VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Vaibhav Dhar (1BM23CD068)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)

BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Vaibhav Dhar (1BM23CD068),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

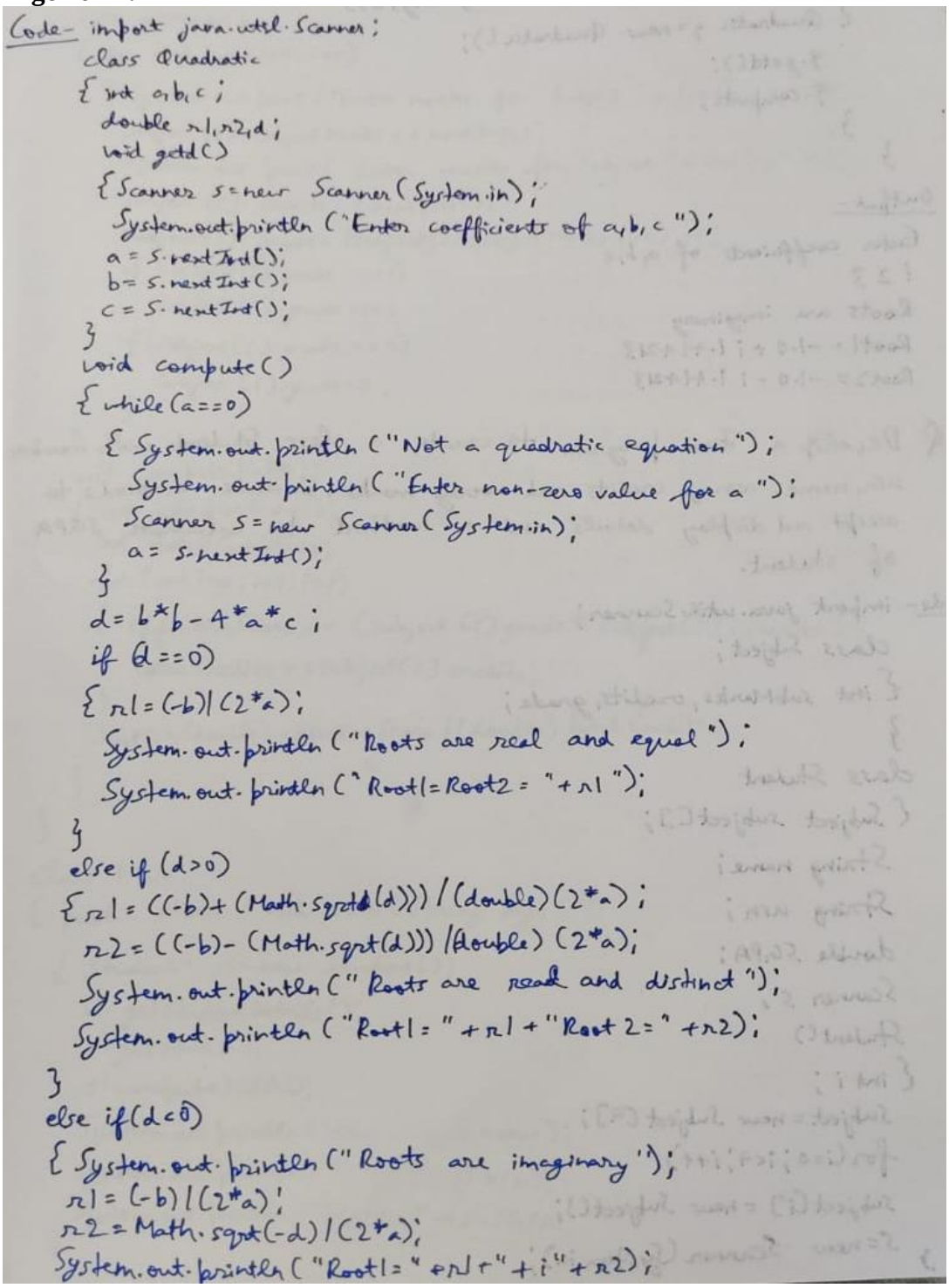
Dr. Seema Patil Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

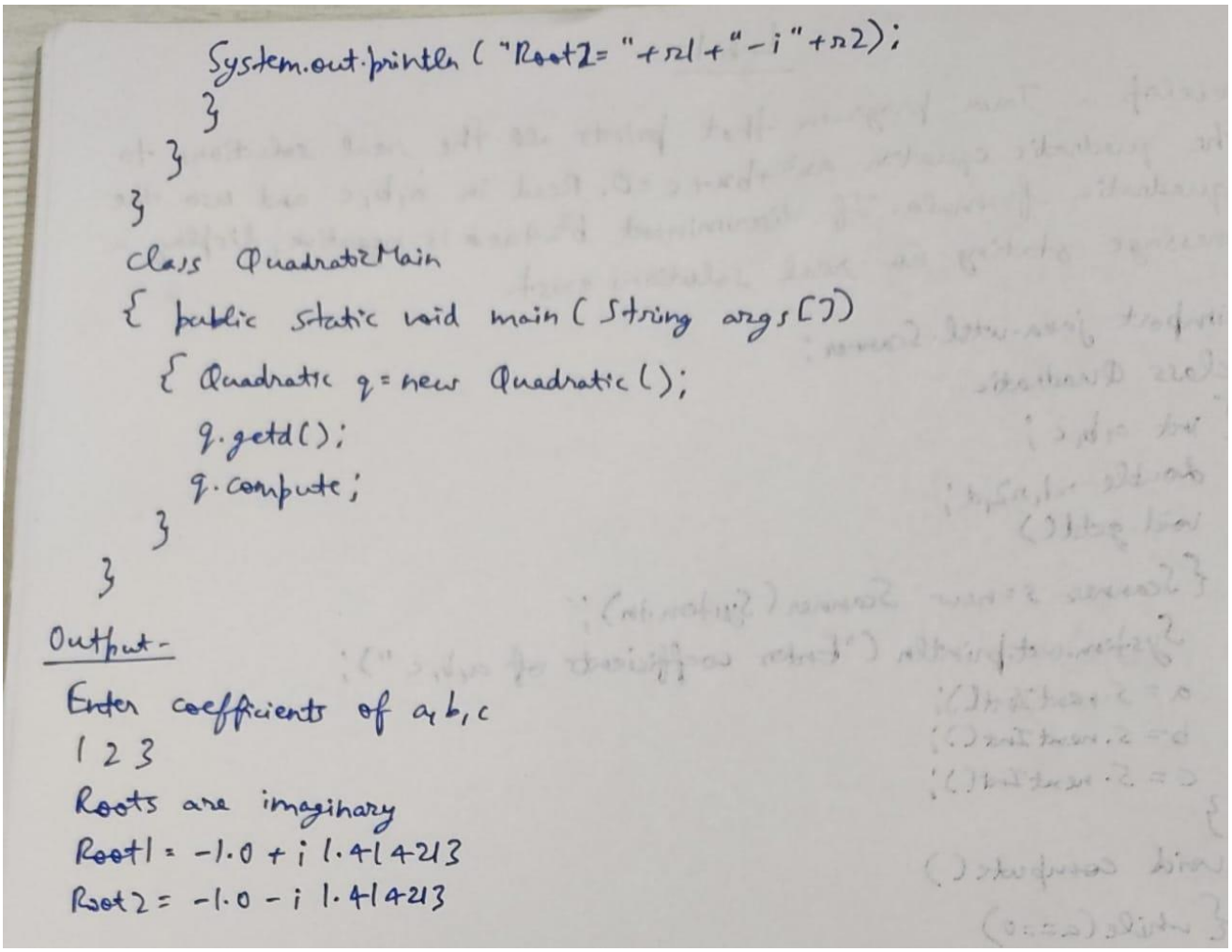
Github Link:
https://github.com/Vaibhav1830/Java-Lab

Program 1
Implement Quadratic Equation

Algorithm:



```
Code- import java.util.Scanner;
class Quadratic
{ int a,b,c;
  double r1,r2,d;
  void getd()
  {Scanner s=new Scanner(System.in);
   System.out.println("Enter coefficients of a,b,c ");
   a=s.nextInt();
   b=s.nextInt();
   c=s.nextInt();
  }
  void compute()
  {while(a==0)
    { System.out.println("Not a quadratic equation");
      System.out.println("Enter non-zero value for a ");
      Scanner s=new Scanner(System.in);
      a=s.nextInt();
    }
    d=b*b-4*a*c;
    if(d==0)
    {r1=(-b)/(2*a);
     System.out.println("Roots are real and equal");
     System.out.println("Root1=Root2="+r1");
    }
    else if(d>0)
   {r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
    r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
    System.out.println("Roots are readl and distinct");
    System.out.println("Root1="+r1+"Root 2="+r2);
  }
  else if(d<0)
  {System.out.println("Roots are imaginary");
   r1=(-b)/(2*a);
   r2=Math.sqrt(-d)/(2*a);
   System.out.println("Root1="+r1+"+i"+r2);
```



Code:

```
import java.util.Scanner;
class Quadratic
{ int a, b, c;
   double r1, r2, d;
   void getd()
   {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the coefficients of a,b,c");
      a = s.nextInt(); b = s.nextInt(); c = s.nextInt();
   }
   void compute()
   {
      while(a==0)
      {
         System.out.println("Not a quadratic equation");
         System.out.println("Enter a non zero value for a:");
         Scanner s = new Scanner(System.in);
```

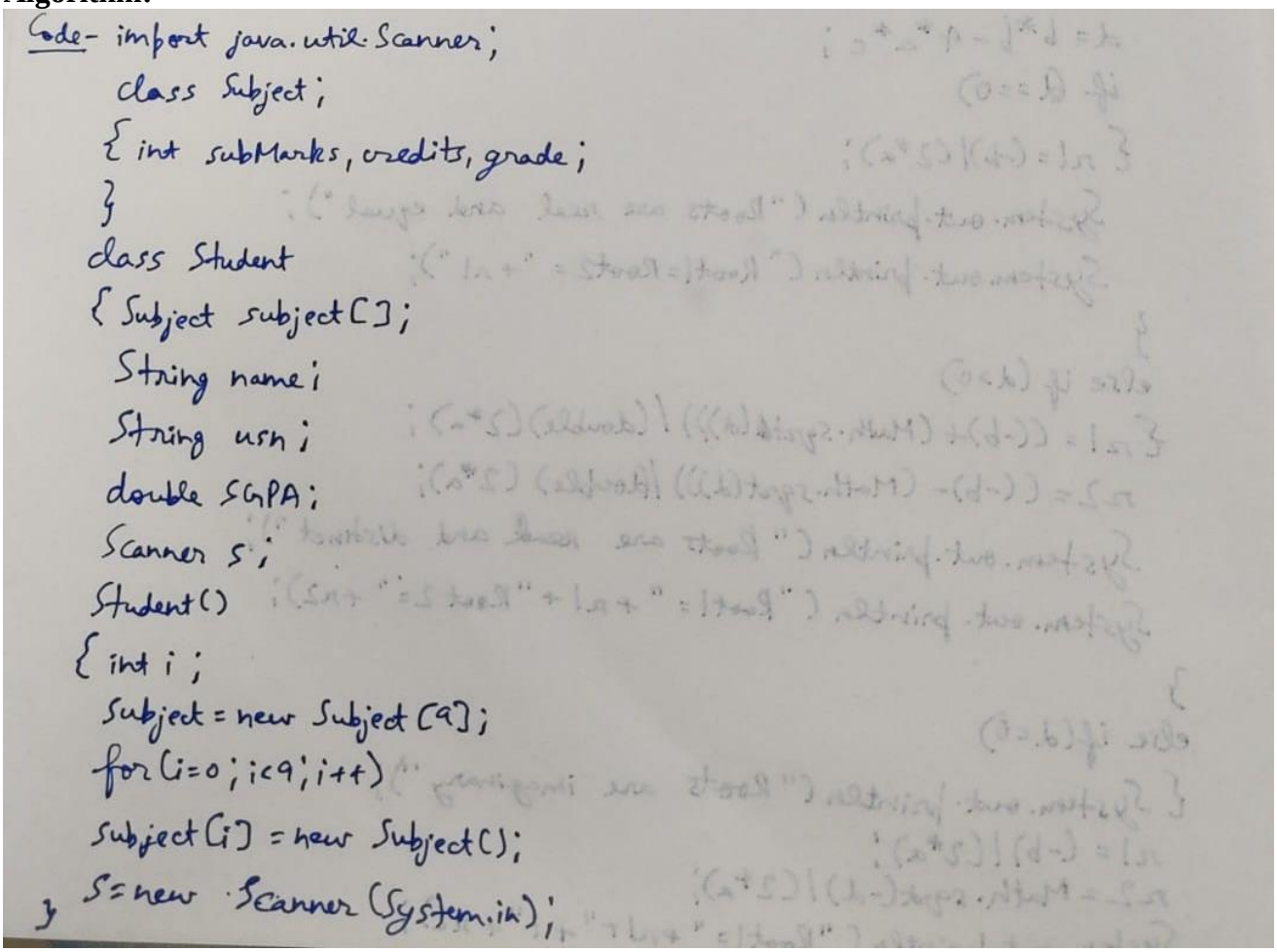
```
        a = s.nextInt();
      }
      d = b*b-4*a*c;
      if(d==0)
      {
        r1 = (-b)/(2*a);
        System.out.println("Roots are real and equal");
        System.out.println("Root1 = Root2 = " + r1);
      }
      else if(d>0)
      {
        r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
        r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
        System.out.println("Roots are real and distinct");
        System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
      }
      else if(d<0)
      {
        System.out.println("Roots are imaginary");
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println("Root1 = " + r1 + " + i"+r2);
        System.out.println("Root1 = " + r1 + " - i"+r2);
      }
   }
}
class QuadraticMain
{
   public static void main(String args[])
   {
      Quadratic q = new Quadratic();
      q.getd();
      q.compute();
   }
}
```

```
"C:\Program Files\Java\jdk-23\bin\
Enter the coefficients of a,b,c
2 3 1
Roots are real and distinct
Roo1 = -0.5 Root2 = -1.0

Process finished with exit code 0
```

```
"C:\Program Files\Java\jdk-23\bin\
Enter the coefficients of a,b,c
1 2 3
Roots are imaginary
Root1 = -1.0 + i1.4142135623730951
Root1 = -1.0 - i1.4142135623730951

Process finished with exit code 0
```

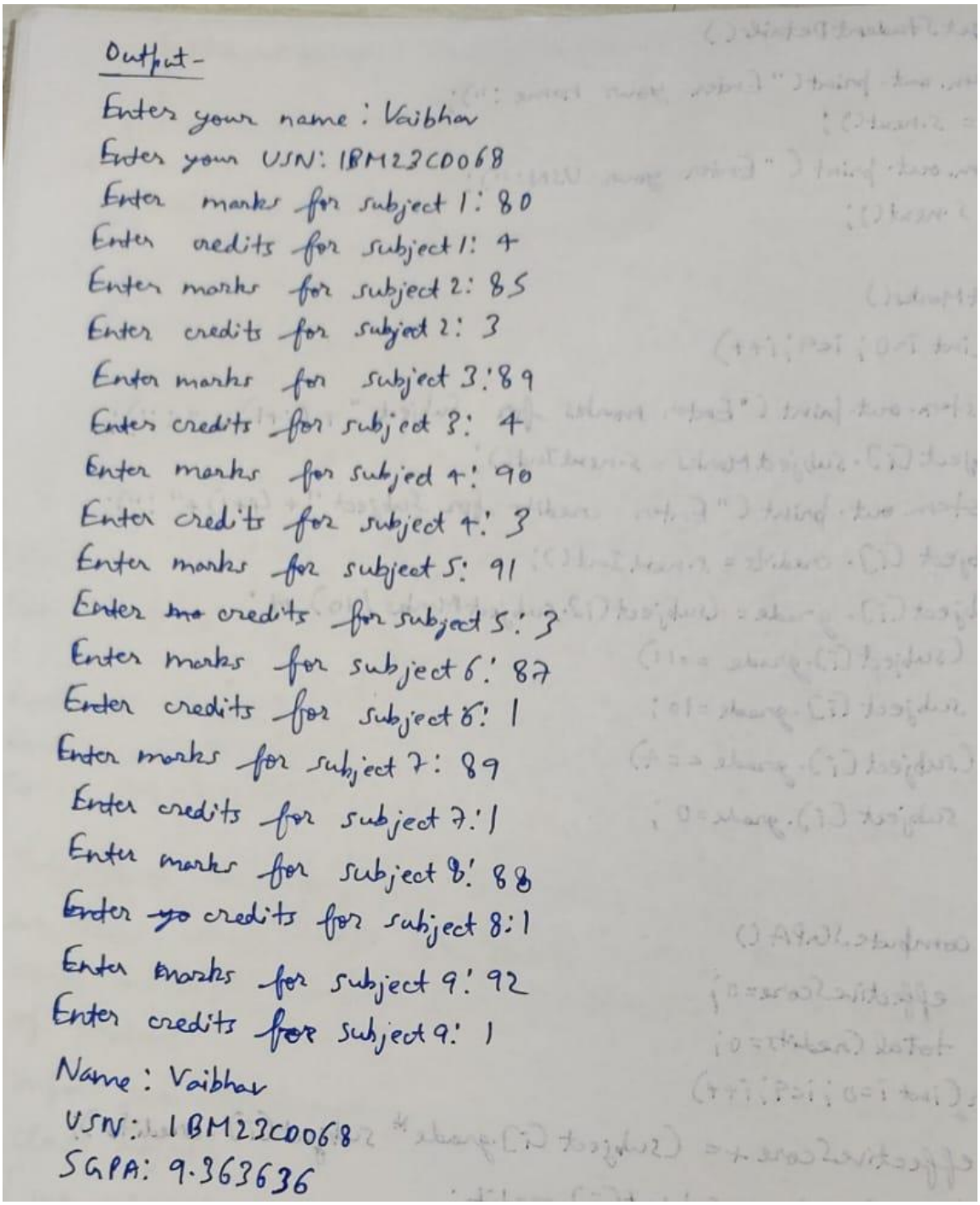
Program 2
SGPA Calculation

Algorithm:



```
Code- import java.util.Scanner;
      class Subject;
      { int subMarks, credits, grade;
      }
      class Student
      { Subject subject[];
        String name;
        String usn;
        double SGPA;
        Scanner s;
        Student()
      { int i;
        Subject = new Subject[9];
        for(i=0;i<9;i++)
        subject[i] = new Subject();
        S=new Scanner(System.in);
      }
```

```
    void getStudentDetails()
    { System.out.print("Enter your name :");
      name = s.next();
      System.out.print("Enter your USN:");
      usn = s.next();
    }
    void getMarks()
    { for (int i=0; i<9; i++)
      { System.out.print("Enter marks for Subject " + (i+1) + " :");
        Subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits for Subject " + (i+1) + " :");
        Subject[i].credits = s.nextInt();
        Subject[i].grade = (Subject[i].subjectMarks / 10) + 1;
        if (Subject[i].grade == 11)
          Subject[i].grade = 10;
        if (Subject[i].grade <= 4)
          Subject[i].grade = 0;
      }
    }
    void computeSGPA()
    { int effectiveScore = 0;
      int totalCredits = 0;
      for (int i=0; i<9; i++)
      { effectiveScore += (Subject[i].grade * Subject[i].credits);
        totalCredits += Subject[i].credits;
      }
      SGPA = (double) effectiveScore / (double) totalCredits;
    }
}
Class Main
{ public static void main(String args[])
  { Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();
    System.out.println("Name:" + s1.name);
    System.out.println("USN: " + s1.usn);
    System.out.println("SGPA:" + s1.SGPA);
  }
}
```



Code:

```
import java.util.Scanner;
class Subject
{
   int subjectMarks;
   int credits;
   int grade;
}
class Student
```

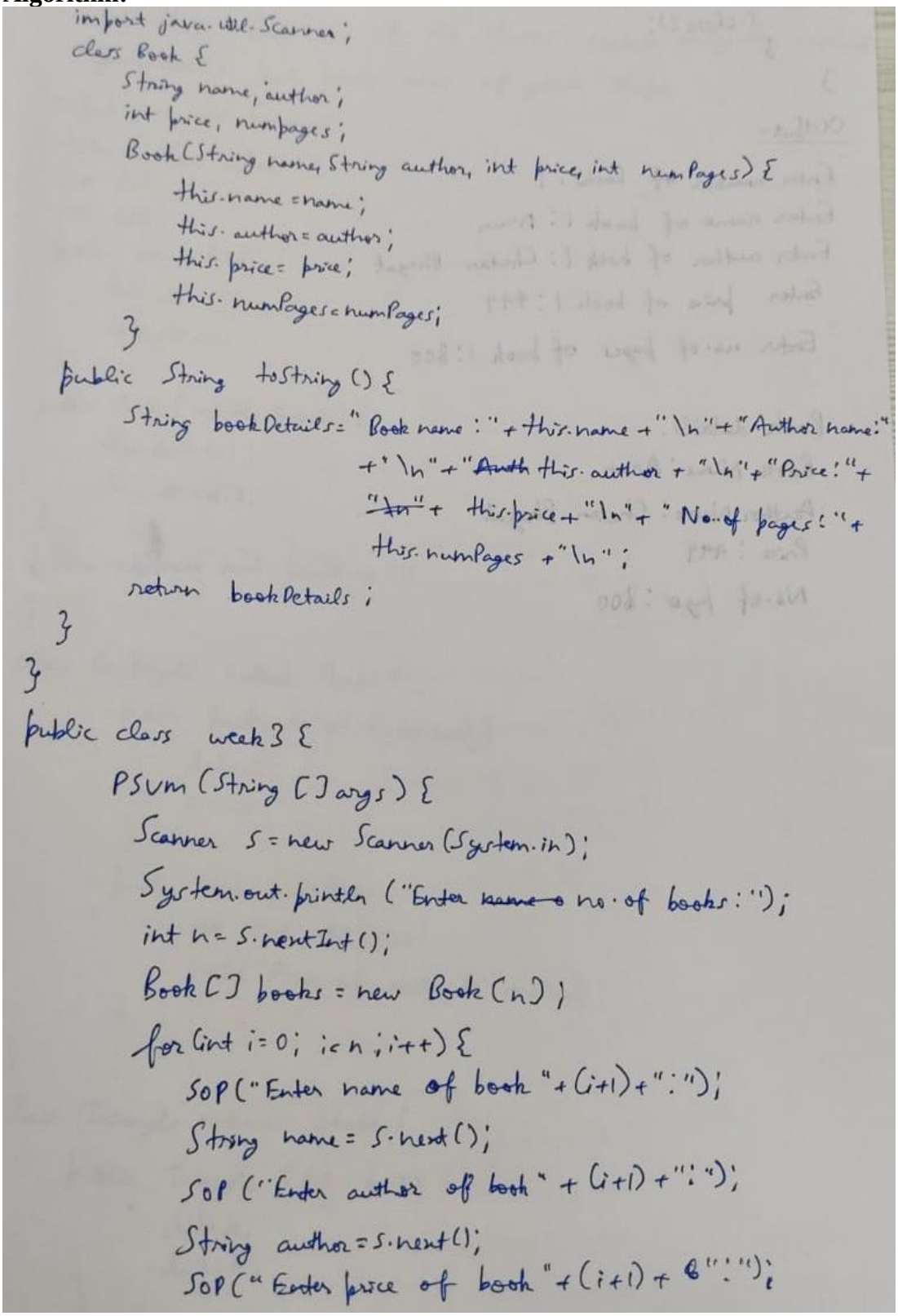
```
{
   Subject subject[];
   String name;
   String usn;
   double SGPA;
   Scanner s;
   Student()
   {
      int i;
      subject = new Subject[9];
      for(i=0;i<9;i++)
         subject[i] = new Subject();
      s = new Scanner(System.in);
   }
   void getStudentDetails()
   {
      System.out.print("Enter your Name: ");
      name = s.next();
      System.out.print("Enter your USN: ");
      usn = s.next();
   }
   void getMarks()
   {
      for(int i=0;i<9;i++)
      {
         System.out.print("Enter marks for subject "+(i+1)+" :");

         subject[i].subjectMarks = s.nextInt();
         System.out.print("Enter credits for subject "+(i+1)+" :");

         subject[i].credits = s.nextInt();
         subject[i].grade = (subject[i].subjectMarks/10) + 1;
         if(subject[i].grade==11)
            subject[i].grade = 10;
         if(subject[i].grade<=4)
            subject[i].grade = 0;

      }
   }
   void computeSGPA()
   {
      int effectiveScore = 0;
      int totalCredits = 0;
      for(int i=0;i<9;i++)
      {
         effectiveScore += (subject[i].grade*subject[i].credits);
         totalCredits += subject[i].credits;
```

```
        }
        SGPA = (double)effectiveScore/(double)totalCredits;
    }
}
class Student_SGPA
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Name: "+s1.name);
        System.out.println("USN: "+s1.usn);
        System.out.println("SGPA: "+s1.SGPA);
    }
}
```

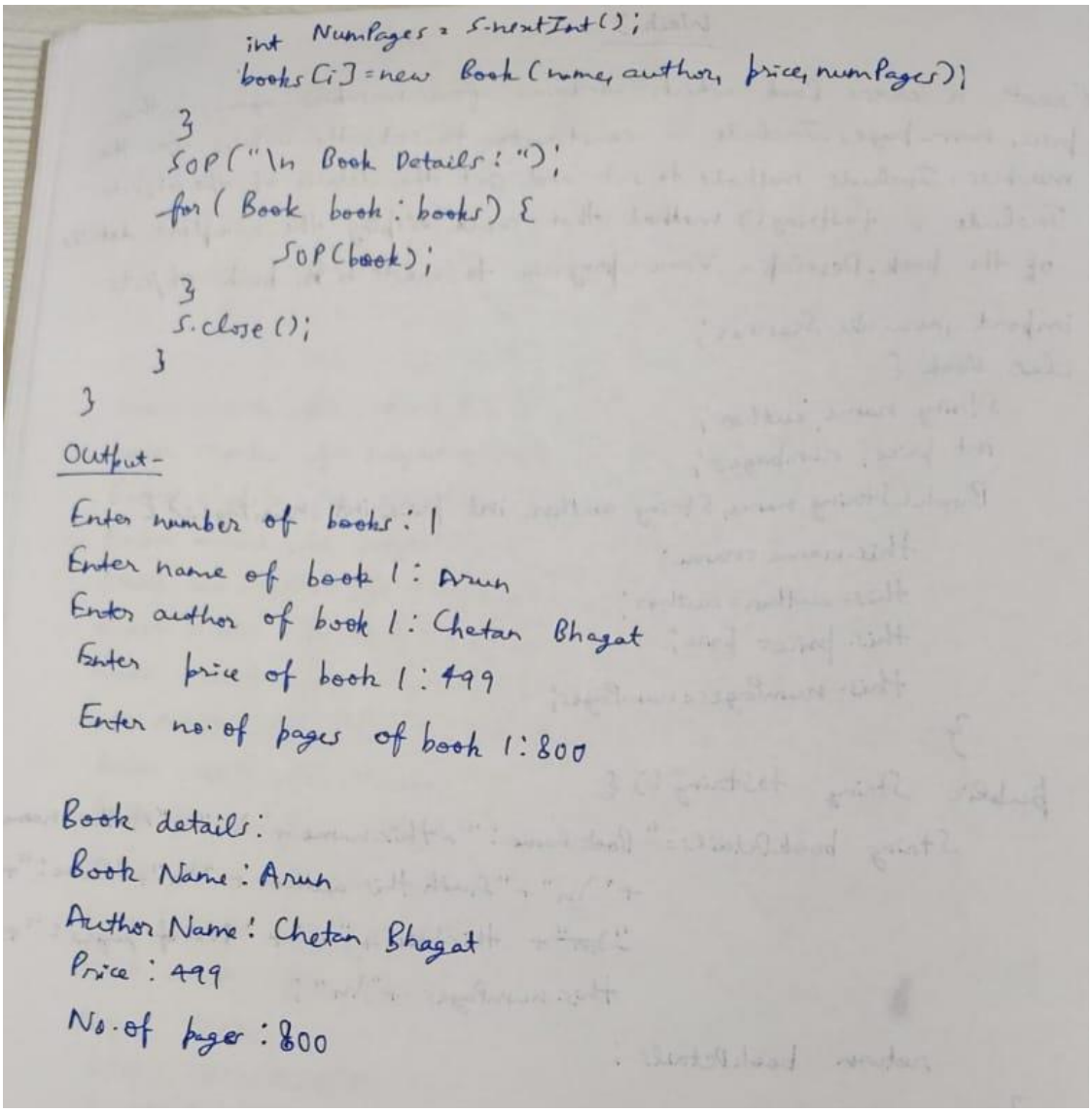
```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Enter your Name: Vaibhav
Enter your USN: 1BM23CD068
Enter marks for subject 1 :82
Enter credits for subject 1 :4
Enter marks for subject 2 :85
Enter credits for subject 2 :4
Enter marks for subject 3 :91
Enter credits for subject 3 :3
Enter marks for subject 4 :88
Enter credits for subject 4 :3
Enter marks for subject 5 :94
Enter credits for subject 5 :3
Enter marks for subject 6 :99
Enter credits for subject 6 :1
Enter marks for subject 7 :87
Enter credits for subject 7 :1
Enter marks for subject 8 :91
Enter credits for subject 8 :1
Enter marks for subject 9 :87
Enter credits for subject 9 :1
Name: Vaibhav
USN: 1BM23CD068
SGPA: 9.380952380952381

Process finished with exit code 0
```

Program 3
Display Book Details

Algorithm:



```
import java.util.Scanner;
class Book {
    String name, author;
    int price, numpages;
    Book(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
public String toString() {
    String bookDetails = "Book name : " + this.name + "\n" + "Author name:"
                         + "\n" + "Auth this.author + "\n" + "Price:" +
                         "\n" + this.price + "\n" + "No. of pages : " +
                         this.numPages + "\n";
    return bookDetails;
  }
}
public class week3 {
    PSVM(String[] args) {
     Scanner s = new Scanner(System.in);
     System.out.println("Enter name no. of books:");
     int n = s.nextInt();
     Book[] books = new Book(n);
     for(int i=0; i<n; i++) {
        SOP("Enter name of book " + (i+1) + ":");
        String name = s.next();
        SOP("Enter author of book " + (i+1) + ":");
        String author = s.next();
        SOP("Enter price of book " + (i+1) + ":");
```



Code:

```
import java.util.Scanner  ;

class Main{
   public static void main(String args[]){
      int n ;
      System.out.print("Enter the number of books:") ;
      Scanner sc  = new Scanner(System.in) ; n = sc.nextInt() ;
      sc.nextLine() ;
      Book books[] = new Book[n];
      for(int i = 0 ; i<n ; i++){
         System.out.print("Enter the book name: ") ;
         String name = sc.nextLine() ;

         System.out.print("Enter the author name: ") ;
```

```
            String author = sc.nextLine()  ;
            System.out.print("Enter the price of the book: ") ;
            int price = sc.nextInt() ;
            System.out.print("Enter the number of pages in the book: ") ;
            int numPages = sc.nextInt() ;
            sc.nextLine() ;

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

        System.out.println("");
        for(int i = 0 ; i<n ; i++){
            System.out.println(books[i].toString()) ;
        }
        System.out.println("Vaibhav Dhar" )  ;
        System.out.print("1BM23CD068") ;
        sc.close();
    }
}


class Book{
    String name , author  ;
    int price , numPages ;

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

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

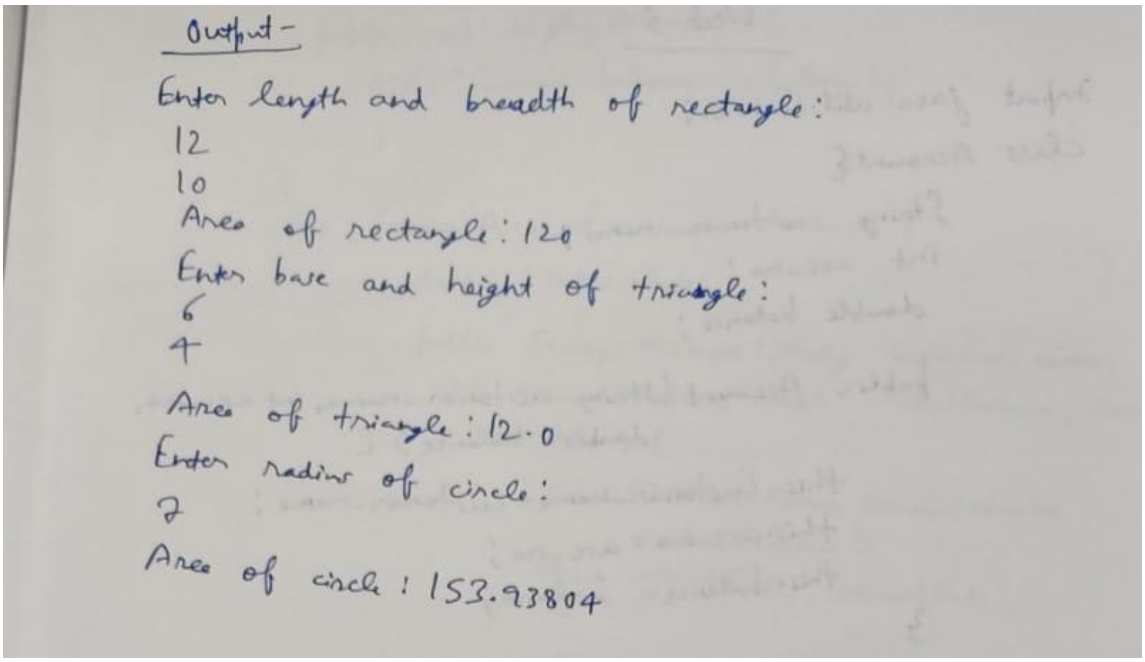
```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Enter the number of books:2
Enter the book name: Wings of Fire
Enter the author name: APJ Abdul Kalam
Enter the price of the book: 550
Enter the number of pages in the book: 1002
Enter the book name: Fire and Ice
Enter the author name: Robert Frost
Enter the price of the book: 450
Enter the number of pages in the book: 889

Book name: Wings of Fire
Author name: APJ Abdul Kalam
Price: 550
Number of pages: 1002

Book name: Fire and Ice
Author name: Robert Frost
Price: 450
Number of pages: 889

Vaibhav Dhar
1BM23CD068
Process finished with exit code 0
```

Program 4
Using Abstract Class Shape

Algorithm:

```
import java.util.Scanner;
abstract class Shape;
int di1;
int di2;
public Shape (){
    this.di1 = 0;
    this.di2 = 0;
}
public Shape ( int di1, int di2) {
    this.di1= di1 ;
    this.di2=di2;
}
public abstract void printArea()
}
Class Rectangle extends Shape {
    public Rectangle(int l, int w) {
        di1=l;
        di2=w;
    }
    public void printArea(){
        int area = di1*di2;
        SOP("Area of rectangle:"+area);
    }
}
class Triangle extends Shape {
    public Triangle (int b, int h){
        di1=b;
        di2=h;
    }
```

```
        public void printArea() {
            double area = 0.5 * di1 * di2;
            SOP("Area of triangle : " + area);
        }
    }
    class Circle extends Shape {
        public Circle(int r) {
            di1 = r;
            di2 = 0;
        }
        public void printArea() {
            double area = Math.PI * di1 * di1;
            SOP("Area of circle : " + area);
        }
    }
    public class Week4 {
        PSVM(String[] args)
        {
            Scanner scanner = new Scanner(System.in);
            SOP("Enter length and breadth for rectangle:");
            int l = scanner.nextInt();
            int b = scanner.nextInt();
            Shape rectangle = new Rectangle(l, b);
            rectangle.printArea();
            SOP("Enter base and height of triangle:");
            int b = scanner.nextInt();
            int h = scanner.nextInt();
            Shape triangle = new Triangle(b, h);
            triangle.printArea();
            SOP("Enter radius of circle:");
            int r = scanner.nextInt();
            Shape circle = new Circle(r);
            circle.printArea();
            scanner.close(); }
    }
```



Code:

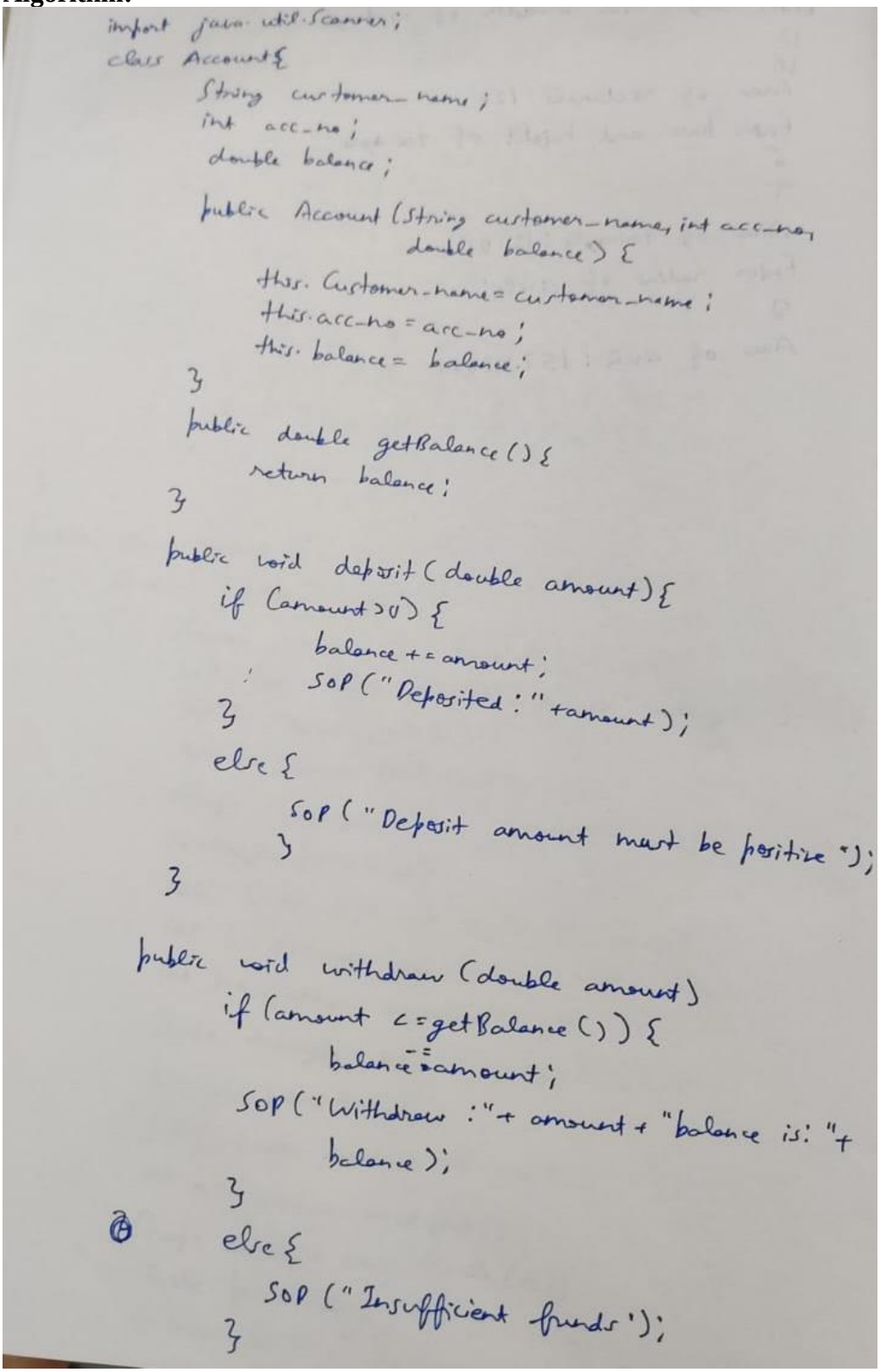
```
import java.util.Scanner ;

class Main{
   public static void main(String[] args){
      Rectangle ob2  = new Rectangle() ;
      Triangle ob1 = new Triangle() ;
      Circle ob3 = new Circle() ;
      ob2.printArea() ;
      ob1.printArea() ;
      ob3.printArea() ;
      System.out.println("Vaibhav Dhar" )  ;
      System.out.print("1BM23CD068") ;
   }
}

abstract class Shape{
   Scanner sc = new Scanner(System.in) ;
   int dimension1 , dimension2  ;
   abstract void printArea();
}

class Rectangle extends Shape{

   Rectangle(){
      System.out.println("Enter the dimensions of the rectangle(Length and Breadth): " ) ;
      dimension1 = sc.nextInt() ;
      dimension2 = sc.nextInt() ;
   }
```

```
    void printArea(){
        System.out.print("The area of the rectangle is = ") ;
        System.out.println(dimension1*dimension2) ;
    }
}

class Triangle extends Shape{
    Triangle(){
        System.out.println("Enter the dimensions of the triangle(base and height): " ) ;
        dimension1 = sc.nextInt() ;
        dimension2 = sc.nextInt() ;
    }

    void printArea(){
        System.out.print("The area of the Triangle is = ") ;
        System.out.println(0.5*dimension1*dimension2) ;
    }
}
class Circle extends Shape{
    Circle(){
        System.out.println("Enter the dimension of the circle(radius): ")  ;
        dimension1 = sc.nextInt() ;

    }

    void printArea(){
        System.out.print("The area of the Circle is = ") ;
        System.out.println(3.1415926535897*dimension1*dimension1) ;
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Enter the dimensions of the rectangle(Length and Breadth):
10 20
Enter the dimensions of the triangle(base and height):
15 20
Enter the dimension of the circle(radius):
14
The area of the rectangle is = 200
The area of the Triangle is = 150.0
The area of the Circle is = 615.7521601035811
Vaibhav Dhar
1BM23CD068
Process finished with exit code 0
```

Program 5
Bank Account Storage

Algorithm:

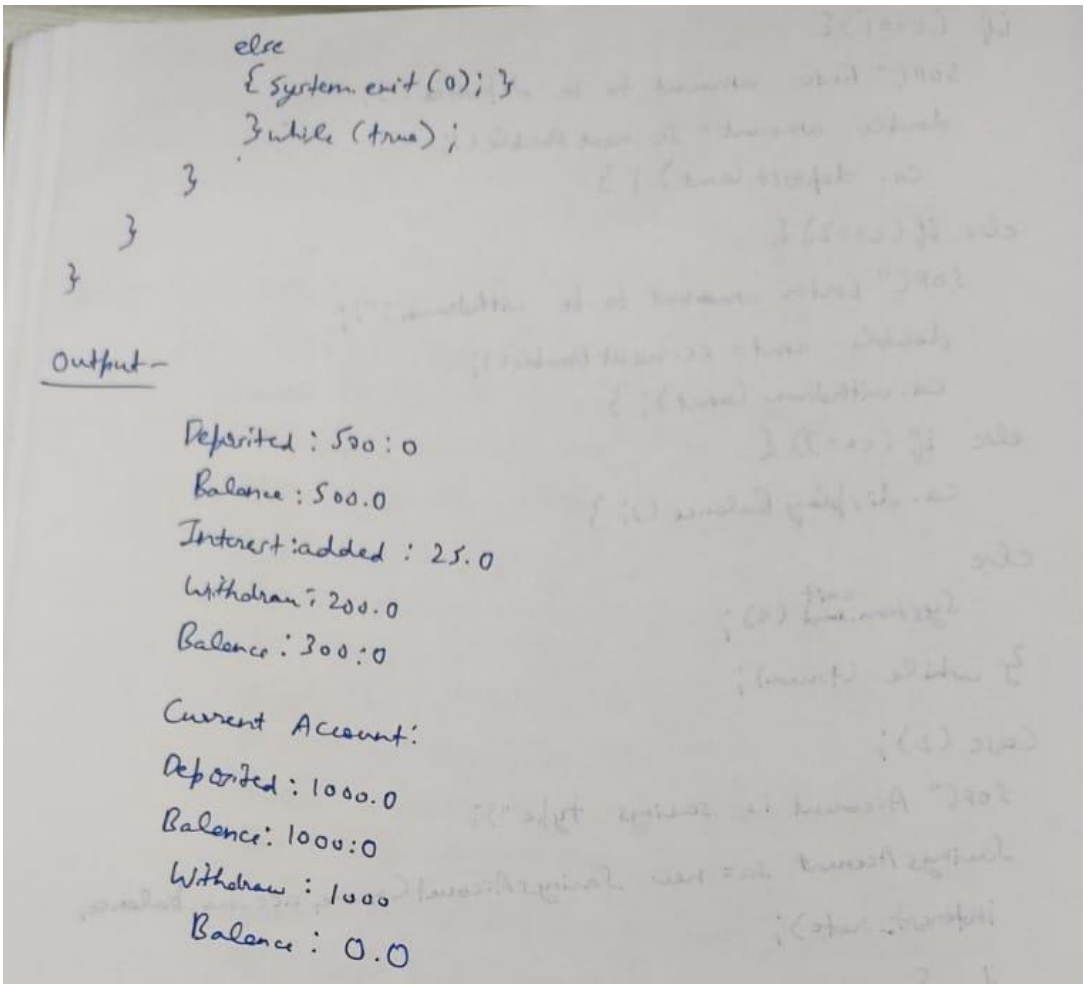


```
import java.util.Scanner;
class Account{
    String customer_name;
    int acc_no;
    double balance;

    public Account(String customer_name, int acc_no,
                   double balance) {
        this.Customer_name = customer_name;
        this.acc_no = acc_no;
        this.balance = balance;
    }
    public double getBalance(){
        return balance;
    }
    public void deposit(double amount){
        if (amount>0) {
            balance += amount;
            SOP("Deposited :"+amount);
        }
        else{
            SOP("Deposit amount must be positive");
        }
    }

public void withdraw(double amount)
    if (amount <= getBalance()) {
            balance -= amount;
        SOP("Withdraw :"+ amount + "balance is: "+
            balance);
        }
        else{
            SOP("Insufficient funds');
        }
```

```
    public void displayBalance() {
        SOP("Current balance: " + balance);
    }
}
class SavingsAccount extends Account {
    double interestRate;
    public SavingsAccount(String customer_name,
        int accountNumber, double initialBalance, double
        interestRate) {
            super(CustomerName, accountNumber,
                initialBalance);
            this.interestRate = interestRate;
        }
    public void computeAndDepositInterest() {
    double interest = getBalance()* interestRate /100);
    deposit(interest);
    }
}
class CurrentAccount extends Account{
    double minimumBalance, servicesCharge;
    public currentAccount(String CustomerName, int
    accountNumber, double initialBalance, double minBalance
    double serviceCharge) {
        super(CustomersName, accountNumber, initialBal
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
}
public void
    CheckMinimumBalance() {
        if (getBalance() < minimumBalance){
        SOP("Balance is below minimum");
        balance -= serviceCharge;
```

```
            SOP(" Deducted Service Charge:" + serviceCharge);
            SOP(" Balance after deduction:" + balance);
        }
    }
}
public class Bank {
    psvm (String[] args) {
        Scanner sc = new Scanner(System.in);
        SOP(" Enter customer name : ");
        String name = sc.nextLine;
        SOP(" Enter acc no : ");
        int accno = sc.~~next line ();~~ nextInt();
        SOP(" Enter initial balance :");
        double Balance = sc.nextDouble();
        SOP(" Enter min balance :");
        double minimum_balance = sc.nextDouble();
        SOP ("Enter interest rate :");
        double interest_rate = sc.nextDouble();
        SOP(" Enter service charge :");
        double service_charge = sc.nextDouble();
        SOP ("Enter choice : \n 1. Current acc \n 2.Savingsacc");
        int ch = sc.nextInt();
        SOP(" Customer name is:" + name + "\n Account
        number : " + acc_no + "\n Vaibhav - IBM23CD068 ");
        switch (ch) {
            case (1):
                SOP(" Account is current type ");
                Current account ca = new CurrentAccount (name, accNo,
                balance, minimum_balance, service charge);
                do {
                    SOP("Enter choice : \n 1. deposit \n 2 Withdraw \n
                    3. Deposit balance.");
                    int c = sc.next Int();
                    ca. checkMinimumBalance ();
```

```
        if (c==1){
            SOP(" Enter amount to be deposited:");
            double amount = sc.nextDouble();
             ca.deposit(amt) ; }
        else if(c==2){
            SOP(" Enter amount to be withdrawn:");
            double amt = sc.nextDouble();
            ca.withdraw(amt); }
        else if(c==3){
            ca.displayBalance(); }
        else
            System.exit(0);
        } while (true);
    Case(2);
    SOP(" Account is savings type:");
    SavingsAccount sa= new SavingsAccount(name, acc-no, balance,
    interest_rate);
    do {
      SOP(" Enter choice: \n 1.Deposit \n 2. Withdraw \n 3. Displa
      balance");
    int cl= sc.nextInt();
    if (cl==1) {
    SOP("Enter amount to be deposited:")
    double amt= sc.nextDouble();
    sa.deposit(amt); }
    else if (ch=2)
    { SOP(" Enter amount to withdraw:")
    double amt= sc.nextDouble();
    Sa.withdraw(amt); }
    else if (cl==3)
    { sa.computeAndDepositInterest();
      Sa.displayBalance(); }
```



Code:

```
import java.util.Scanner;

public class Bank {
   static Scanner sc = new Scanner(System.in);
   Account ob1;
   void createAccount() {
      String customer;
      int account;
      String type;
      int initBal;
      System.out.print("Enter the customer name: ");
      customer = sc.nextLine();
      System.out.print("Enter account Number: ");
      account = sc.nextInt();
      sc.nextLine();  // Consume the newline
      System.out.print("Enter Account type (Savings or Current): ");
      type = sc.nextLine();
      System.out.print("Enter the initial Balance: ");
      initBal = sc.nextInt();
```

```
        if (type.equals("Savings")) {
            ob1 = new Savings(customer, account, initBal);
        } else {
            ob1 = new Current(customer, account, initBal);
        }
    }
    public static void main(String[] args) {
        Bank bank = new Bank();
        bank.createAccount();
        while (true) {
            System.out.println("-------------------MENU-----------------");
            System.out.println("1. Deposit     2. Withdraw");
            System.out.println("3. Compute interest");
            System.out.println("4. Display account details");
            System.out.println("5. exit " ) ;
            int choice = sc.nextInt();
            switch (choice) {
                case 1:
                    bank.ob1.deposit();
                    break;
                case 2:
                    bank.ob1.withdraw();
                    break;
                case 3:
                    if (bank.ob1 instanceof Savings) {
                        ((Savings) bank.ob1).computeInterest();
                    } else {
                        System.out.println("Interest computation is only available for Savings accounts.");
                    }
                    break;
                case 4:
                    bank.ob1.display();
                    break;
                case 5:
                    break ;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
            if(choice == 5) break ;
        }
    }
}
class Account {
    String customerName;
    int accountNumber;
    int balance;
```

```
    Account(String customer, int accountNum, int bal) {
        customerName = customer;
        accountNumber = accountNum;
        balance = bal;
    }

    void deposit() {
        System.out.print("Enter the amount to deposit: ");
        int amt = Bank.sc.nextInt();
        balance += amt;
        System.out.println("Deposited: " + amt + ", New Balance: " + balance);
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < 0) {
            System.out.println("Insufficient Balance to withdraw the given amount.");
        } else {
            balance -= amt;
            System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
        }
    }
    void display() {
        System.out.println("The Balance in the account is " + balance);
    }
}
class Savings extends Account {
    double interestPercent;

    Savings(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
        System.out.print("Enter the interest percentage on the account: ");
        interestPercent = Bank.sc.nextDouble();
    }
    void computeInterest() {
        balance += balance * (interestPercent / 100);
        System.out.println("Amount after applying interest is: " + balance);
    }
}
class Current extends Account {
    int minBalance = 1000;

    Current(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
```

```
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < minBalance) {
            System.out.println("Insufficient Balance to maintain the minimum required.");
        } else {
            balance -= amt;
            System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
        }
    }
}
```

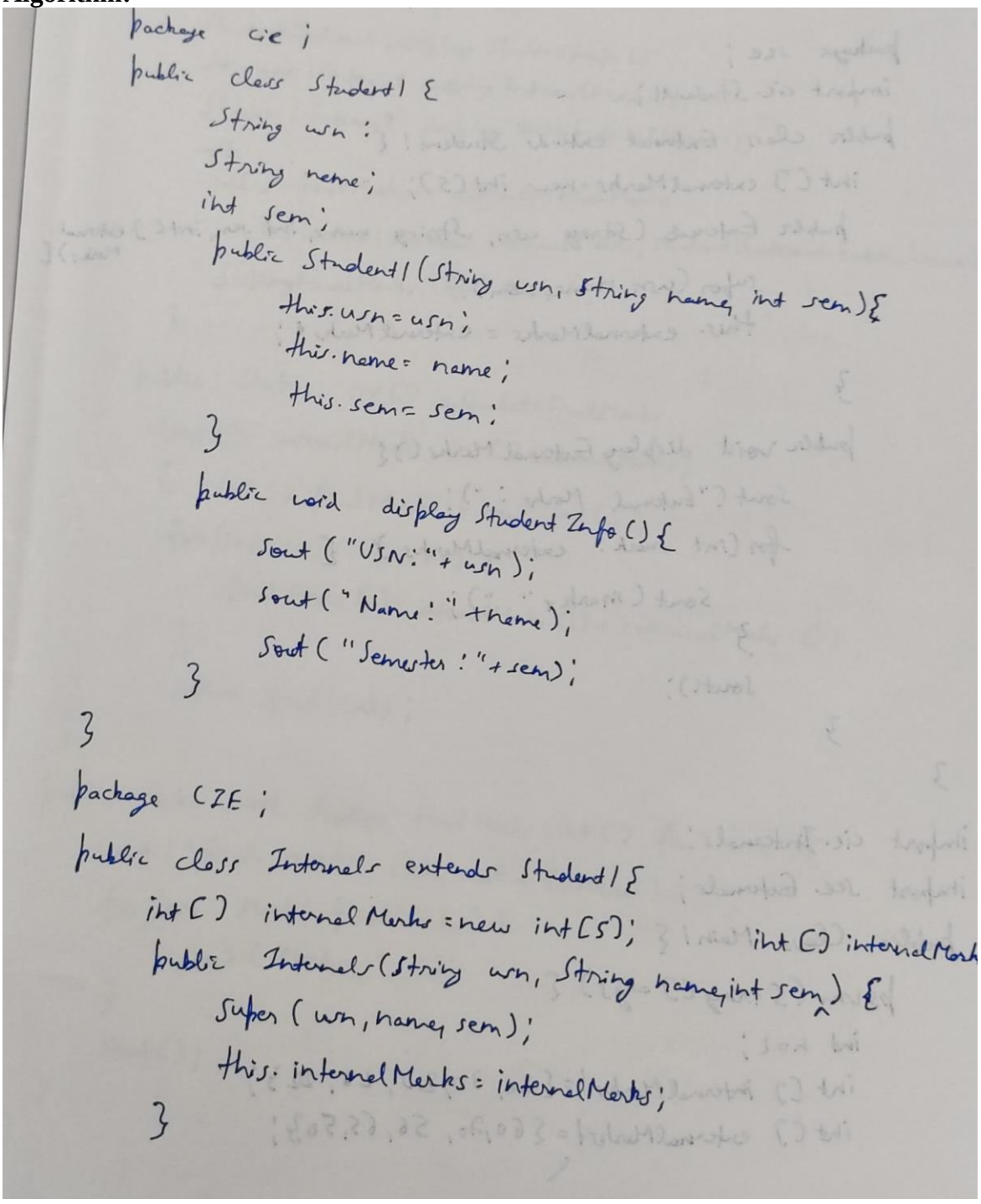
```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Enter the customer name: Vaibhav
Enter account Number: 12
Enter Account type (Savings or Current): Current
Enter the initial Balance: 1500
--------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
1
Enter the amount to deposit: 200
Deposited: 200, New Balance: 1700
--------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
2
Enter the amount to withdraw: 400
Amount of 400 withdrawn successfully. Current Balance is 1300
--------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
4
The Balance in the account is 1300
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Enter the customer name: Vaibhav
Enter account Number: 11
Enter Account type (Savings or Current): Savings
Enter the initial Balance: 1000
Enter the interest percentage on the account: 15
-------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
3
Amount after applying interest is: 1150
-------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
1
Enter the amount to deposit: 400
Deposited: 400, New Balance: 1550
-------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
4
The Balance in the account is 1550
-------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
2
Enter the amount to withdraw: 200
Amount of 200 withdrawn successfully. Current Balance is 1350
-------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
5

Process finished with exit code 0
```

Program 6

Creating Packages CIE and SEE

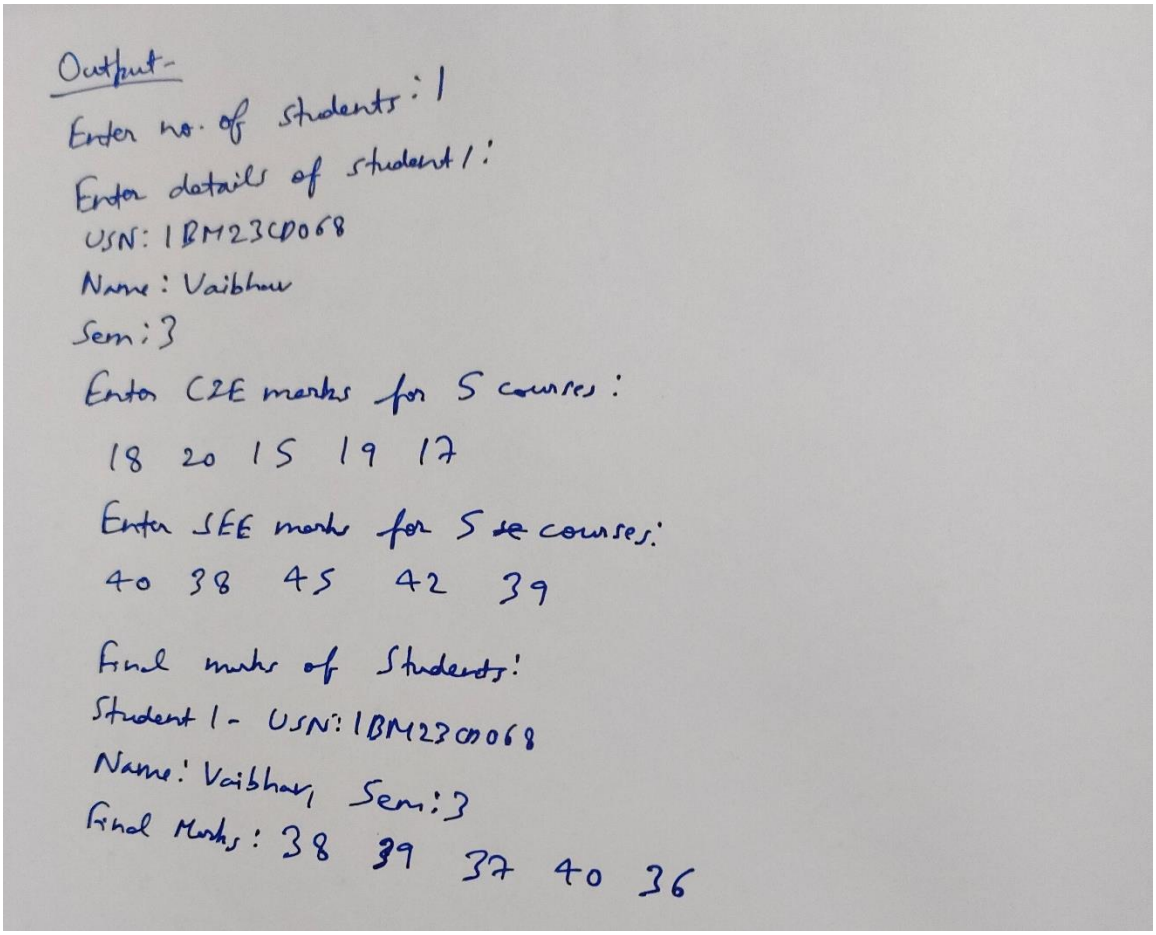
Algorithm:



```
package cie;
public class Student1 {
    String usn;
    String name;
    int sem;
    public Student1(String usn, String name, int sem){
        this.usn=usn;
        this.name= name;
        this.sem= sem;
    }
    public void displayStudentInfo(){
        Sout("USN:"+usn);
        Sout("Name:"+name);
        Sout("Semester:"+sem);
    }
}
package CIE;
public class Internals extends Student1{
    int[] internalMarks=new int[5];
    public Internals(String usn, String name,int sem) {
        super(usn,name,sem);
        this.internalMarks= internalMarks;
    }
```

```
    public void displayInternalMarks() {
        Sout("Internal marks:");
        for(int Mark: internalMarks) {
            Sout(Marks + " ");
        }
        Sout();
    }
}

package see;
import cie.Student1;
public class External extends Student1 {
    int[] externalMarks=new int[5];
    public External(String usn, String name, int sem, int[] external
                                                        Marks){
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }
    public void displayExternalMarks(){
        Sout("External Marks : ");
        for(int mark: externalMarks) {
            Sout(Mark+" ");
        }
        Sout();
    }
}

import cie.Internals;
import see.Externals;
public Class Main1{
    psvm(String[] args) {
        int n=2;
        int[] InternalMarks1={20, 30, 25, 28, 22};
        int[] externalMarks1={60,70, 56, 65,50};
```

```
External studentIExternal = new External
        ("1BM23CS068", "Vaibhav", 3, internalMarks);
int [] internal marks 2 = {18, 25, 20, 23, 28};
int[] external marks 2 = { 50, 65, 60, 58, 45};
Internals Student 2 Internal = new Internals
        ("1BM23CS001", "Aadit", 3, internalMarks);
Sout ("Student I Info :");
Student Internal . display Student Info ();
Student I Internal. display Internal Marks();
Student I External . display ~~Inter~~ External Marks();
~~Sto~~
int[] final Marks I = calculate Final Marks
    (Student I Internal. internal Marks, student IExternal. External marks);
display Final Marks (Ifinal Marks I);
}

public Static int[] calculate Final Marks
(int [] internal Marks, int[] External Marks)
{ int [] final Marks = new int[5];
for (int i=0; i< 5; i++) {
    final Marks [i] = internal Marks [i] + external Marks [i];
}
return final Marks;
}

public static void display Final Marks (int[] final Marks) {
    Sout ("Final marks (Internal + External) : ");
    for (int Mark : final Marks) {
        Sout ( Mark + " ");
    }
    Sout ();
}
}
```



Code:

```
package CIE;

import java.util.Scanner;

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

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

```
    // Method to display student details
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
package CIE;

import java.util.Scanner;

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

    // Method to input internal marks
    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter internal marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }
}
```

```
package SEE;

import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int[] marks = new int[5];      // SEE marks
    protected int[] finalMarks = new int[5]; // Final marks

    // Constructor to initialize the marks arrays
    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }

    // Method to input SEE marks
    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
```

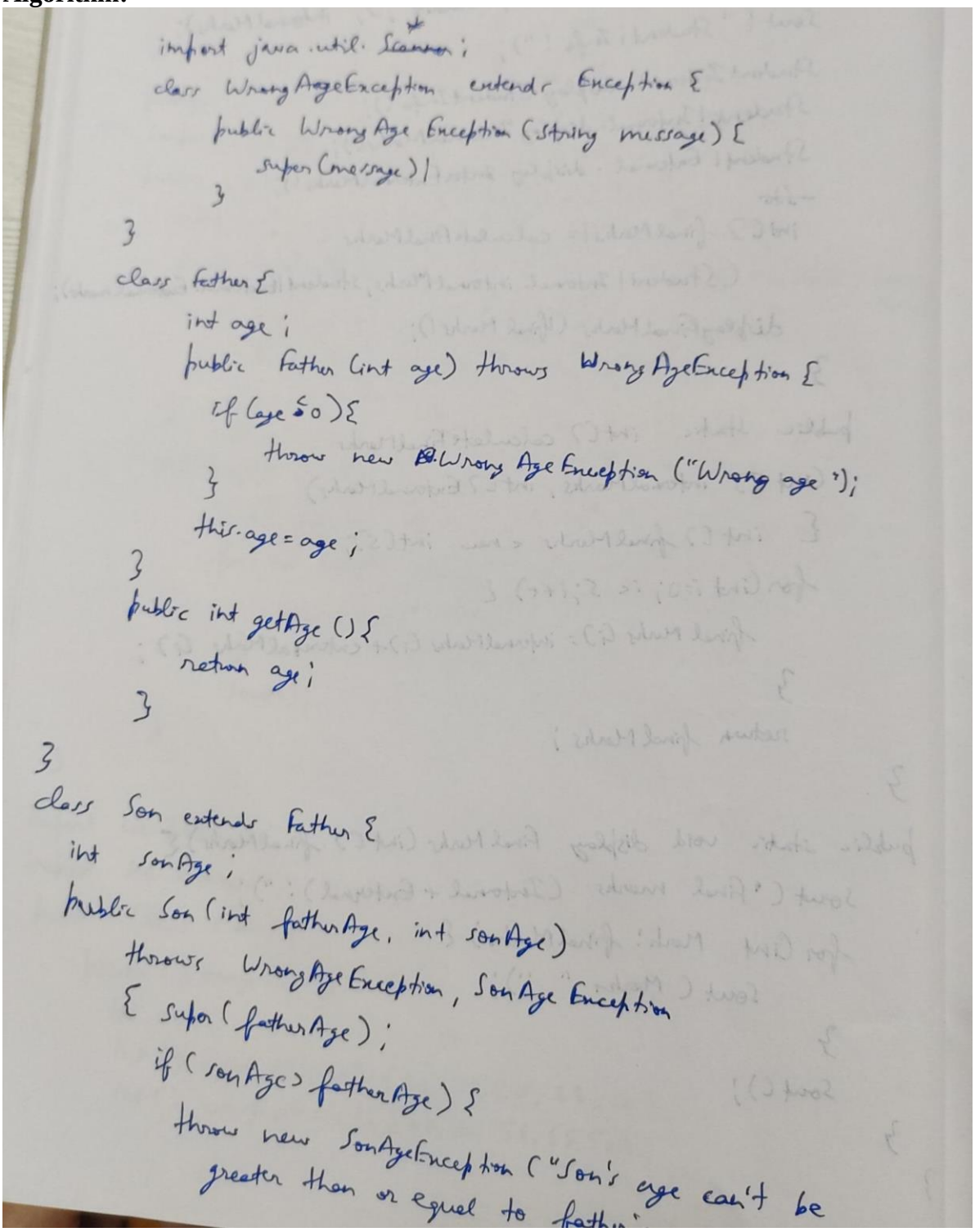
```
        System.out.println("Enter SEE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter SEE marks for subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }

    // Method to calculate final marks (internal + external)
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + this.marks[i]; // Final marks = internal + external
        }
    }

    // Method to display final marks
    public void displayFinalMarks() {
        displayStudentDetails();  // Display student details (inherited from Student)
        System.out.println("Final Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}



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

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

        // Input number of students
        System.out.print("Enter number of students: ");
        int n = s.nextInt();
        s.nextLine(); // Consume newline

        Externals[] students = new Externals[n];

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

```
            students[i].calculateFinalMarks();
        }

        // Display final marks for each student
        System.out.println("\nDisplaying final marks for all students:");
        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();
        }

        s.close();
    }
}
```

Program 7

Handling Exceptions in Inheritance Tree

Algorithm:

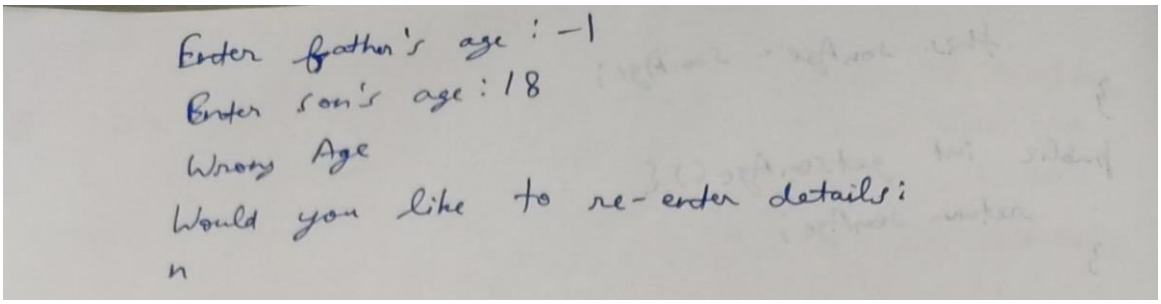


```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father{
    int age;
    public Father(int age) throws WrongAgeException {
      if(age<0){
        throw new WrongAgeException("Wrong age");
      }
      this.age=age;
    }
    public int getAge(){
      return age;
    }
}
class Son extends Father {
  int sonAge;
  public Son(int fatherAge, int sonAge)
    throws WrongAgeException, SonAgeException
    { super(fatherAge);
      if(sonAge>fatherAge){
        throw new SonAgeException("Son's age can't be
          greater than or equal to fathe
```

```
            }
            this.sonAge = sonAge;
        }
        public int getsonAge(){
            return sonAge;
        }
    }
    public class Main7 {
        psvm(String args[]) {
        while(true) {
            Scanner sc= new new Scanner(System.in);
            Sout("Enter father's age:");
            int F_age = sc.nextInt();
            Sout("Enter son's age:");
            int S_Age = sc.nextInt();
            try {
                Son son= new son(F_Age, S_Age);
                Sout("Accepted successfully");
            }
            Catch (WrongAgeException e) {
                Sout(e.getMessage());
            }
            Catch (SonAgeException e) {
                Sout(e.getMessage);
    }
    }
}
```

Output-

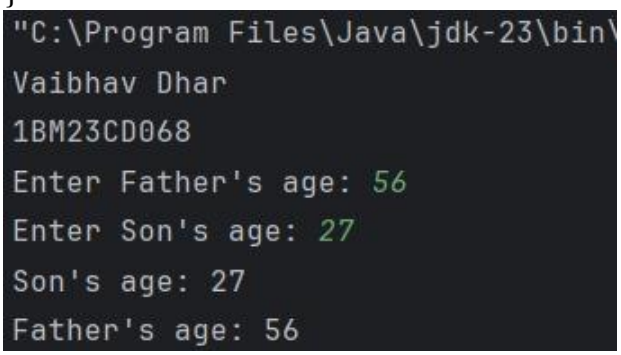
Enter father's age:45
Enter son's age:18
Accepted successfully
Would you like to re-enter details: y

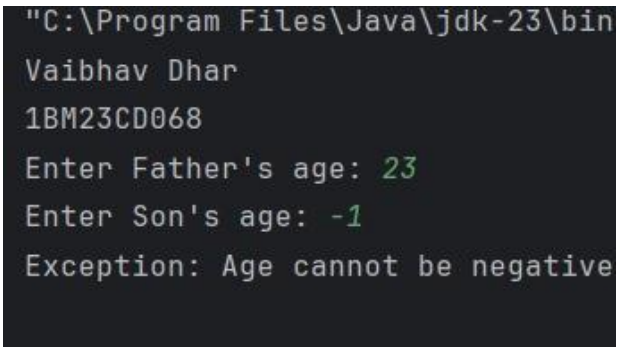


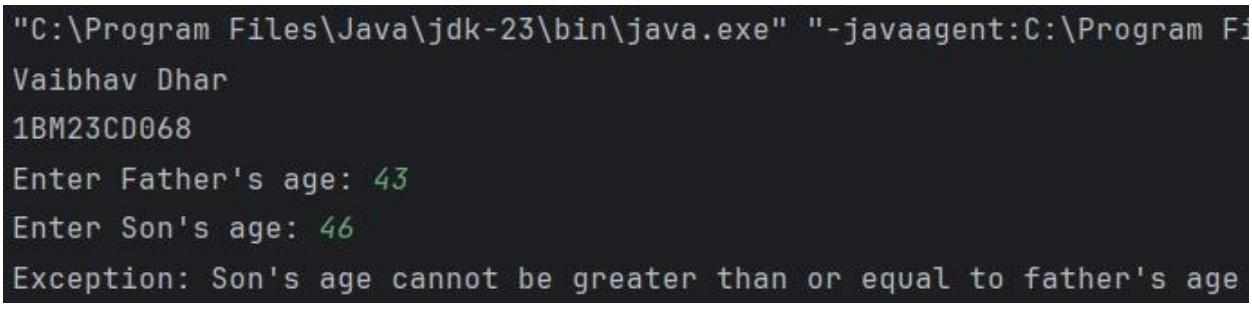
Code:

```
import java.util.Scanner;
class WrongAge extends Exception {
   public WrongAge() {
      super("Age Error");
   }
   public WrongAge(String message) {
      super(message);
   }
}
class InputScanner {
   Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
   int fatherAge;
   public Father() throws WrongAge {
      System.out.print("Enter Father's age: ");
      fatherAge = s.nextInt();
      if (fatherAge < 0) {
         throw new WrongAge("Age cannot be negative");
      }
   }
   public void display() {
      System.out.println("Father's age: " + fatherAge);
   }
}
class Son extends Father {
   int sonAge;
   public Son() throws WrongAge {
      super();
      System.out.print("Enter Son's age: ");
      sonAge = s.nextInt();
      if (sonAge >= fatherAge) {
         throw new WrongAge("Son's age cannot be greater than or equal to father's age");
      } else if (sonAge < 0) {
         throw new WrongAge("Age cannot be negative");
      }
   }
```

```
    public void display() {
        System.out.println("Son's age: " + sonAge);
        super.display();  // This calls the Father's display method
    }
}
public class Exception_Handling{
    public static void main(String[] args) {
        try {
            System.out.println("Vaibhav Dhar\n1BM23CD068");
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

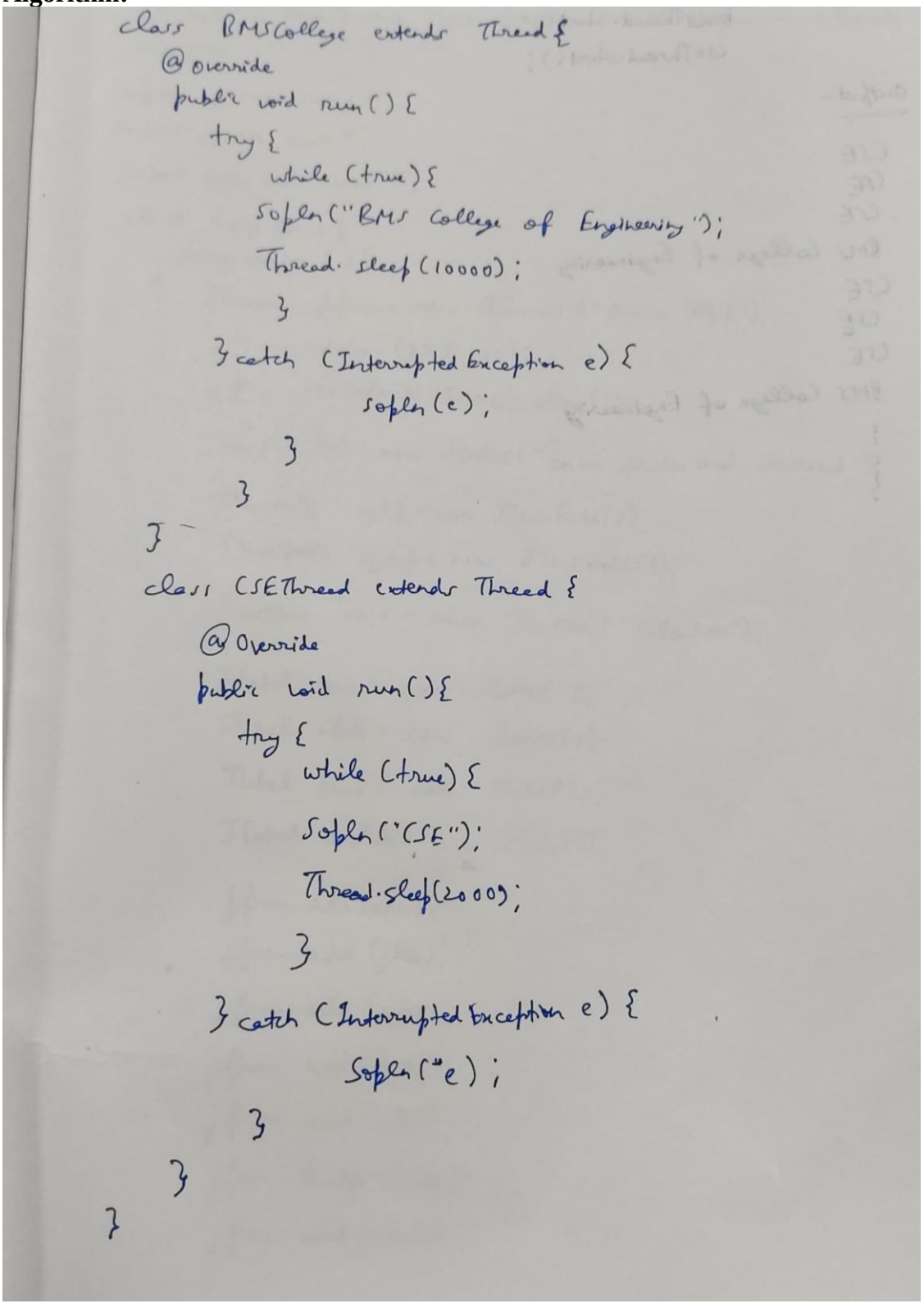




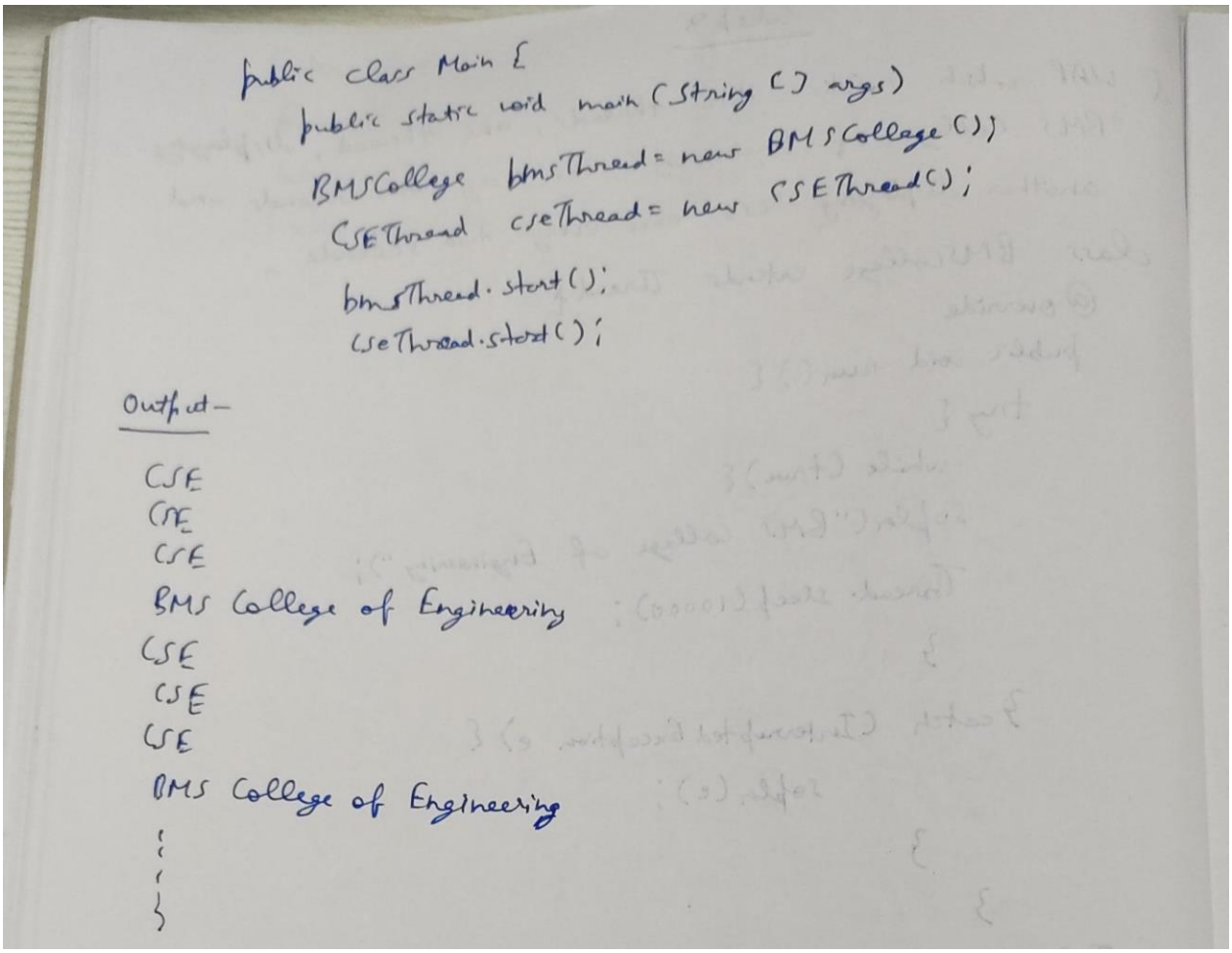


Program 8
Threads Creation

Algorithm:



```
class BMSCollege extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
            Sopln("BMS College of Engineering");
            Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
                Sopln(e);
            }
        }
}
class CSEThread extends Thread {
    @Override
    public void run() {
        try {
            while (true) {
            Sopln("CSE");
            Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
                Sopln("e");
            }
        }
}
```



Code:

```
public class Threads {
   static class BMSDisplayThread extends Thread {
      public void run() {
         try {
            while (true) {
               System.out.println("BMS College of Engineering");
               Thread.sleep(10000);  // Sleep for 10 seconds
            }
         } catch (InterruptedException e) {
            System.out.println(e);
         }
      }
   }

   static class CSEDisplayThread extends Thread {
      public void run() {
         try {
            while (true) {
               System.out.println("CSE");
```

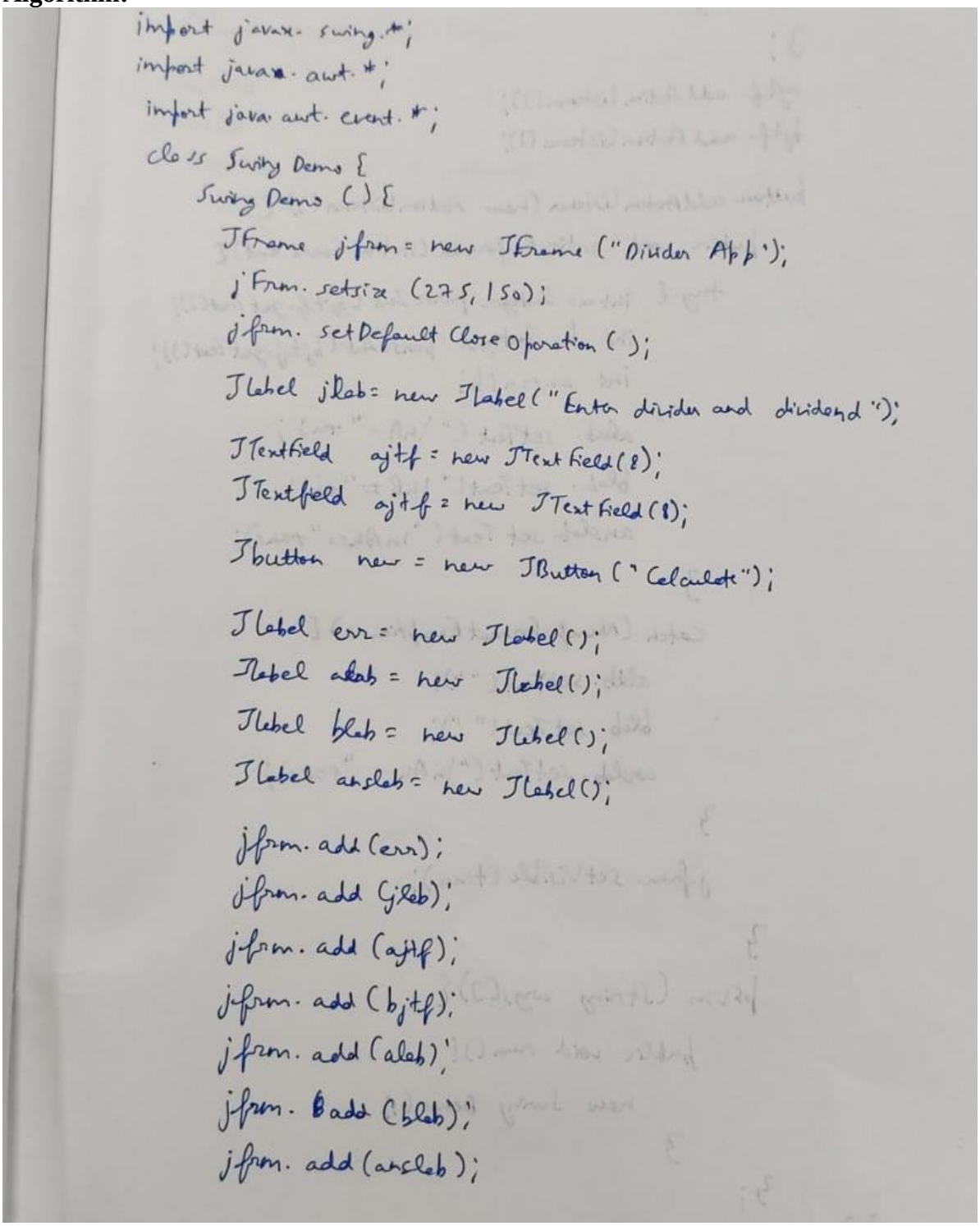
```
            Thread.sleep(2000);
          }
        } catch (InterruptedException e) {
          System.out.println(e);
        }
      }
    }

    public static void main(String[] args) {
      BMSDisplayThread bmsThread = new BMSDisplayThread();
      CSEDisplayThread cseThread = new CSEDisplayThread();
      System.out.println("Vaibhav Dhar");
      System.out.println("1BM23CD068");
      bmsThread.start();
      cseThread.start();
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
Vaibhav Dhar
1BM23CD068
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE

Process finished with exit code 130
```

Program 9

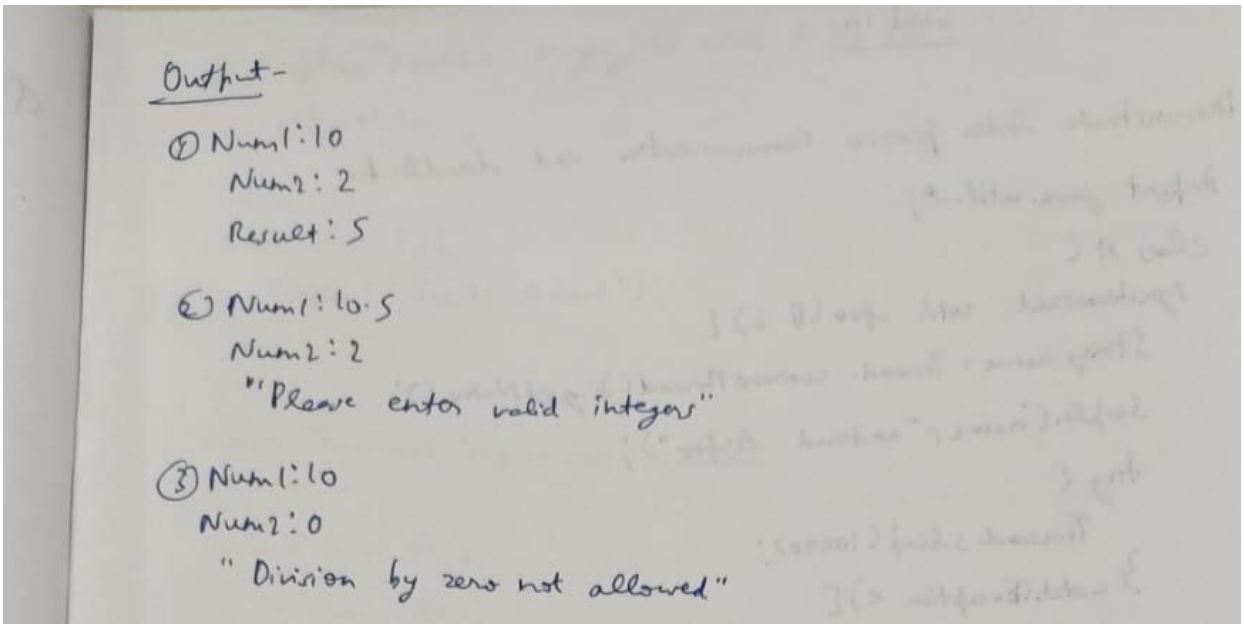
User Interface Creation

Algorithm:



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo () {
        JFrame jfrm = new JFrame ("Divider App");
        jFrm.setSize (275, 150);
        jfrm.setDefaultCloseOperation ( );
        JLabel jlab= new JLabel("Enter divider and dividend ");
        JTextField ajtf = new JTextField(8);
        JTextfield ajtf = new JTextField(8);
        Jbutton new = 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 (alab);
        jfrm.add (blab);
        jfrm.add (anslab);
```

```
ajtf.addActionListener(1);
bjtf.addActionListener(1);

button.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent evt){
        Sopln("Action event from a text field");
    }
};
ajtf.addActionListener(1);
bjtf.addActionListener(1);

button.addActionListener(new ActionListener()){
    public void actionPerformed(ActionEvent evt){
        try { int a= Integer.parseInt(ajtf.getText());
              int b= Integer.parseInt(bjtf.getText());
              int ans=a/b;
              alab.setText("\nA = "+a);
              blab.setText("\nB = "+b);
              anslab.setText("\nAns="+ans);
        }
        catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("\nAns = "+ans);
        }
        jfrm.setVisible(true);
    }
    psvm(String args[]){
        public void run(){
            new SwingDemo();
        }
};
```



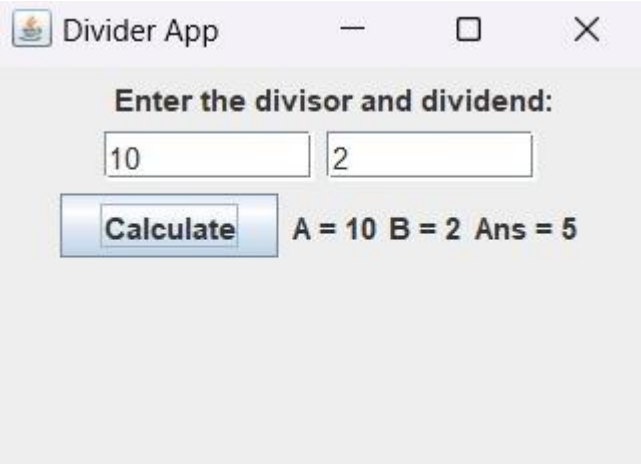
Code:

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

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 divisor and dividend:");

      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      JButton button = new JButton("Calculate");

      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
      JLabel anslab = new JLabel();

      jfrm.add(err);
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
```

```
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());

                    int ans = a / b;

                    alab.setText("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                    err.setText("");
                } catch (NumberFormatException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmeticException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("B should be NON zero!");
                }
            }
        });

        jfrm.setVisible(true);
    }

    public static void main(String args[]) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
} import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
```

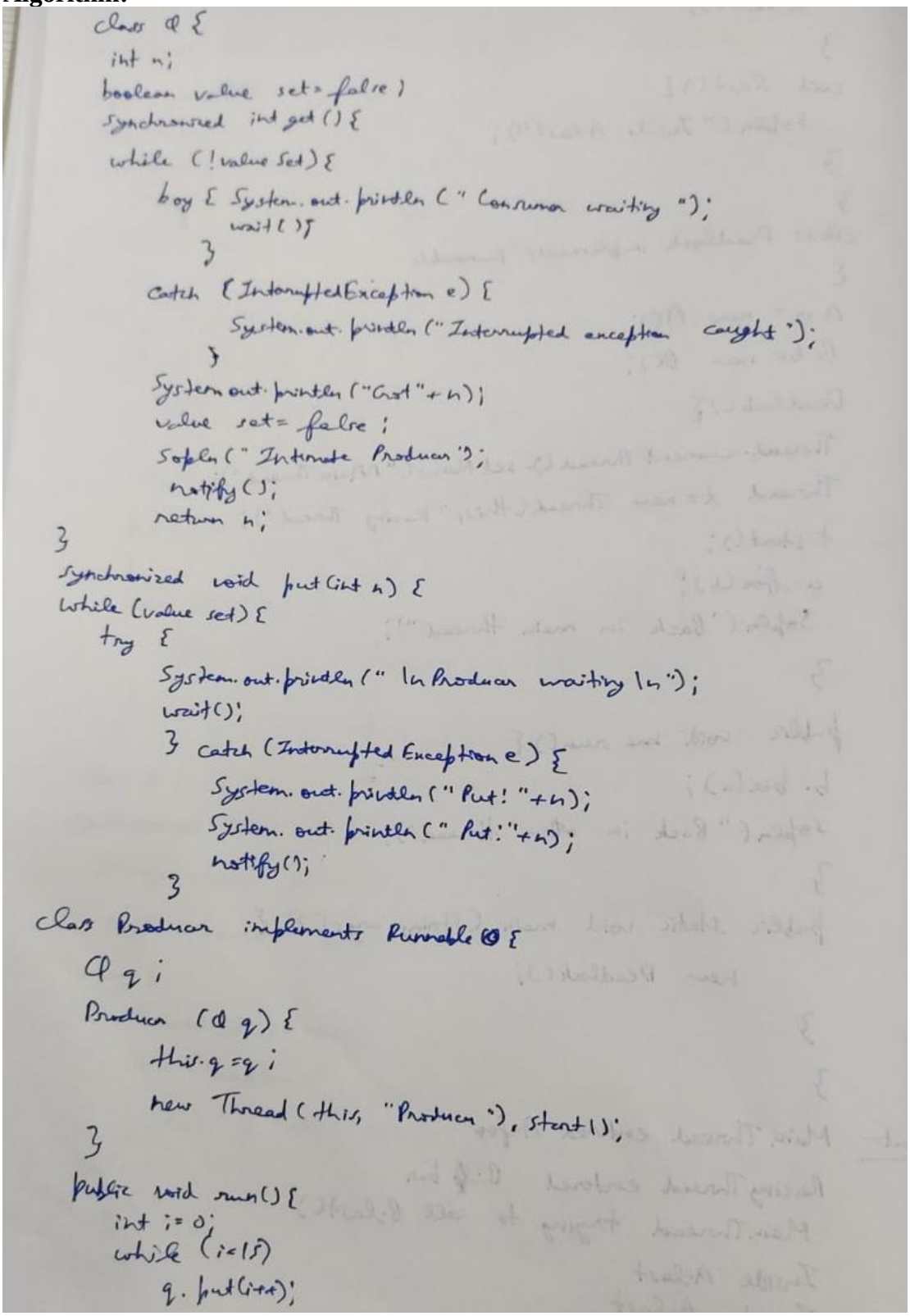
```
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 200);
jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

button.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      try {
         int a = Integer.parseInt(ajtf.getText());
         int b = Integer.parseInt(bjtf.getText());

         int ans = a / b;

         alab.setText("A = " + a);
         blab.setText("B = " + b);
         anslab.setText("Ans = " + ans);
         err.setText("");
      } catch (NumberFormatException e) {
         alab.setText("");
         blab.setText("");
         anslab.setText("");
         err.setText("Enter Only Integers!");
      } catch (ArithmeticException e) {
         alab.setText("");
         blab.setText("");
```

```
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });
    jfrm.setVisible(true);
  }

  public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new SwingDemo();
      }
    });
  }
}
```

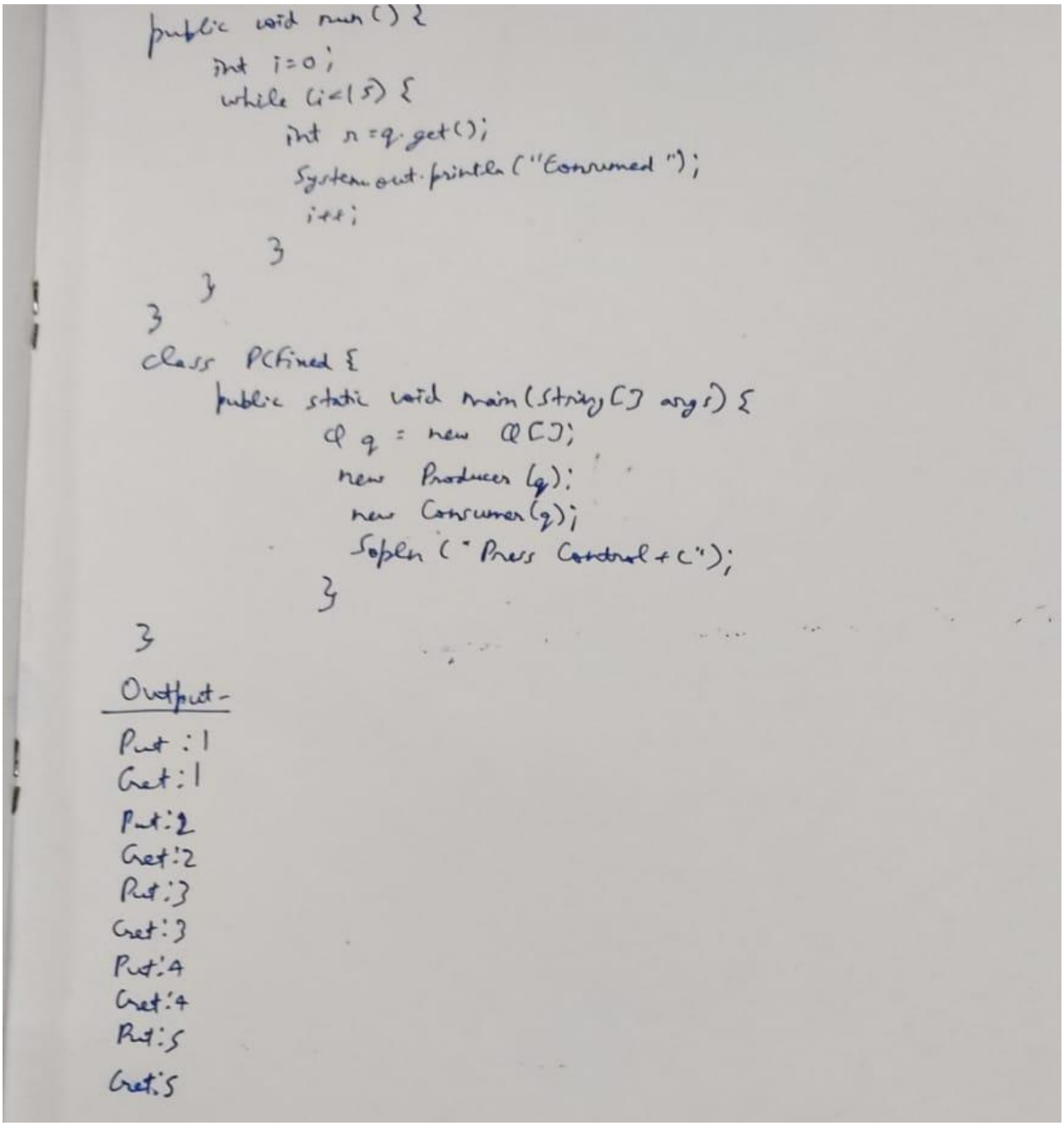


Program 10 a)
Demonstrating IPC

Algorithm:



```
class Q {
 int n;
 boolean value set = false )
 Synchronized int get () {
 while (!value set) {
     try { System.out.println (" Consumer waiting ");
       wait ()
       }
     catch (InterruptedException e) {
         System.out.println (" Interrupted exception caught ");
       }
     System out println ("Got" + n);
     value set = false ;
     System.out.println (" Intimate Producer ");
      notify ();
     return n;
}
Synchronized void put (int n) {
while (value set) {
   try {
       System.out.println (" In Producer waiting \n ");
       wait();
       } catch (InterruptedException e) {
          System.out.println (" Put! "+n);
          System.out.println (" Put:"+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++);
```



Code:

```
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++;
        }
    }
}
class IPC {
    public static void main(String args[]) {
        Q q = new Q();
        System.out.println("Vaibhav Dhar\n1BM23CD068");
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

```
"C:\Program Files\Java\j
Vaibhav Dhar
1BM23CD068
Put: 0

Intimate Consumer


Producer waiting

Press Control-C to stop.
Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer
```

```
Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer


Producer waiting

Got: 4
```

```
Intimate Producer

consumed:4
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

consumed:6
Put: 7
```

```
Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer


Producer waiting

Got: 8

Intimate Producer

consumed:8
Put: 9

Intimate Consumer


Producer waiting
```

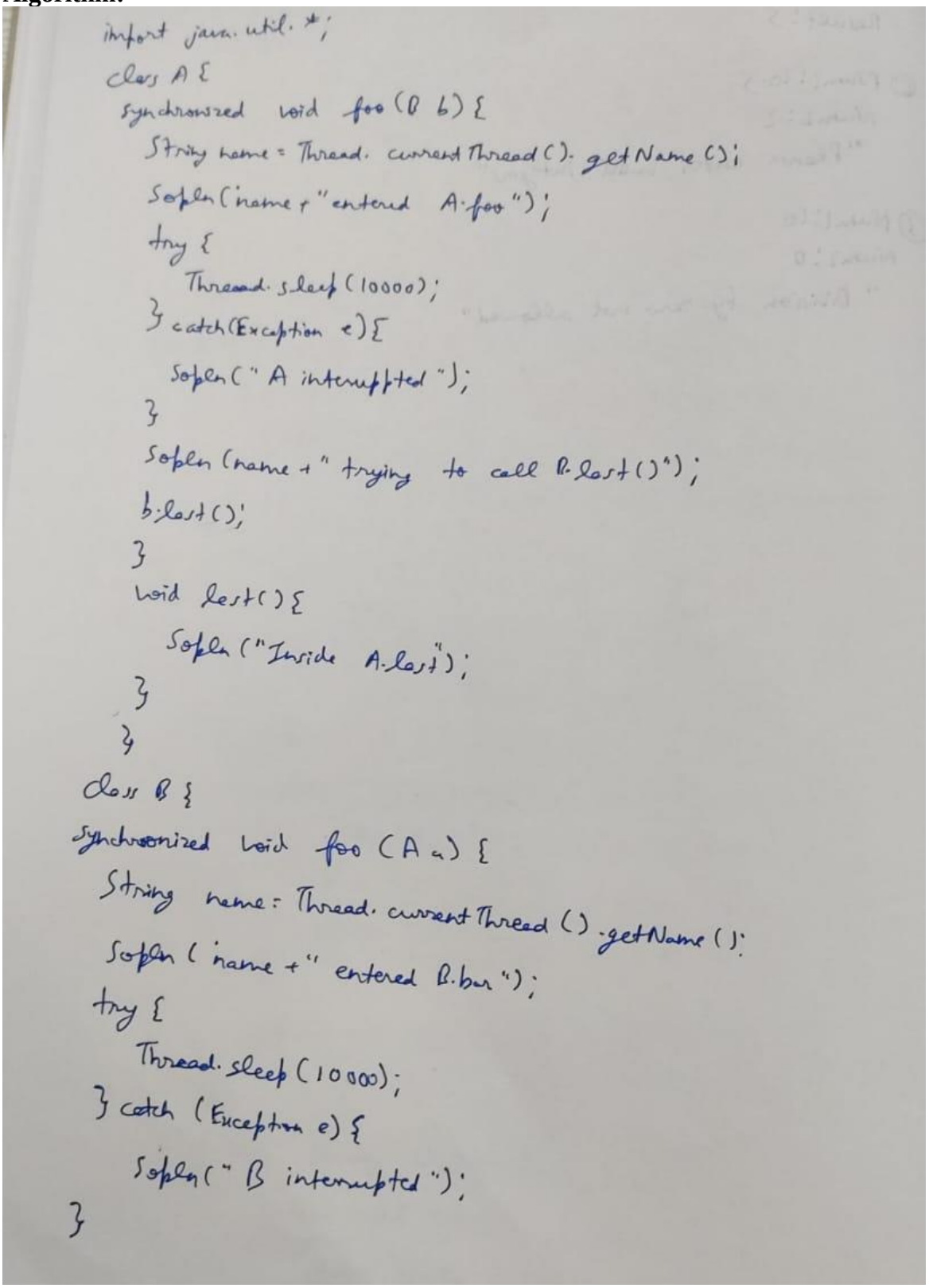
```
Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer


Producer waiting

Got: 10

Intimate Producer

consumed:10
Put: 11

Intimate Consumer


Producer waiting

Got: 11

Intimate Producer
```

```
consumed:11
Put: 12

Intimate Consumer


Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer
```

```
Got: 14

Intimate Producer

consumed:14

Process finished with exit code 0
```

Program 10 b)
Demonstrating Deadlock

Algorithm:



```
import java.util.*;
class A {
 synchronized void foo(B b) {
   String name = Thread.currentThread().getName();
   System.out.println(name + " entered A.foo");
   try {
     Thread.sleep(10000);
   } 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 foo(A a) {
  String name = Thread.currentThread().getName();
  System.out.println(name + " entered B.bar");
  try {
    Thread.sleep(10000);
  } catch (Exception e) {
    System.out.println(" B interrupted ");
}
```

```
        Sopln("name + "trying to call A.last()");
        a.last();
    }
    void last(){
        Sopln("Inside A.last");
    }
}
class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock(){
Thread.currentThread().setName("Main Thread");
Thread t = new Thread(this, "Racing Thread");
t.start();
a.foo(b);
Sopln("Back in main thread");
}
public void run(){
b.bar(a);
Sopln("Back in other thread");
}
public static void main(String args[]) {
    new Deadlock();
}
}
```

Output-

```
Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Inside A.last
Back in other thread
```

Code:

```
import java.util.*;
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();
   }

   synchronized 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();
   }

   synchronized 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();
   }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files
MainThread entered == A.foo
RacingThread entered B.bar
MainThread trying to call B.last
RacingThread trying to call A.last
```