ooj>java Main.java
Enter Father's age: 45
Enter Son's age: 45
Father's age is: 45
Exception: Son's age cannot be greater than or equal to Father's age.

E:\ooj>java Main.java
Enter Father's age: -40
Enter Son's age: 23
Exception: Father's age cannot be negative.

E:\ooj>java Main.java
Enter Father's age: 65
Enter Son's age: -34
Father's age is: 65
Exception: Son's age cannot be negative.
```

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.

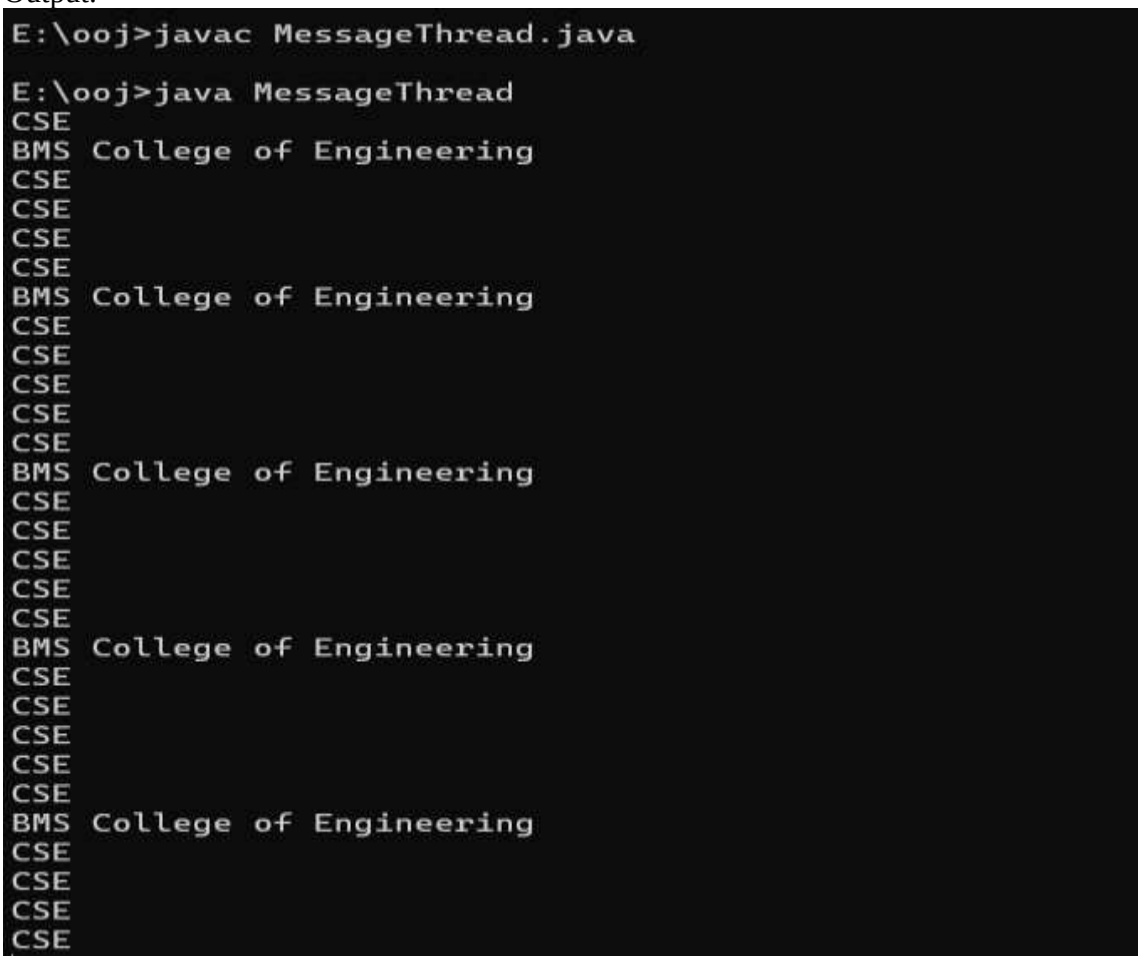
Algorithm:



Code:

```
class MessageThread extends Thread {
    private String message;
    private int interval;
  public MessageThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }@Override
    public void run() {
        while (true) { try {
                System.out.println(message);
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                System.out.println(e);}} }public static void main(String[] args) {
        MessageThread thread1 = new MessageThread("BMS College of Engineering", 10000); // 10000
milliseconds = 10 seconds
        thread1.start();
 MessageThread thread2 = new MessageThread("CSE", 2000); // 2000 milliseconds = 2 seconds
        thread2.start();}}}
```

Output:



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.

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class IntegerDivisionApp {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Integer Division App");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);

        // Create the main panel
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2, 10, 10));

        // Create and add components to the panel
        JLabel labelNum1 = new JLabel("Num1:");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Num2:");
        JTextField textNum2 = new JTextField();
        JLabel labelResult = new JLabel("Result:");
        JTextField textResult = new JTextField();
        textResult.setEditable(false); // Make the result field read-only
        JButton divideButton = new JButton("Divide");

        panel.add(labelNum1);
        panel.add(textNum1);
        panel.add(labelNum2);
        panel.add(textNum2);
        panel.add(labelResult);
        panel.add(textResult);
        panel.add(divideButton);

        // Add action listener to the divide button
        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    // Parse the input numbers
                    int num1 = Integer.parseInt(textNum1.getText());
```

```
                int num2 = Integer.parseInt(textNum2.getText());

                // Perform division and display the result
                int result = num1 / num2;
                textResult.setText(String.valueOf(result));
            } catch (NumberFormatException ex) {
                // Handle invalid number format
                JOptionPane.showMessageDialog(frame, "Please enter valid integers for Num1 and Num2.",
"Invalid Input", JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException ex) {
                // Handle division by zero
                JOptionPane.showMessageDialog(frame, "Division by zero is not allowed.", "Arithmetic Error",
JOptionPane.ERROR_MESSAGE);
            }
        }
    });

    // Add the panel to the frame
    frame.add(panel);

    // Make the frame visible
    frame.setVisible(true);
  }
}
```

Output: