VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

StudentName (1BM23CS074)

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 **Chethan K S (1BM23CS074),** 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.

Geetha N Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

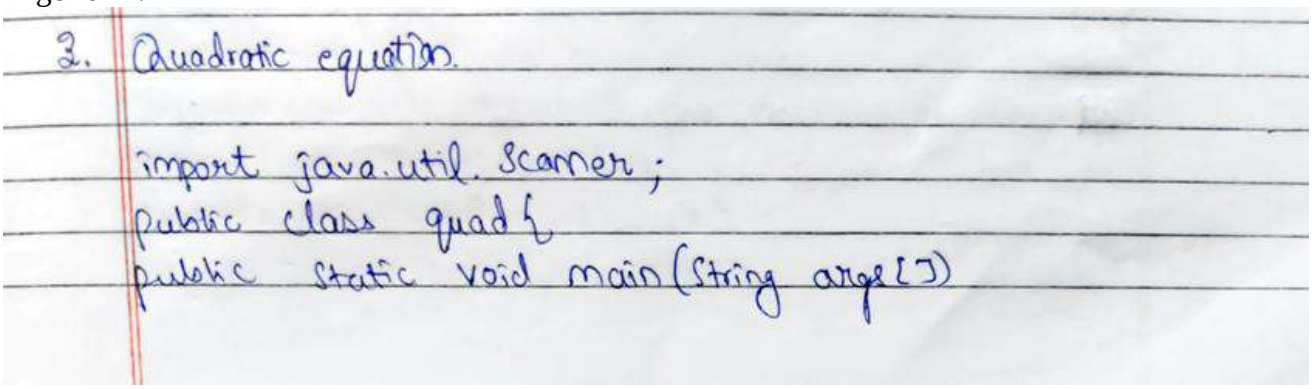
Github Link:
https://github.com/Chethan-K-S/OOJ_LAB

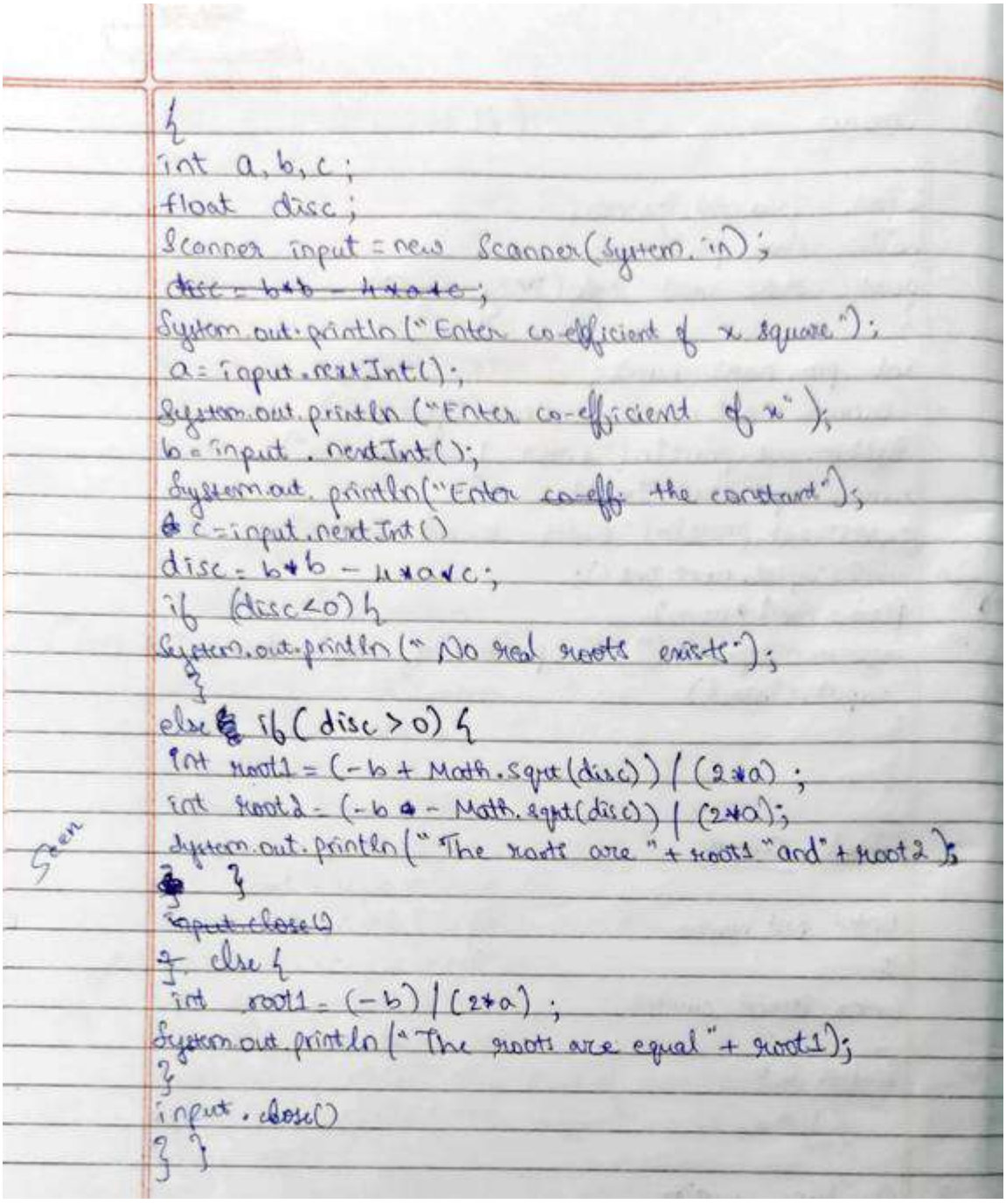
Program 1

Implement Quadratic Equation

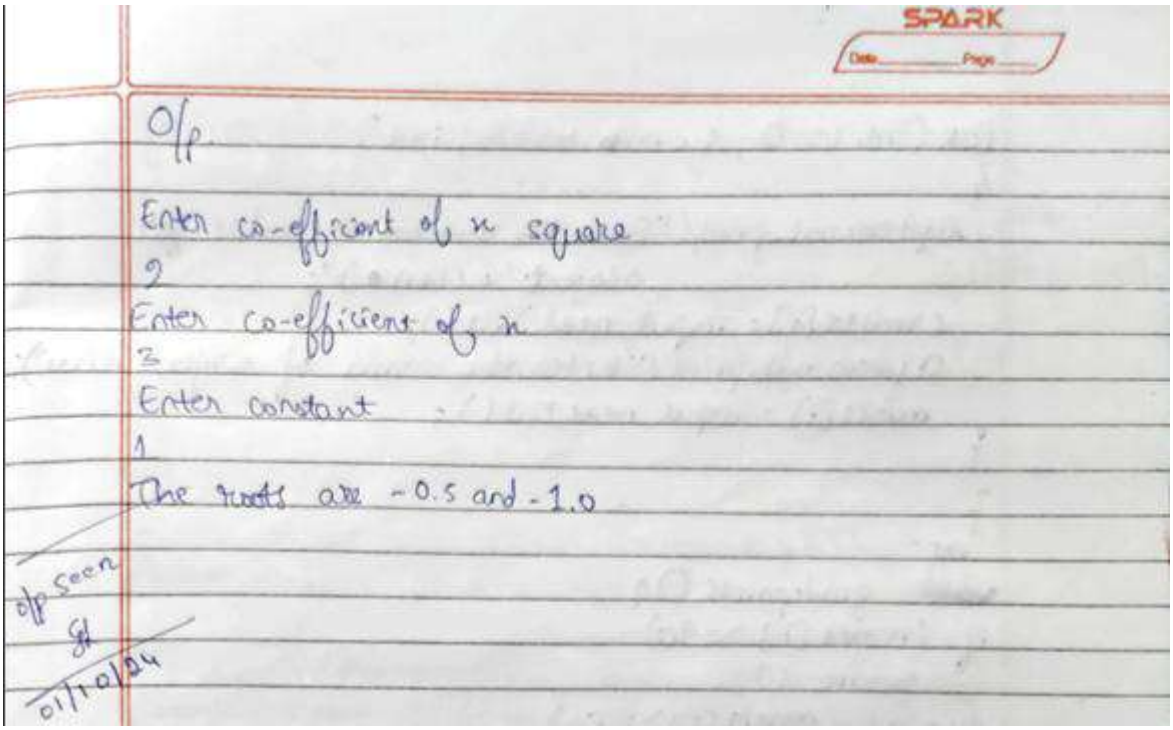
Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions

Algorithm:





```
{
int a, b, c;
float disc;
Scanner input = new Scanner(System.in);
disc = b*b - 4*a*c;
System.out.println("Enter co-efficient of x square");
a = input.nextInt();
System.out.println("Enter co-efficient of x");
b = input.nextInt();
System.out.println("Enter the constant");
c = input.nextInt()
disc = b*b - 4*a*c;
if (disc<0) {
System.out.println("No real roots exists");
}
else if(disc>0) {
int root1 = (-b + Math.sqrt(disc)) / (2*a);
int root2 = (-b - Math.sqrt(disc)) / (2*a);
System.out.println("The roots are " + root1 "and" + root2 );
}
input.close()
} else {
int root1 = (-b) / (2*a);
System.out.println("The roots are equal" + root1);
}
input.close()
} }
```



Code:

```
import java.util.Scanner;

public class quad {
   public static void main(String[] args) {
      int a, b, c;
      float disc;
      Scanner input = new Scanner(System.in);

      System.out.println("Enter co-efficient of x square");
      a = input.nextInt();
      System.out.println("Enter co-efficient of x ");
      b = input.nextInt();
      System.out.println("Enter the constant");
      c = input.nextInt();

      disc = b * b - 4 * a * c;

      if (disc < 0) {
         System.out.println("No real root exists");
      } else if (disc > 0) {
         double root1 = (-b + Math.sqrt(disc)) / (2 * a);
         double root2 = (-b - Math.sqrt(disc)) / (2 * a);
         System.out.println("The roots are " + root1 + " and " + root2);
      } else {
         double root1 = (-b) / (2 * a);
         System.out.println("The roots are equal " + root1);
```

```
        }

        System.out.println("CHETHAN K S\n1BM23CSO74");
        input.close();
    }
}
```

Output

```
Enter co-efficient of x square
2
Enter co-efficient of x
3
Enter the constant
1
The roots are-0.5and-1.0
CHETHAN K S
1BM23CSO74
```

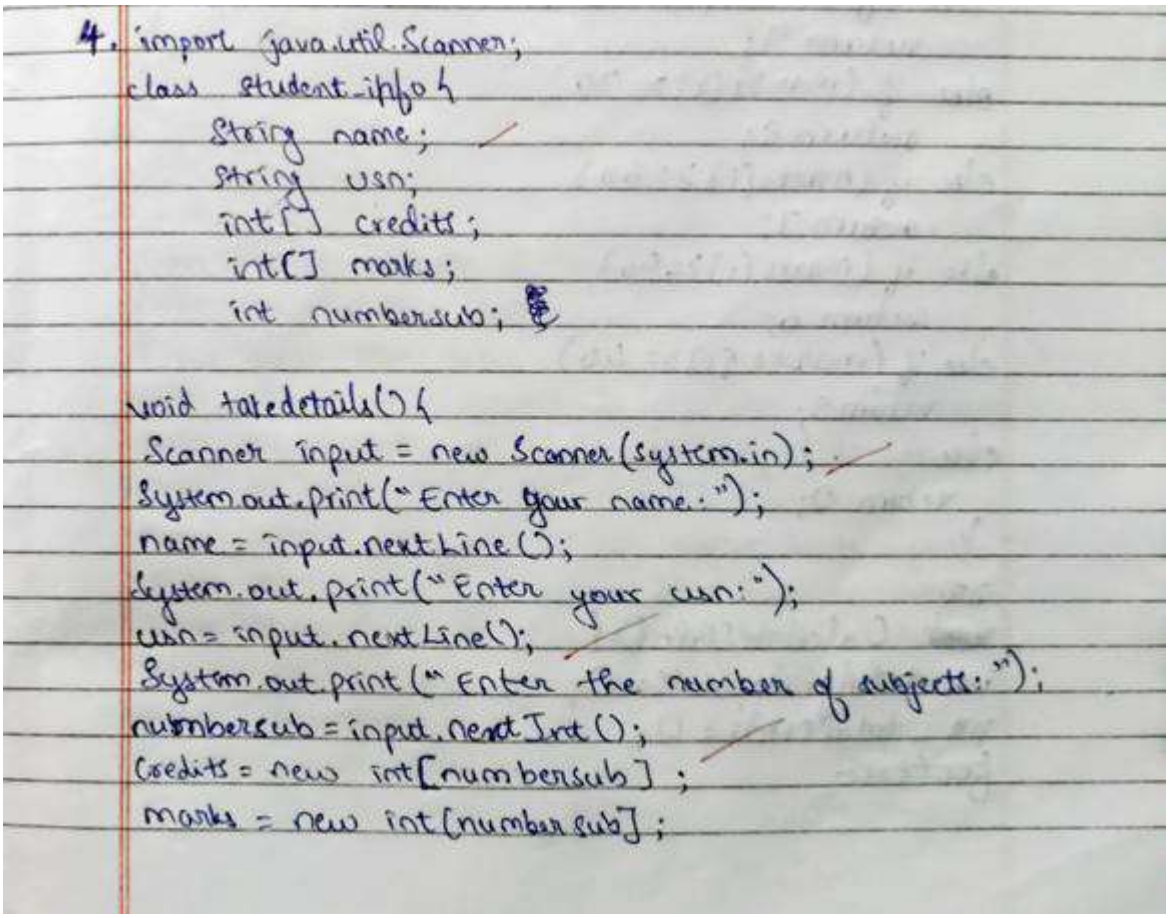
```
Enter co-efficient of x square
4
Enter co-efficient of x
4
Enter the constant
2
No real root exists
CHETHAN K S
1BM23CSO74
```

```
Enter co-efficient of x square
2
Enter co-efficient of x
4
Enter the constant
2
 The roots are equal -1.0
```

Program 2

Calculating SGPA

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.



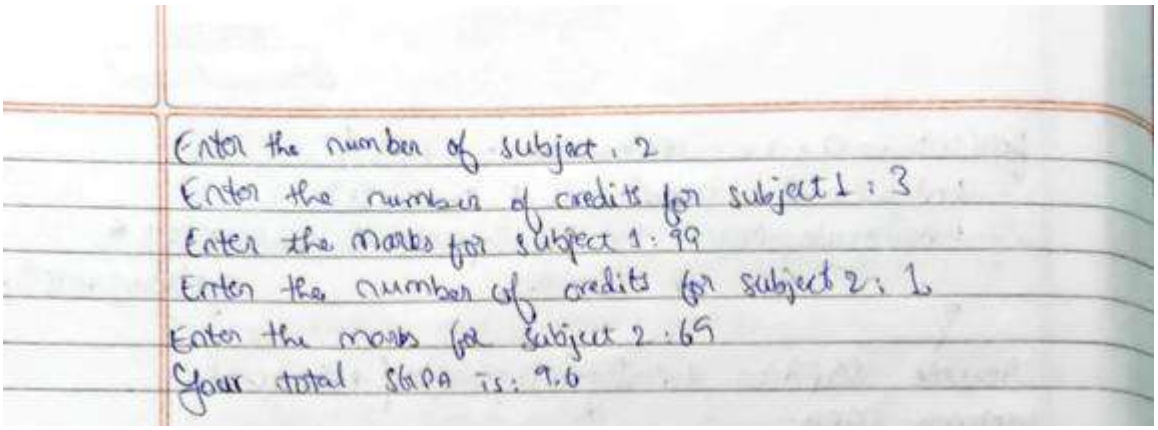
```
4. import java.util.Scanner;
class Student_info {
    String name;
    String usn;
    int[] credits;
    int[] marks;
    int numbersub;

void takedetails(){
Scanner input = new Scanner(System.in);
System.out.print("Enter your name:");
name = input.nextLine();
System.out.print("Enter your usn:");
usn = input.nextLine();
System.out.print("Enter the number of subjects:");
numbersub = input.nextInt();
credits = new int[numbersub];
marks = new int[numbersub];
```

```
for (int i = 0; i < numberSub; i++)
{
  System.out.print("Enter the number of credits for
                    Subject" + (i+1) + ");
   credits[i] = input.nextInt();
   System.out.print("Enter the marks of subject" + (i+1));
    marks[i] = input.nextInt();
}
}
int
void gradepoints(){
if (marks[i] >= 90)
    return 10;
else if (marks[i] >= 80)
    return 9;
else if (marks[i] >= 70)
    return 8;
else if (marks[i] >= 60)
    return 7;
else if (marks[i] >= 50)
    return 6;
else if (marks[i] >= 40)
    return 5;
else
  return 0;
}
int
void CalculateSGPA(){
int totalgradepoints = 0;
int totalcreds = 0;
```

```
for(int i=0; i<numberSub; i++){
    totalCredS = totalCredS + Credits[i];
    totalgradepoints = totalgradepoints + credits[i] *
                                    gradepoints[i];
}
double SGPA = totalgradepoints / totalCreds;
return SGPA;
}

public static void main(String args[])
public class Student_info
{
    System.out.println("Your total SGPA is"
                + student1.Calculate SGPA());
}
}
```

Output

```
Enter no. of students 2
Enter your name: Chethan
Enter your USN: 074
Enter number of subjects: 3
Enter the number of credits for subject 1: 2
Enter the marks for subject 1: 92
Enter the number of credits for subject 2: 4
Enter the marks for subject 2: 88
Enter the number of Credits for subject 3: 3
Enter the marks for subject 3: 80
Your total SGPA is: 9.0
enter your name: raja
Enter your USN 070
```



Code:

```
import java.util.Scanner;

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

    void takeDetails() {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter your name: ");
        name = input.nextLine();
        System.out.print("Enter your USN: ");
        usn = input.nextLine();
        System.out.print("Enter number of subjects: ");
        numberOfSubjects = input.nextInt();

        credits = new int[numberOfSubjects];
        marks = new int[numberOfSubjects];

        for (int i = 0; i < numberOfSubjects; i++) {
            System.out.print("Enter the number of credits for subject " + (i + 1) + ": ");
            credits[i] = input.nextInt();
            System.out.print("Enter the marks for subject " + (i + 1) + ": ");
            marks[i] = input.nextInt();
        }
    }

    int gradePoints(int i) {
        if (marks[i] >= 90) return 10;
        else if (marks[i] >= 80) return 9;
```

```
        else if (marks[i] >= 70) return 8;
        else if (marks[i] >= 60) return 7;
        else if (marks[i] >= 50) return 6;
        else if (marks[i] >= 40) return 5;
        else return 0;
    }

    double calculateSGPA() {
        int totalGradePoints = 0;
        int totalCredits = 0;

        for (int i = 0; i < numberOfSubjects; i++) {
            totalCredits += credits[i];
            totalGradePoints += credits[i] * gradePoints(i);
        }

      return totalGradePoints/totalCredits;
    }
}

public class student_info {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
            System.out.print("Enter no of students ");
            int no=input.nextInt();
            for(int i = 0; i < no; i++)
            {

                    Student student = new Student();
                    student.takeDetails();
                    System.out.println("Your total SGPA is: " + student.calculateSGPA());
            }
             System.out.print("Chethan K S\n1BM23CS074");
    }
}
```

Output:

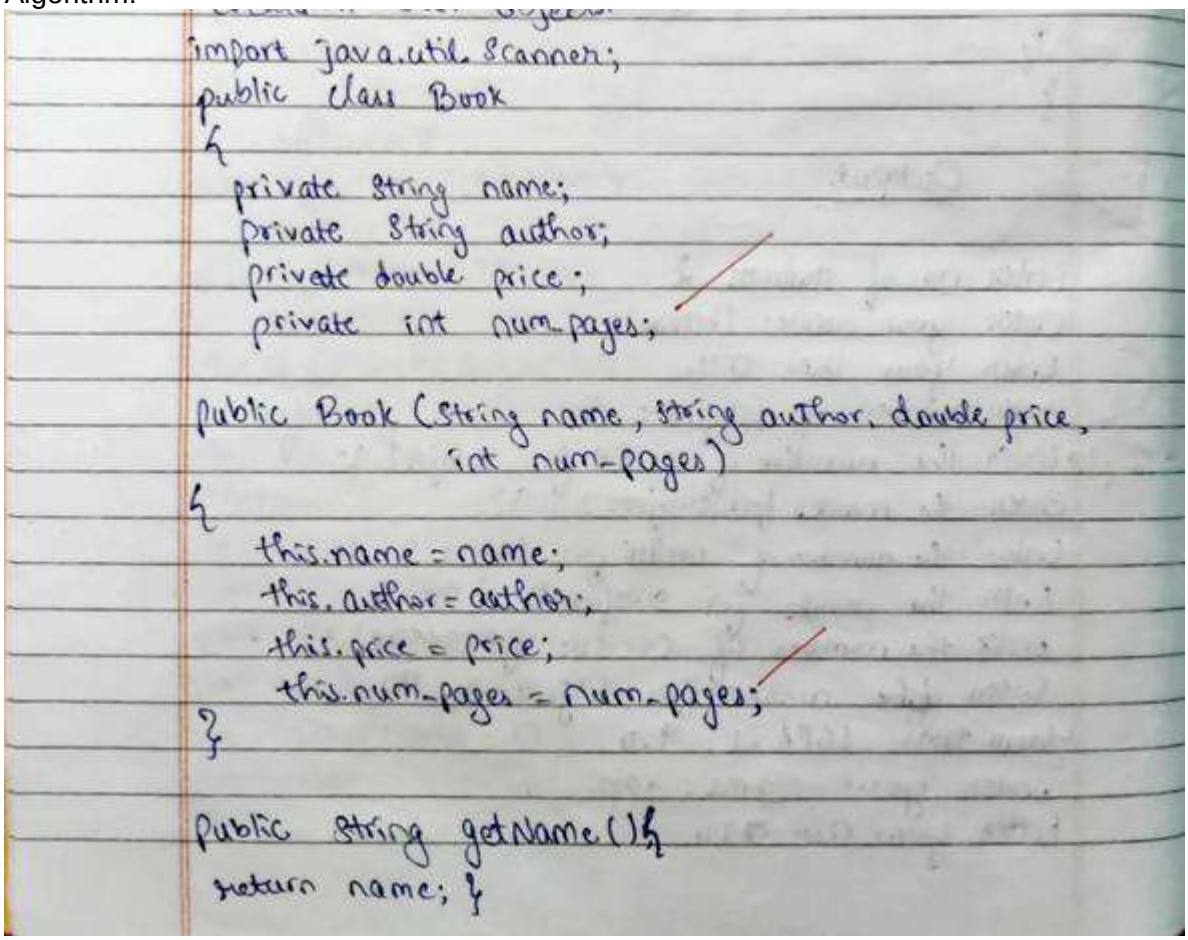
```
Enter no of students 2
Enter your name: Chethan
Enter your USN: 074
Enter number of subjects: 3
Enter the number of credits for subject 1: 2
Enter the marks for subject 1: 92
Enter the number of credits for subject 2: 4
Enter the marks for subject 2: 88
Enter the number of credits for subject 3: 3
Enter the marks for subject 3: 80
Your total SGPA is: 9.0
Enter your name: raja
Enter your USN: 070
Enter number of subjects: 2
Enter the number of credits for subject 1: 3
Enter the marks for subject 1: 99
Enter the number of credits for subject 2: 1
Enter the marks for subject 2: 69
Your total SGPA is: 9.0
Chethan K S
1BM23CS074
```

Program 3

Book Details

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:



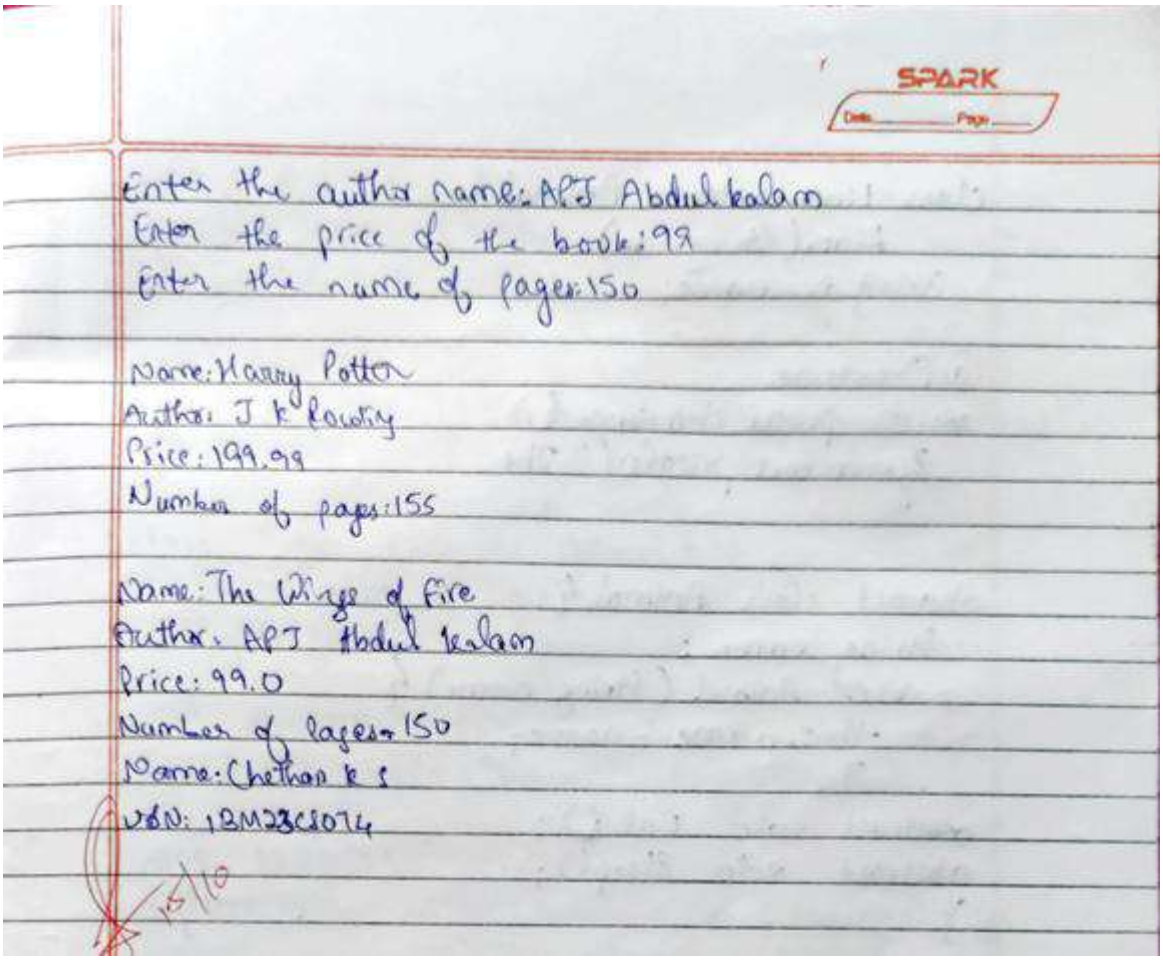
```
import java.util.Scanner;
public class Book
{
  private String name;
  private String author;
  private double price;
  private int num_pages;

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

public String getAuthor (String author) {
    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 num_pages; }

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

public String toString() {
return "Name: " + name + " Author: " + author + "Price: "
    + price + " No of pages: " + num_pages ;
}

public static void main (String args[])
{ Scanner input = new Scanner (System.in);
System.out.print("Enter the number of Books ");
int n = input.nextInt();
```

```
Book[] books = new Book[n];
for (int i=0; i<n; i++)
{
    System.out.print("Details of Book "+(i+1));
    System.out.print("Name of the Book: ");
    String name = input.nextLine();
    System.out.print("\nEnter the author name:");
    String author = input.nextLine();
    System.out.print("\nEnter the price:");
    double price=input.nextDouble();
    System.out.print("\nEnter the no of pages:");
    int num_pages = input.nextInt();
    ~~scanner.nextLine();~~
    ~~input.nextLine();~~

    books[i] = new Book(name, author, price,
                        num_pages);
    } for (Book book : books) {
    System.out.println(book.toString());
    }
    ~~book~~
    input.close();
    }
}
```

Output

```
Enter the number of books: 2
Enter the name of the book: Harry Potter
Enter the author name: J K Rowling
Enter the price of the book: 199.99
Enter the number of pages: 155
Enter the name of the book: The Wings of Fire
```



Code:

```
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 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 num_pages;
    }

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

    @Override
    public String toString() {
        return "\nName: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of pages: "
+ num_pages;
    }

    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print("\nEnter the number of books: ");
        int n = input.nextInt();
        input.nextLine();
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.print("\nEnter the name of the book: ");
            String name = input.nextLine();
            System.out.print("\nEnter the author name: ");
            String author = input.nextLine();
```

```
            System.out.print("\nEnter the price of the book: ");
            double price = input.nextDouble();
            System.out.print("\nEnter the number of pages: ");
            int num_pages = input.nextInt();
            input.nextLine();
            books[i] = new Book(name, author, price, num_pages);
        }

        for (Book book : books) {
            System.out.println(book.toString());
        }

        System.out.print("Chethan K S\n1BM23CS074");
        input.close();
    }
}
```

Output:

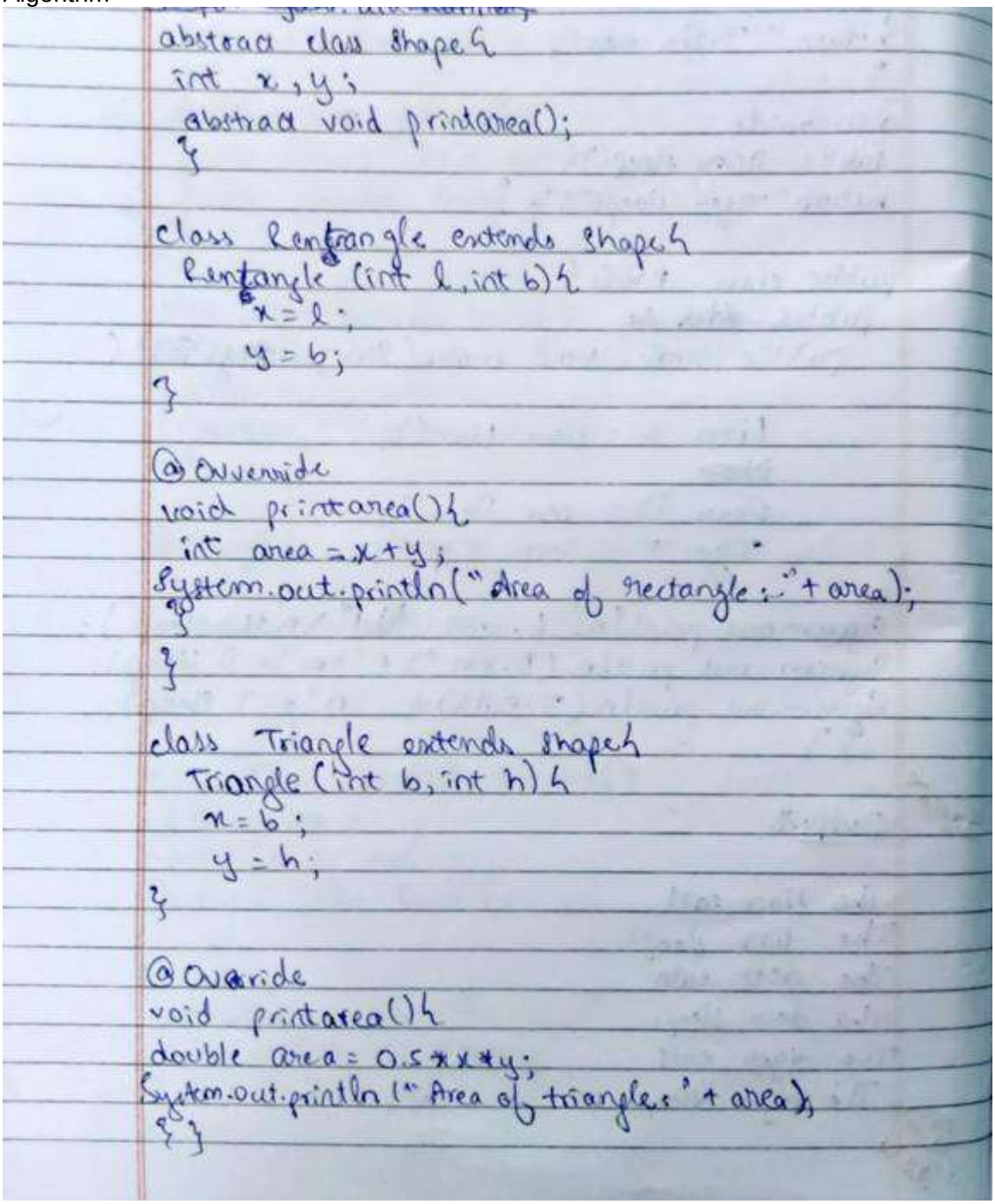
```
 Enter the number of books:2

 Enter the name of the book:Harry Potter

 Enter the author name:J K Rowling

 Enter the price of the book:199.99

 Enter the number of pages:155

 Enter the name of the book:The wings of fire

 Enter the author name:APJ Abdul Kalam

 Enter the price of the book:99

 Enter the number of pages:150

Name:Harry Potter
Author:J K Rowling
Price:199.99
Number of pages:155

Name:The wings of fire
Author:APJ Abdul Kalam
Price:99.0
Number of pages:150
Chethan K S
1BM23CS074
```

Program 4

Abstract Class Shape

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



```
abstract class Shape{
int x,y;
abstract void printarea();
}

class Rectangle extends Shape{
Rectangle (int l, int b){
    x = l;
    y = b;
}

@Override
void printarea(){
int area = x*y;
System.out.println("Area of rectangle:"+area);
}
}

class Triangle extends Shape{
  Triangle (int b, int h){
   x = b;
    y = h;
}

@Override
void printarea(){
double area = 0.5*x*y;
System.out.println("Area of triangle:"+area);
}}
```

```
class Circle extends Shape {
    int radius;
    Circle(int radius) {
        this.radius = radius;
    }

    @Override
    void printarea() {
        double area = 3.14 * radius * radius;
        System.out.println("The area of circle is: " + area);
    }
}

public class Geometry {
    public static void main(String args[]) {
        Shape r = new Rectangle(5, 11);
        Shape t = new Triangle(20, 9);
        Shape c = new Circle(5);
        r.printarea();
        t.printarea();
        c.printarea();
    }
}
```

Output

```
Area of rectangle: 16
Area of triangle: 90.0
Area of circle is: 78.5
```

22/10/24

Code:

```
abstract class Shape {
   int x, y;

   abstract void printarea();
}

class Rectangle extends Shape {
   Rectangle(int l, int b) {
      x = l;
      y = b;
   }

   @Override
   void printarea() {
      int area = x * y;
      System.out.println("Area of rectangle is " + area);
   }
}

class Triangle extends Shape {
   Triangle(int b, int h) {
      x = b;
      y = h;
   }

   @Override
   void printarea() {
      double area = 0.5 * x * y;
      System.out.println("Area of triangle is " + area);
   }
}

class Circle extends Shape {
   int radius;

   Circle(int radius) {
      this.radius = radius;
   }

   @Override
   void printarea() {
      double area = 3.14 * radius * radius;
      System.out.println("Area of circle is " + area);
   }
}
```

```
public class Geometry {
   public static void main(String args[]) {
      Shape r = new Rectangle(5, 11);
      Shape t = new Triangle(20, 9);
      Shape c = new Circle(5);

      r.printarea();
      t.printarea();
      c.printarea();

      System.out.print("Chethan K S\n1BM23CS074");
   }
}
```

Output:

```
Area of rectangle is 16
Area of triangle is 90.0
Area of circle is 78.5
Chethan K S
1BM23CS074
```

Program 5

Bank Details

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

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

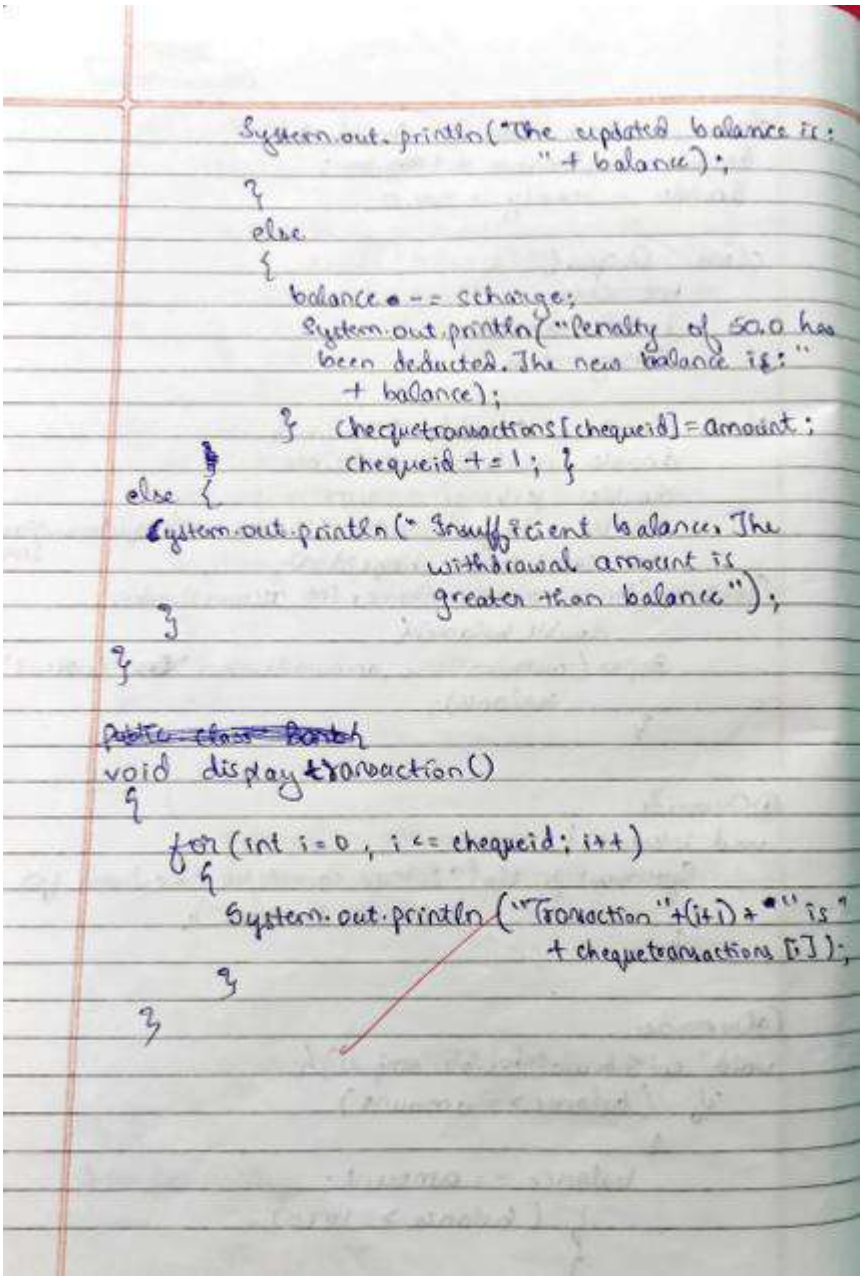
Algorithm:

```
import java.util.*;

abstract class Account{
  String customerName;
  int accountNumber;
  double balance
  String accountType;

Account(String customername, int accountNumber,
          double balance, String accountType) {
    this.customername = customername;
    this.accountNumber = accountnumber;
    this.balance = balance;
    this.accountType = accountType;
    }

void deposit(double amount){
      balance += amount;
      System.out.println("Deposit succesful. New
                          balance = " + balance);
    }

void display(){
  System.out.println("Balance: " + balance); }

Abstract void interest();
abstract void withdraw(double amount);
}
```

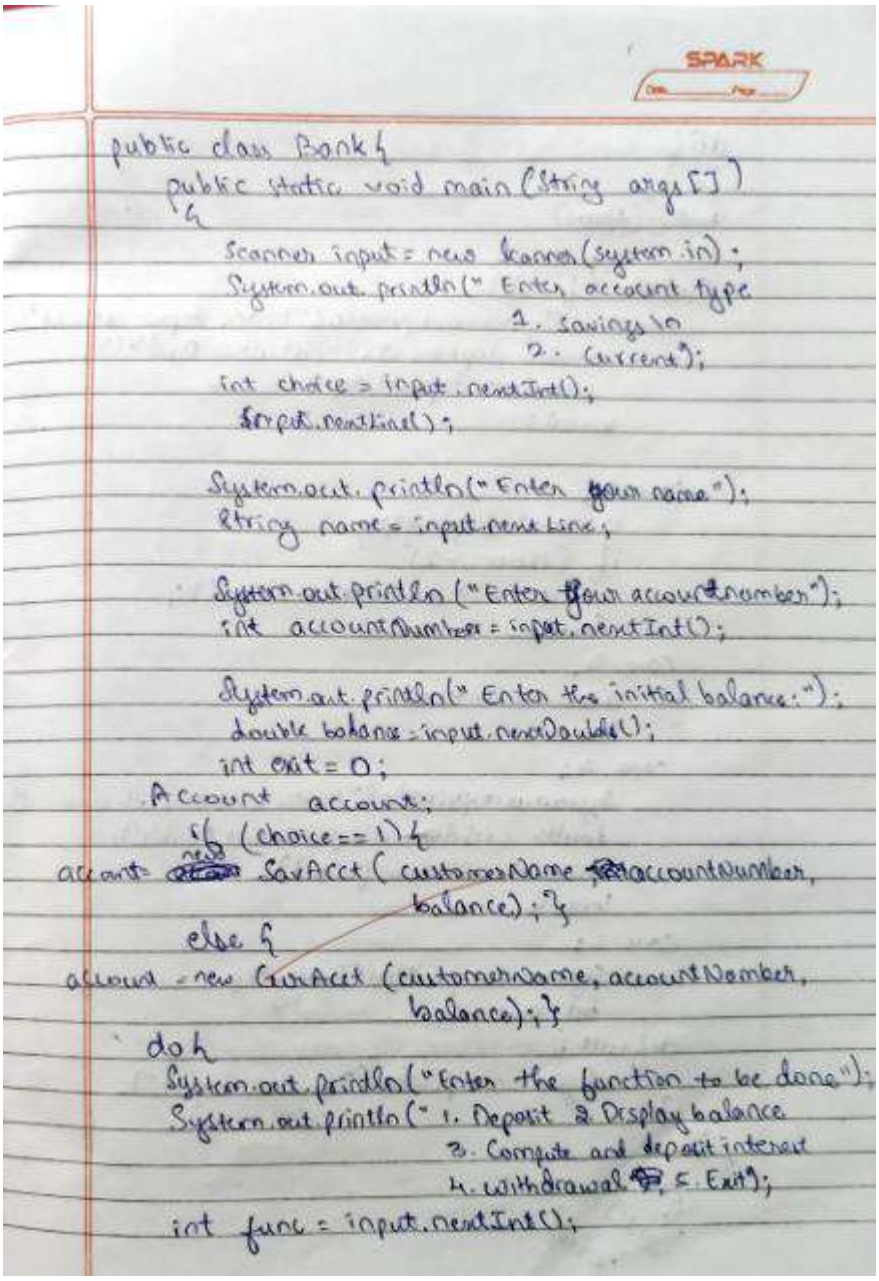
```
class SavAcct extends Account {
    double interestRate = 0.05;

    SavAcct(String customerName, int AccountNumber,
            double balance) {
        super(customerName, accountNumber,
              "Savings", balance);
    }

@Override
void interest(){
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest added. New
                        balance = "+balance);
    }

@Override
void withdraw(double amount) {
 if (balance >= amount)
    {
        balance -= amount;
        System.out.println("withdrawal successful,
          new balance = " + balance);
    }

  else
    {
      System.out.println("Insufficient balance");
    }
}}
```

~~String status~~

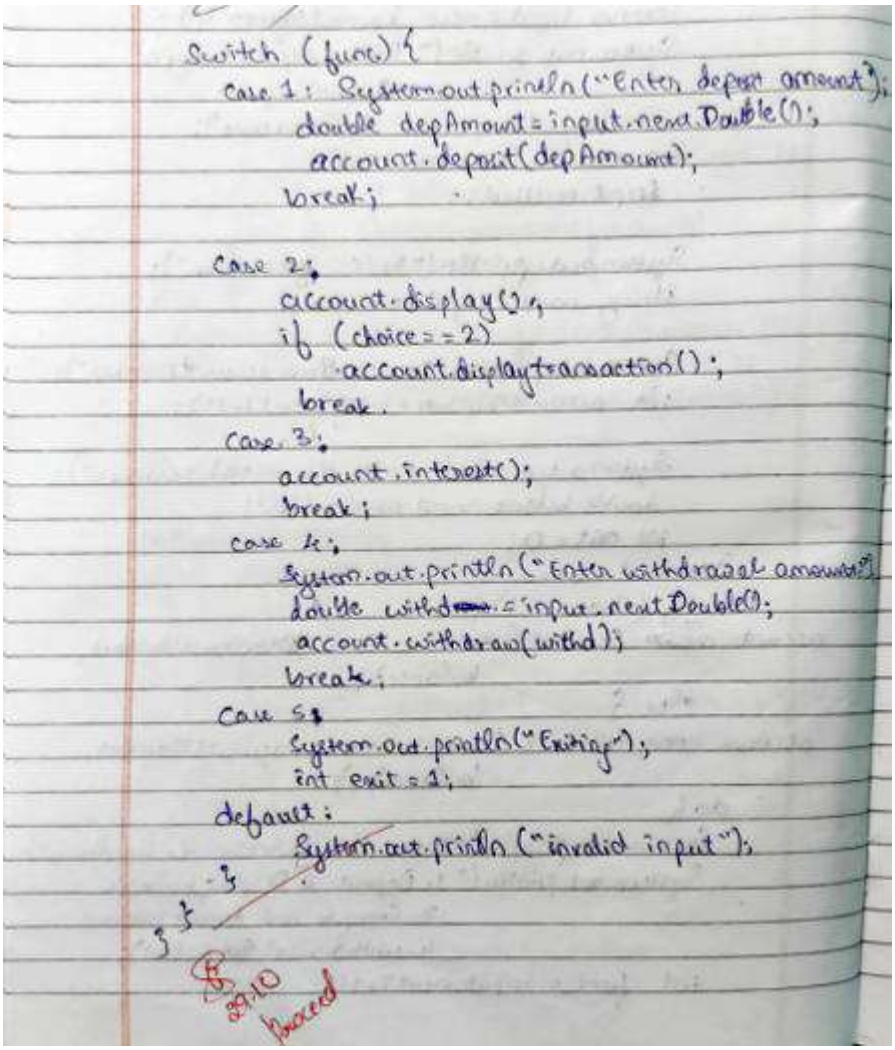
```
class CurAcct extends Account{
    double minbalance = 1000.00
    double charge = 50.00
    double chequetransactions[] = new chequetransactions
                                  [100];
    int ~~checkid = 0~~, chequeid = 0;
CurAcct(String customerName, int accountNumber,
        double balance){
    super(customerName, accountNumber, "Current",
          balance);
    }

@Override
void interest(){
    System.out.println("Interest cannot be calculated for
                        Current Account");
    }

@Override
void withdraw(double amount){
  if (balance >= amount)
    {
      balance -= amount;
        if (balance >= 1000)
        {
```



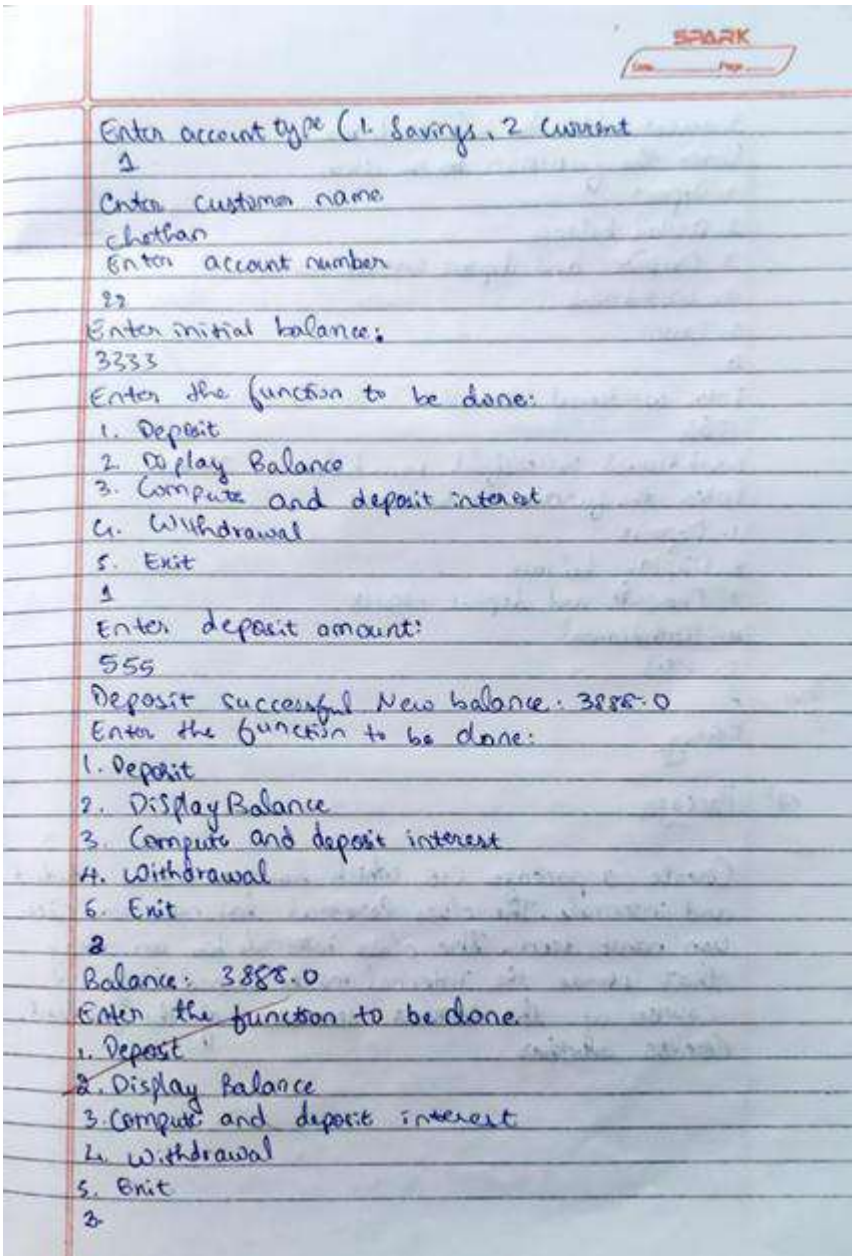
```
        System.out.println("The updated balance is:
                            " + balance);
        }
        else
        {
          balance -= charge;
          System.out.println("Penalty of 50.0 has
          been deducted. The new balance is:"
            + balance);
          } chequetransactions[chequeid] = amount;
            chequeid += 1; }
    else {
     System.out.println(" Insufficient balance, The
                        withdrawal amount is
                        greater than balance");
      }
}

public class Bank
void displaytransaction()
{
   for (int i = 0 ; i <= chequeid; i++)
   {
     System.out.println("Transaction "+(i+1)+" is "
                        + chequetransactions[i]);
   }
}
```



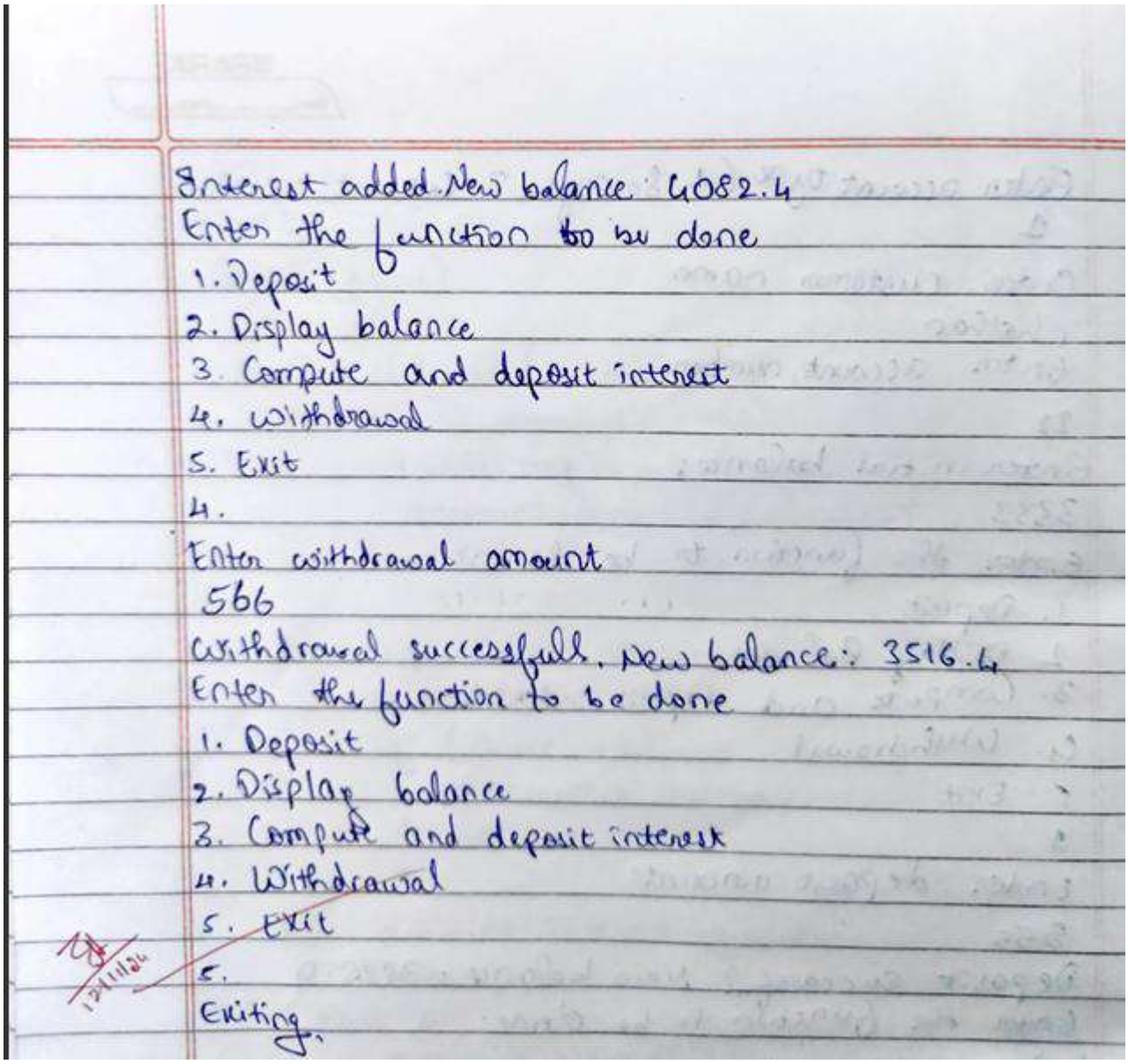
```
public class Bank {
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter account type
                            1. Savings Ac
                            2. Current");
        int choice = input.nextInt();
        input.nextLine();

        System.out.println("Enter your name");
        String name = input.nextLine;

        System.out.println("Enter your account number");
        int accountNumber = input.nextInt();

        System.out.println("Enter the initial balance:");
        double balance = input.nextDouble();
        int exit = 0;
    Account account;
        if (choice == 1) {
account = new SavAcct(customerName, accountNumber,
                      balance);}
        else {
account = new CurAcct(customerName, accountNumber,
                      balance);}
    do {
        System.out.println("Enter the function to be done");
        System.out.println("1. Deposit 2 Display balance
                            3. Compute and deposit interest
                            4. Withdrawal 5. Exit");
        int func = input.nextInt();
```



```
switch (func) {
    case 1: System.out.println("Enter deposit amount");
        double depAmount = input.nextDouble();
        account.deposit(depAmount);
        break;

    case 2:
        account.display();
        if (choice == 2)
            account.displayTransaction();
        break;
    case 3:
        account.interest();
        break;
    case 4:
        System.out.println("Enter withdrawal amount");
        double withd = input.nextDouble();
        account.withdraw(withd);
        break;
    case 5:
        System.out.println("Exiting");
        int exit = 1;
    default:
        System.out.println("invalid input");
    }
}
```



Enter account type (1. Savings, 2 Current
1
Enter customer name
chethan
Enter account number
22
Enter initial balance:
3333
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
555
Deposit successful New balance: 3888.0
Enter the function to be done:
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 3888.0
Enter the function to be done.
1. Deposit
2. Display Balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3



Code:

```
import java.util.*;

abstract class Account {
    String customerName;
    int accountNumber;
    double balance;
    String accountType;

    Account(String customerName, int accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
```

```
        this.balance = balance;
    }

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

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

    abstract void computeInterest();

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    final double interestRate = 0.04;

    SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

    @Override
    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New balance: " + balance);
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    double minBalance = 1000.00;
    double charge = 50.00;
    double[] chequeTransactions = new double[100];
    int chequeId = 0;
```

```
    CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    @Override
    void computeInterest() {
        System.out.println("Interest cannot be calculated for a Current Account.");
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            if (balance >= minBalance) {
                System.out.println("The updated balance is: " + balance);
            } else {
                balance -= charge;
                System.out.println("Penalty of 50.0 has been deducted. The new balance is: " + balance);
            }
            chequeTransactions[chequeId] = amount;
            chequeId++;
        } else {
            System.out.println("Insufficient balance. The withdrawal amount is greater than balance.");
        }
    }

    void displayTransactions() {
        for (int i = 0; i < chequeId; i++) {
            System.out.println("Transaction " + (i + 1) + ": " + chequeTransactions[i]);
        }
    }
}

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

        System.out.println("Enter account type:");
        System.out.println("1. Savings");
        System.out.println("2. Current");
        int choice = input.nextInt();
        input.nextLine();

        System.out.println("Enter your name:");
        String name = input.nextLine();
```

```
System.out.println("Enter your account number:");
int accountNumber = input.nextInt();

System.out.println("Enter the initial balance:");
double balance = input.nextDouble();

Account account;

if (choice == 1) {
   account = new SavAcct(name, accountNumber, balance);
} else {
   account = new CurAcct(name, accountNumber, balance);
}

int exit = 0;

while (exit != 1) {
   System.out.println("\nEnter the function to be done:");
   System.out.println("1. Deposit");
   System.out.println("2. Display balance");
   System.out.println("3. Compute and deposit interest");
   System.out.println("4. Withdrawal");
   System.out.println("5. Exit");

   int func = input.nextInt();

   switch (func) {
      case 1:
         System.out.println("Enter deposit amount:");
         double depAmount = input.nextDouble();
         account.deposit(depAmount);
         break;

      case 2:
         account.displayBalance();
         break;

      case 3:
         account.computeInterest();
         break;

      case 4:
         System.out.println("Enter withdrawal amount:");
         double withdrawAmount = input.nextDouble();
         account.withdraw(withdrawAmount);
         break;
```

```
            case 5:
                exit = 1;
                System.out.println("Exiting");
                break;

            default:
                System.out.println("Invalid input");
        }

        if (choice == 2) {
            ((CurAcct) account).displayTransactions();
        }
    }
    System.out.print("Chethan K S\n1BM23CS074");
    input.close();
  }
}
```

Output:

```
Enter account type:
1. Savings
2. Current
1
Enter your name:
Chethan
Enter your account number:
22
Enter the initial balance:
3333

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
555
Deposit successful. New balance: 3888.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 3888.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest added. New balance: 4043.52

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
4
Enter withdrawal amount:
566
Withdrawal successful. New balance: 3477.52

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
5
Exiting
Chethan K S
1BM23CS074
```

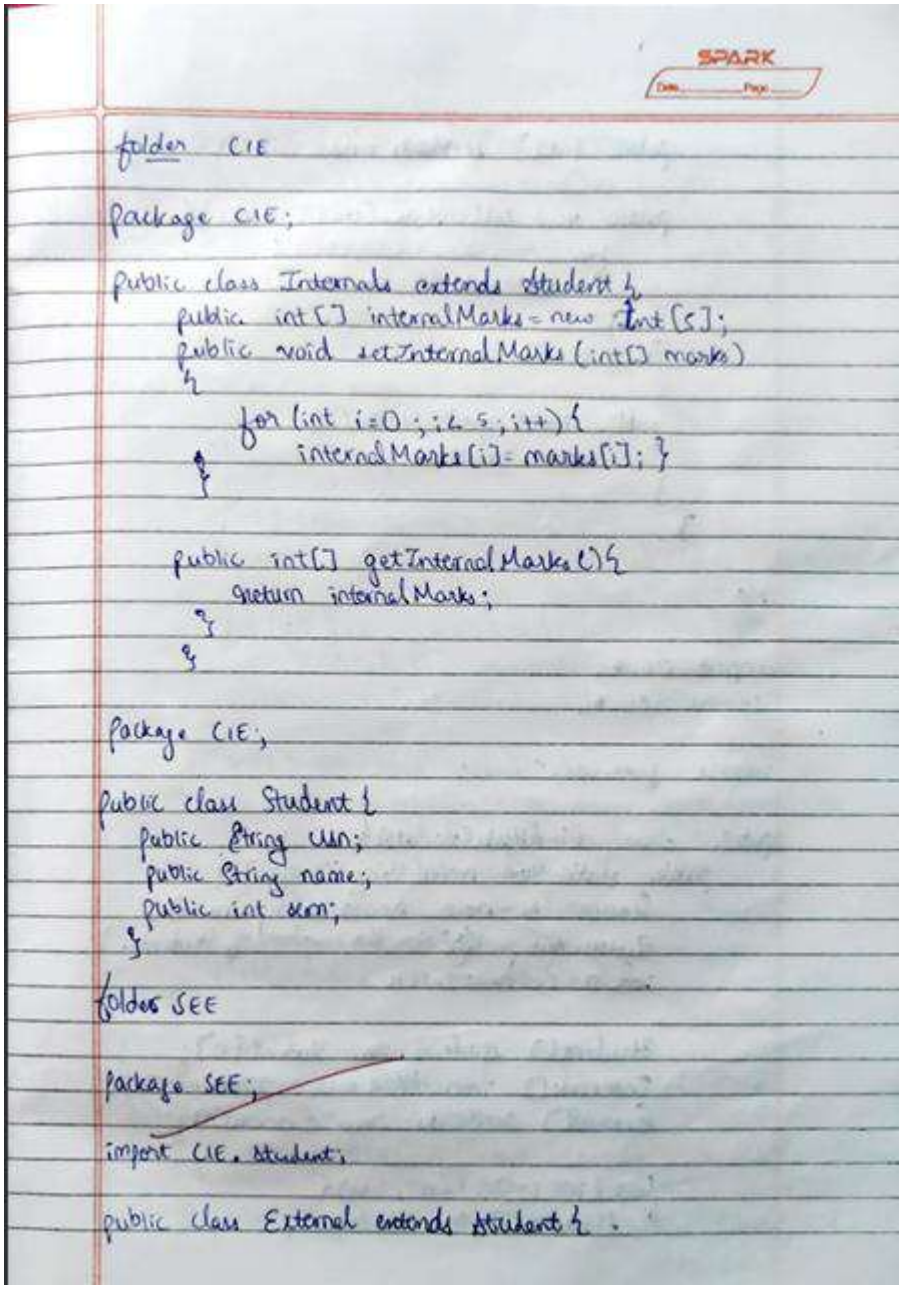
```
Enter account type:
1. Savings
2. Current
2
Enter your name:
Chethan
Enter your account number:
2645
Enter the initial balance:
2534

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
1
Enter deposit amount:
3145
Deposit successful. New balance: 5679.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
2
Balance: 5679.0

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
3
Interest cannot be calculated for a Current Account.

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
4
Enter withdrawal amount:
5131655
Insufficient balance. The withdrawal amount is greater than balance.

Enter the function to be done:
1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdrawal
5. Exit
5
Exiting
Chethan K S
1BM23CS074
```

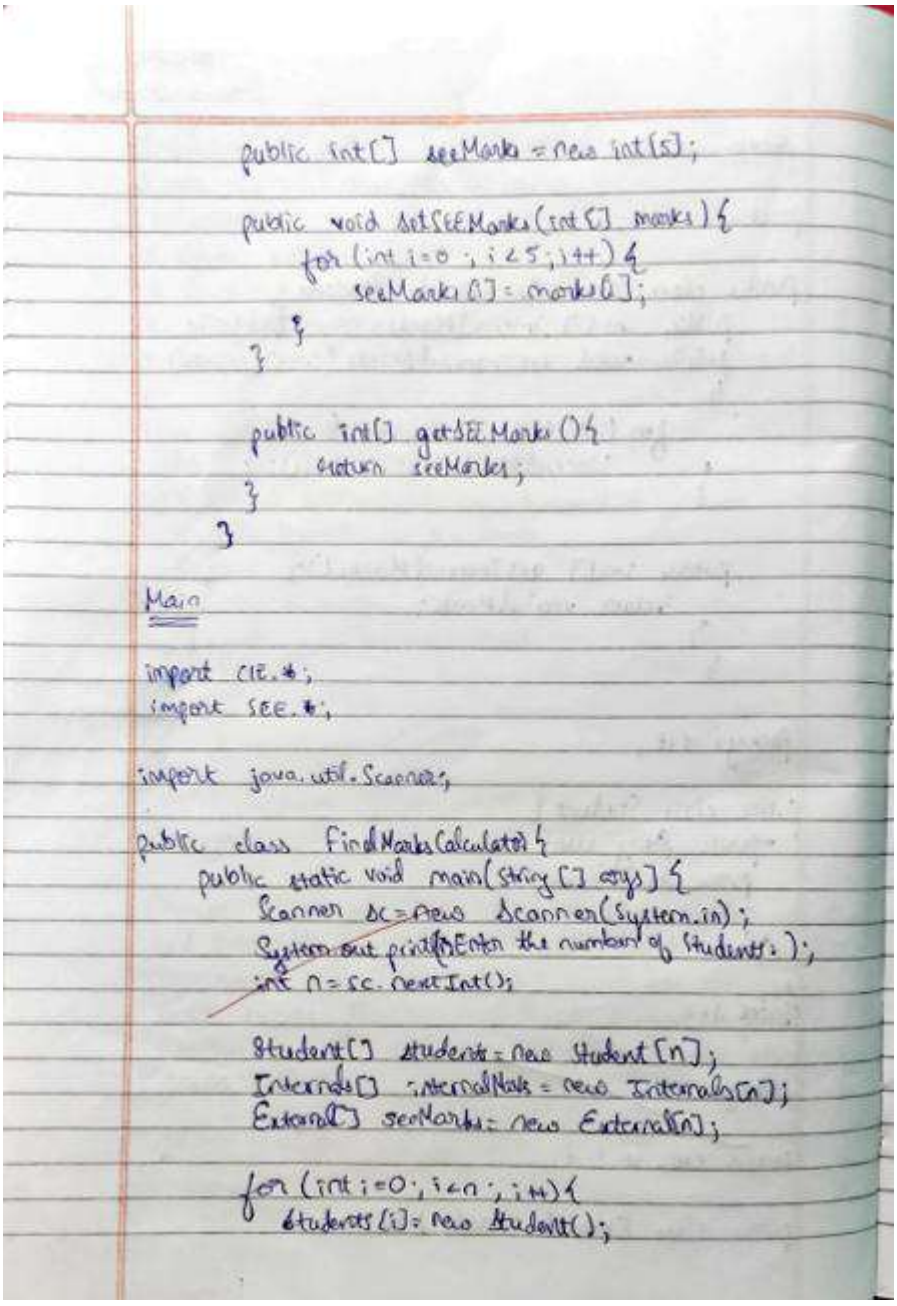
Program 6

Packages

Create a package CIE which has two classes- Student 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 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.

Algorithm:

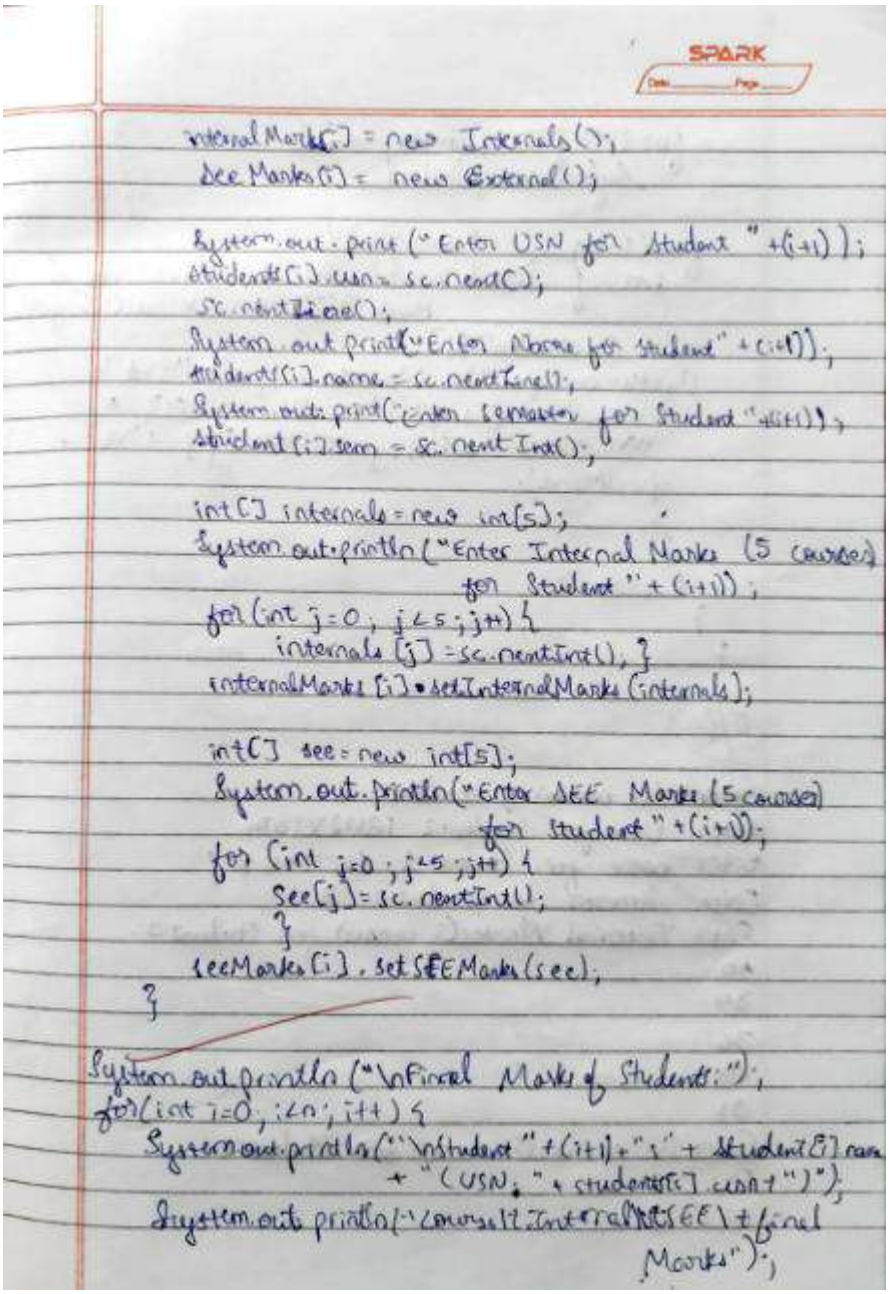




```
    public int[] seeMarks = new int[5];

    public void setSEEMarks(int[] marks){
        for(int i=0 ; i<5; i++){
            seeMarks[i] = marks[i];
        }
    }

    public int[] getSEEMarks(){
        return seeMarks;
    }
}
```

Main

```
import CIE.*;
import SEE.*;

import java.util.Scanner;

public class FinalMarksCalculator{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of students: ");
        int n = sc.nextInt();

        Student[] students = new Student[n];
        Internals[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];

        for(int i=0; i<n; i++){
            students[i] = new Student();
```

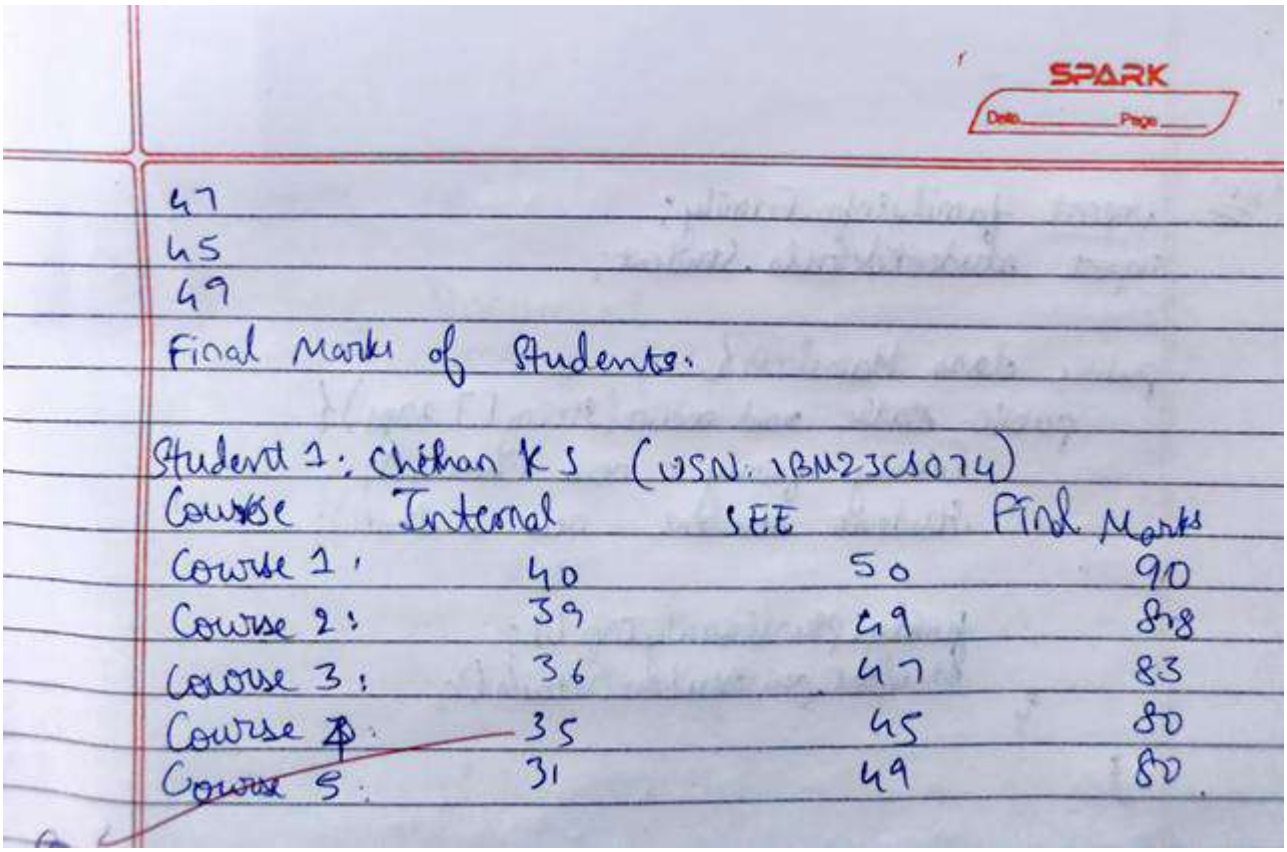


```
internalMarks[i] = new Internals();
seeMarks[i] = new External();

System.out.print("Enter USN for Student " +(i+1));
students[i].usn = sc.next();
sc.nextLine();
System.out.print("Enter Name for Student" +(i+1));
students[i].name = sc.nextLine();
System.out.print("Enter Semester for Student "+(i+1));
students[i].sem = sc.nextInt();

int[] internals = new int[5];
System.out.println("Enter Internal Marks (5 courses)
                    for Student " + (i+1));
for(int j=0; j<5; j++) {
    internals[j] = sc.nextInt(); }
internalMarks[i].setInternalMarks(internals);

int[] see = new int[5];
System.out.println("Enter SEE Marks (5 courses)
                    for Student" +(i+1));
for (int j=0; j<5; j++) {
    see[j] = sc.nextInt();
    }
seeMarks[i].setSEEMarks(see);
}

System.out.println("\nFinal Marks of Students:");
for(int i=0; i<n; i++) {
    System.out.println("\nStudent " +(i+1)+": " + students[i].name
                    + " (USN: " + students[i].usn + ")");
    System.out.println("Course\t Internal\tSEE\t Final
                    Marks");
```

```
for(int j=0; j<5; j++) {
    int finalMark = internalMarks[i].getInternal
                    Marks()[j] + seeMarks[i].get
    SEEMarks()[j];
System.out.println("Course" + (j+1) + ":\t" +
  internalMarks[i].getInternalMarks()[j] +
  "\t\t" + seeMarks[i].getSEEMarks[j] + "\t" +
  finalMark);
  }
  }
  sc.close();
  }
}
```

O/P:

```
Enter the number of students: 1
Enter USN for student1 1BM23CS074
Enter name for Student 1 CHETHAN K
Enter Semester for student 1: 3
Enter Internal Marks(5 courses) for student 1:
40
39
36
35
31
Enter SEE Marks(5 courses) for Student 1
50
49
```



Code:

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;
}
```

```
package CIE;

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

    public void setInternalMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }

    public int[] getInternalMarks() {
        return internalMarks;
```

```
    }
}

package SEE;

import CIE.Student;

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

    public void setSEEMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            seeMarks[i] = marks[i];
        }
    }

    public int[] getSEEMarks() {
        return seeMarks;
    }
}

import CIE.*;
import SEE.*;

import java.util.Scanner;

public class FinalMarksCalculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();

        Student[] students = new Student[n];
        Internals[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            internalMarks[i] = new Internals();
            seeMarks[i] = new External();

            System.out.print("Enter USN for Student " + (i + 1) + ": ");
            students[i].usn = sc.next();
            sc.nextLine();

            System.out.print("Enter Name for Student " + (i + 1) + ": ");
            students[i].name = sc.nextLine();
```

```
            System.out.print("Enter Semester for Student " + (i + 1) + ": ");
            students[i].sem = sc.nextInt();

            int[] internals = new int[5];
            System.out.println("Enter Internal Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                internals[j] = sc.nextInt();
            }
            internalMarks[i].setInternalMarks(internals);

            int[] see = new int[5];
            System.out.println("Enter SEE Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                see[j] = sc.nextInt();
            }
            seeMarks[i].setSEEMarks(see);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ": " + students[i].name + " (USN: " +
students[i].usn + ")");
            System.out.println("Course\tInternal\tSEE\tFinal Marks");
            for (int j = 0; j < 5; j++) {
                int finalMark = internalMarks[i].getInternalMarks()[j] + seeMarks[i].getSEEMarks()[j];
                System.out.println("Course " + (j + 1) + ":\t" + internalMarks[i].getInternalMarks()[j] +
"\t\t" + seeMarks[i].getSEEMarks()[j] + "\t" + finalMark);
            }
        }
        System.out.print("Chethan K S\n1BM23CS074");
        sc.close();
    }
}
```

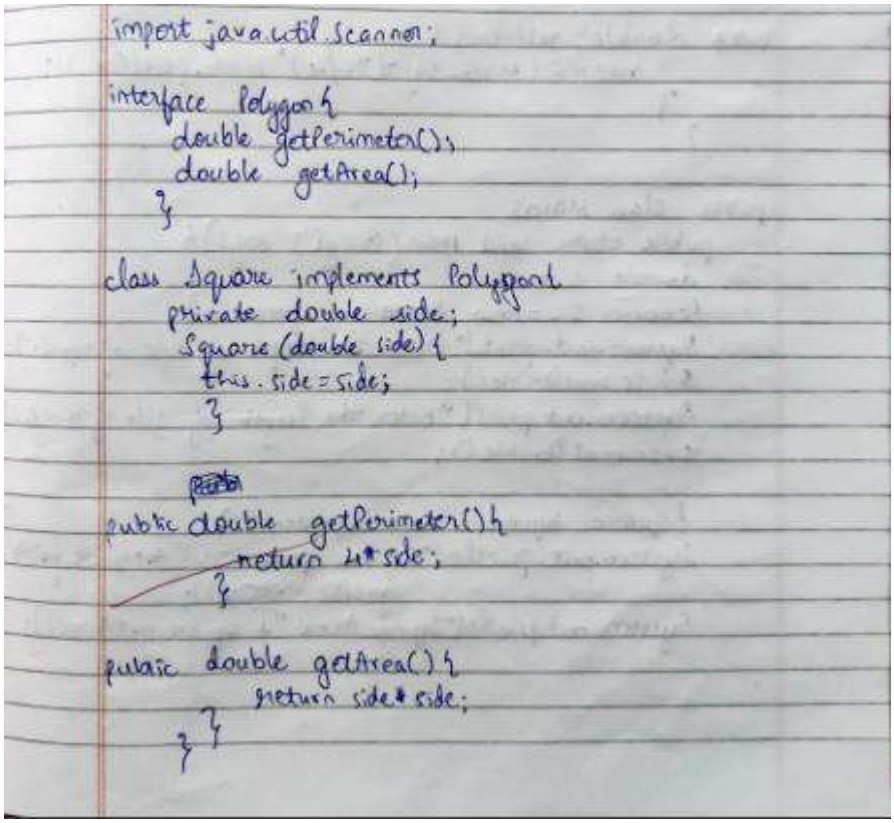
Output:

```
Enter the number of students: 1
Enter USN for Student 1: 1BM23CS074
Enter Name for Student 1: CHETHAN K S
Enter Semester for Student 1: 3
Enter Internal Marks (5 courses) for Student 1:
40
39
36
35
31
Enter SEE Marks (5 courses) for Student 1:
50
49
47
45
49

Final Marks of Students:

Student 1: CHETHAN K S (USN: 1BM23CS074)
Course  Internal        SEE     Final Marks
Course 1:       40              50      90
Course 2:       39              49      88
Course 3:       36              47      83
Course 4:       35              45      80
Course 5:       31              49      80
Chethan K S
1BM23CS074
```

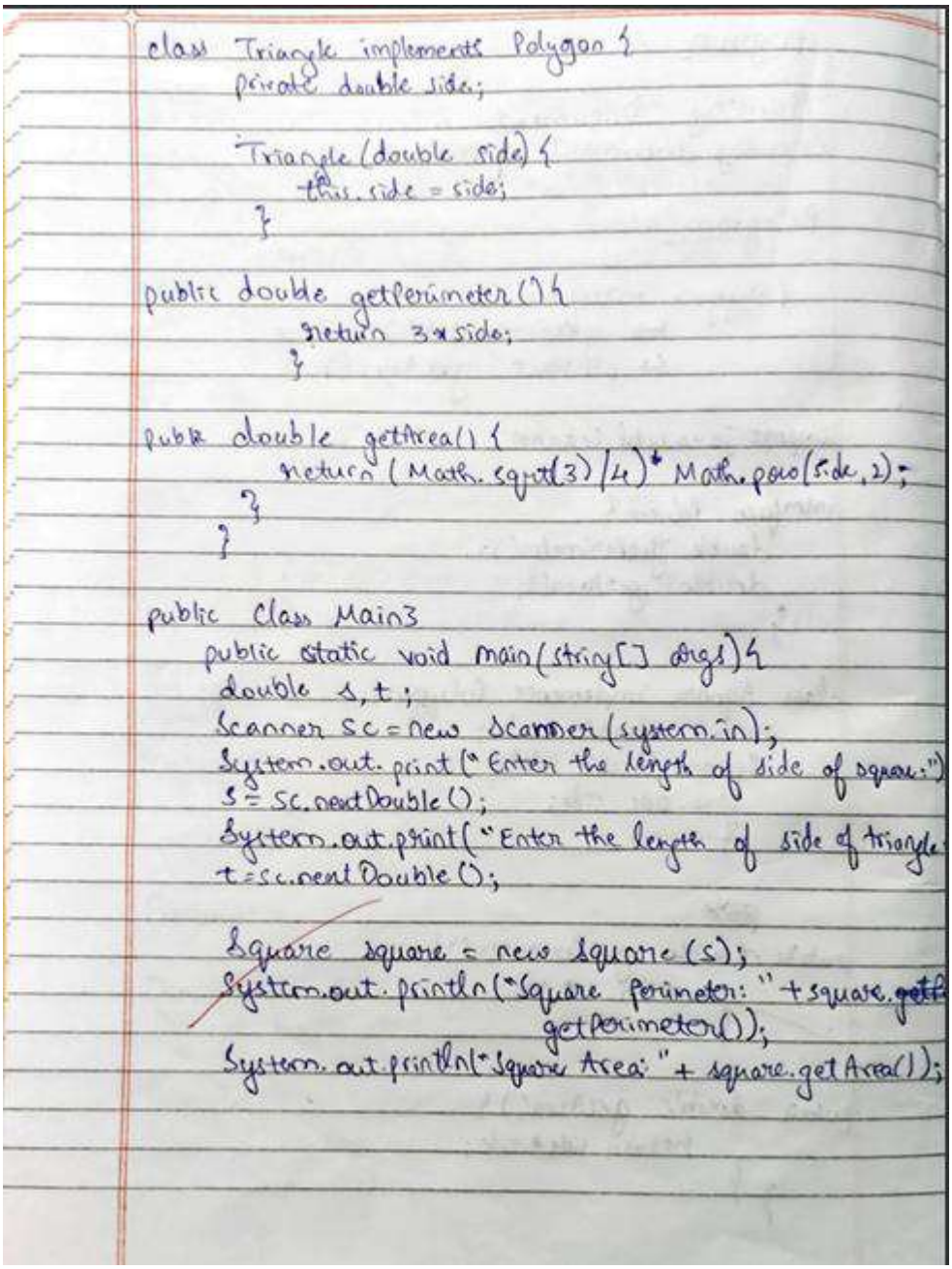
Program 7

Interfaces

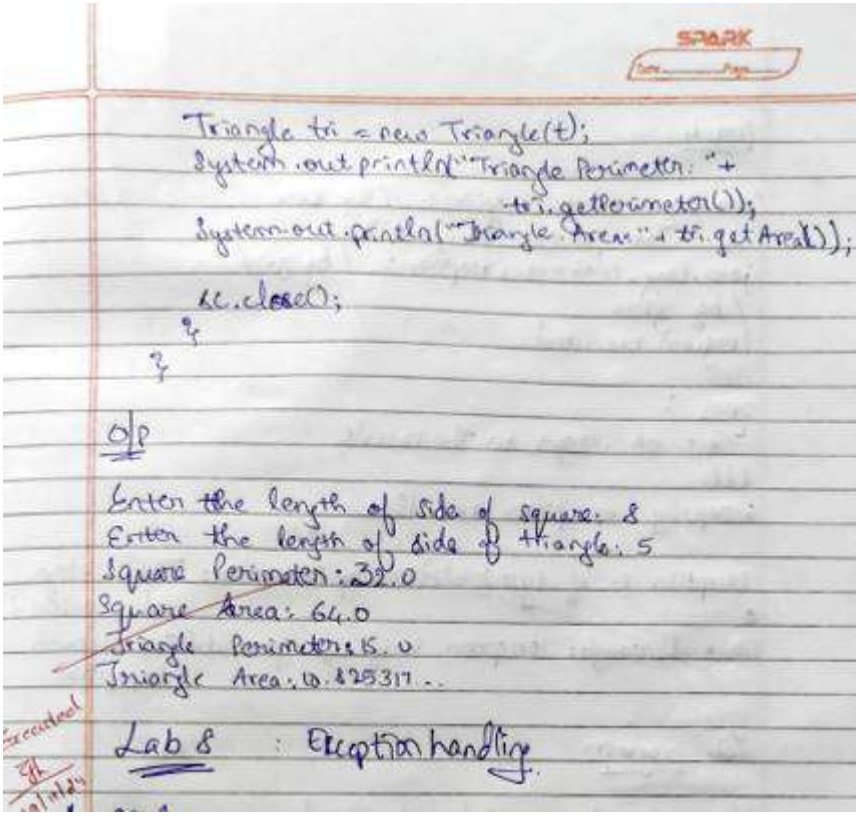
Algorithm:



```
import java.util.Scanner;

interface Polygon {
    double getPerimeter();
    double getArea();
}

class Square implements Polygon
    private double side;
    Square(double side) {
        this.side = side;
    }

public double getPerimeter() {
        return 4*side;
    }

public double getArea() {
        return side*side;
    }
}
```



```
class Triangle implements Polygon {
    private double side;

    Triangle (double side) {
        this.side = side;
    }

public double getPerimeter() {
        return 3*side;
    }

Publ double getArea() {
        return (Math.sqrt(3)/4)* Math.pow(side,2);
    }
}

public Class Main3
    public static void main(String[] args) {
    double s,t;
    Scanner sc = new Scanner(system.in);
    System.out.print("Enter the length of side of square:")
    s = sc.nextDouble();
    System.out.print("Enter the length of side of triangle
    t=sc.nextDouble();

    Square square = new Square(s);
    System.out.println("Square perimeter: "+square.getP
                        getPerimeter());
    System.out.println("Square Area: "+ square.getArea());
```



Code:

```
import java.util.Scanner;

interface Polygon {
    double getPerimeter();
    double getArea();
}

class Square implements Polygon {
    private double side;

    Square(double side) {
        this.side = side;
    }

    @Override
    public double getPerimeter() {
        return 4 * side;
    }

    @Override
    public double getArea() {
        return side * side;
```

```
    }
}

class Triangle implements Polygon {
    private double side;

    Triangle(double side) {
        this.side = side;
    }

    @Override
    public double getPerimeter() {
        return 3 * side;
    }

    @Override
    public double getArea() {
        return (Math.sqrt(3) / 4) * Math.pow(side, 2);
    }
}

public class maininterface {
    public static void main(String[] args) {
        double s, t;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the length of side of square: ");
        s = sc.nextDouble();

        System.out.print("Enter the length of side of triangle: ");
        t = sc.nextDouble();

        Square square = new Square(s);
        System.out.println("Square Perimeter: " + square.getPerimeter());
        System.out.println("Square Area: " + square.getArea());

        Triangle tri = new Triangle(t);
        System.out.println("Triangle Perimeter: " + tri.getPerimeter());
        System.out.println("Triangle Area: " + tri.getArea());
        System.out.print("Chethan K S\n1BM23CS074");
        sc.close();
    }
}
```

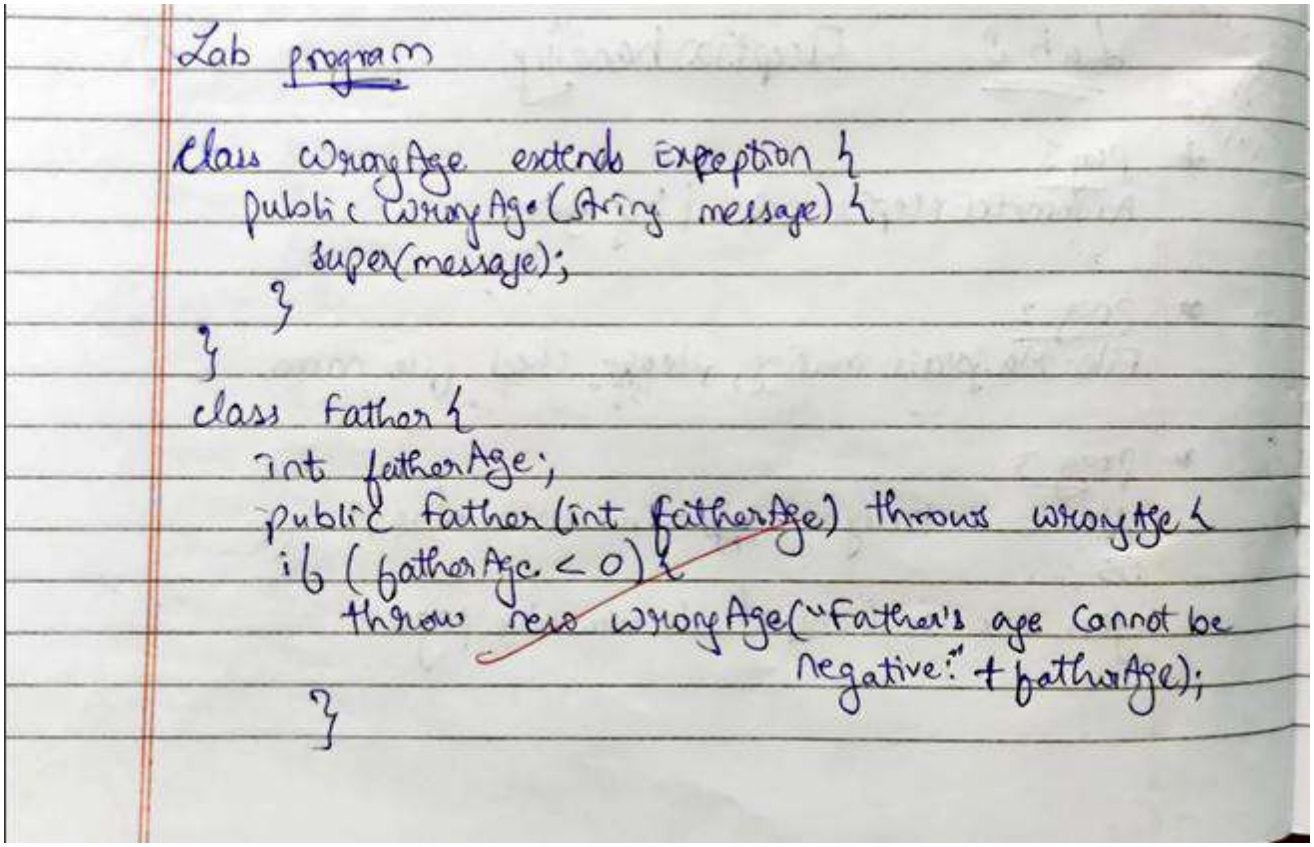
Output:

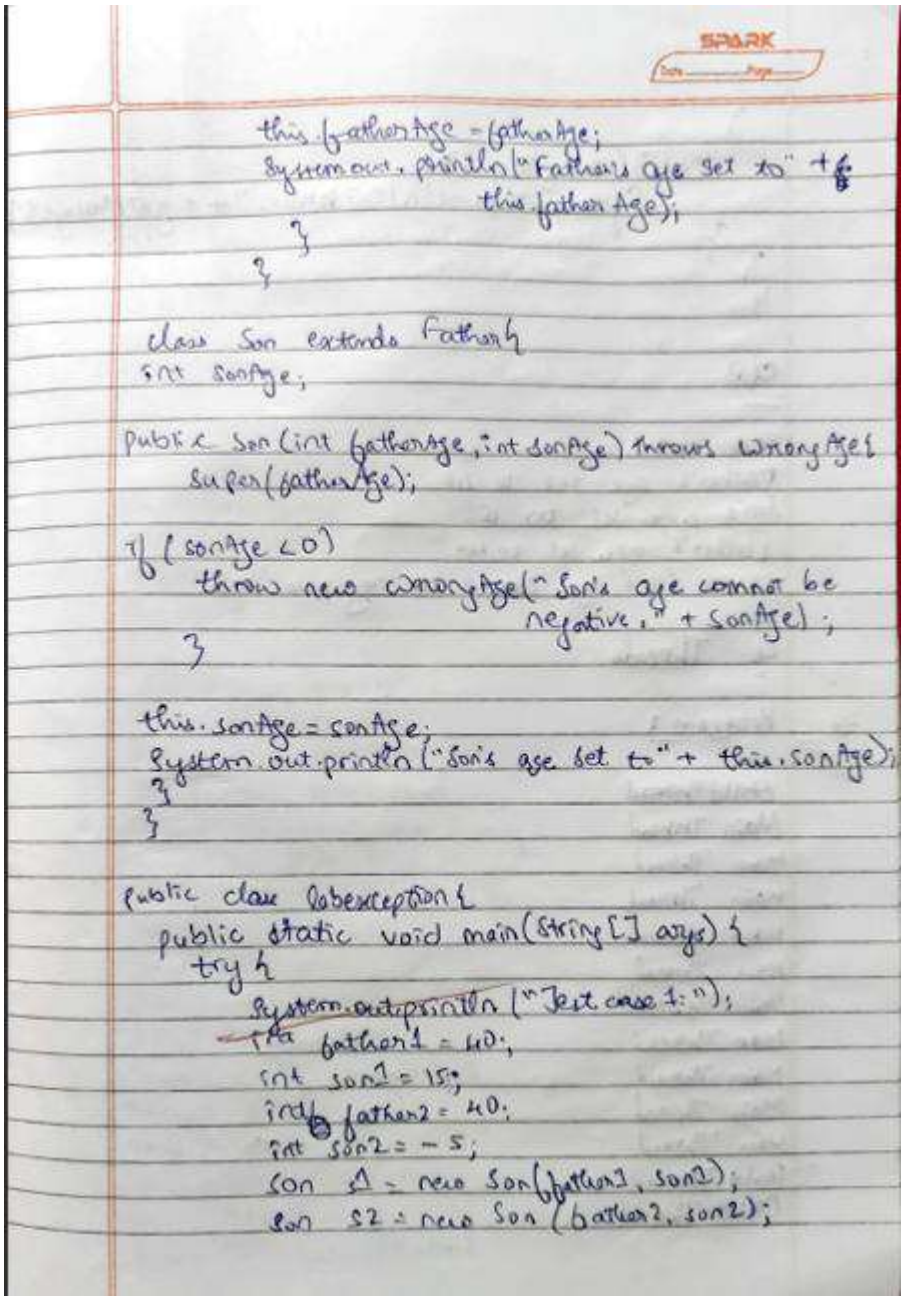
```
Enter the length of side of square: 8
Enter the length of side of triangle: 5
Square Perimeter: 32.0
Square Area: 64.0
Triangle Perimeter: 15.0
Triangle Area: 10.825317547305483
Chethan K S
1BM23CS074
```

Program 8

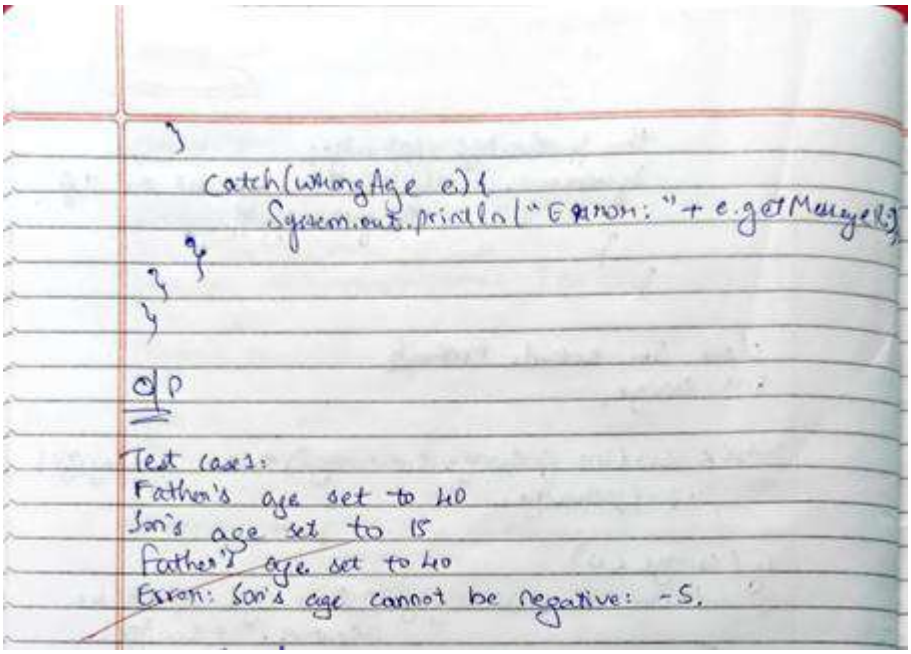
Exception Handling
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the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

Algorithm:





```
        this.fatherAge = fatherAge;
        System.out.println("Father's age set to" +
                           this.fatherAge);
    }
}

class Son extends Father{
int sonAge;

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

if (sonAge < 0)
    throw new WrongAge("Son's age cannot be
                        negative," + sonAge);
    }

this.sonAge = sonAge;
System.out.println("Son's age set to" + this.sonAge);
}
}

public class JobException{
  public static void main(String[] args){
    try{
        System.out.println("Test case 1:");
        int father1 = 40;
        int son1 = 15;
        int father2 = 40;
        int son2 = -5;
        Son s1 = new Son(father1, son1);
        Son s2 = new Son(father2, son2);
```



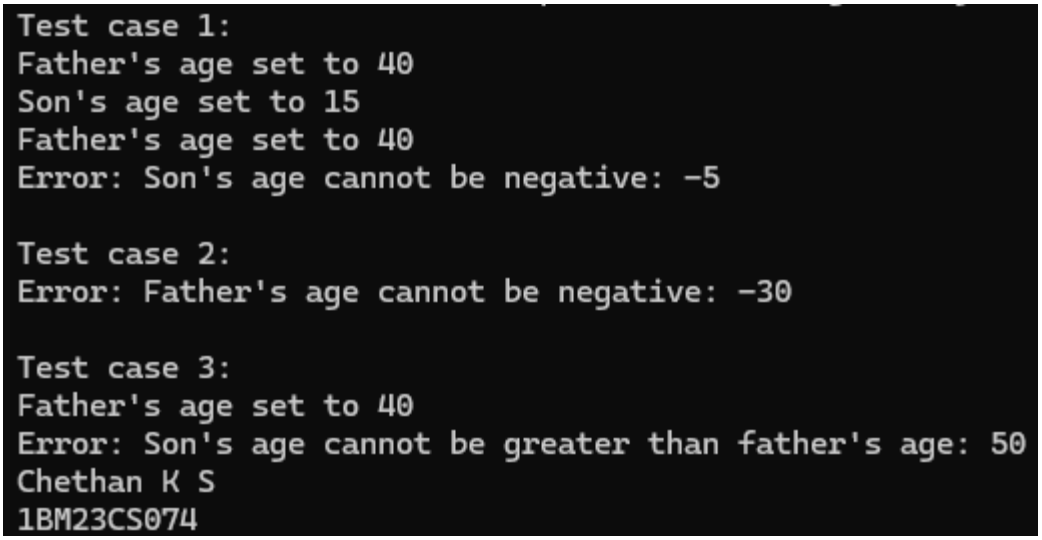
Code:

```
class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    int fatherAge;

    public Father(int fatherAge) throws WrongAge {
        if (fatherAge < 0) {
            throw new WrongAge("Father's age cannot be negative: " + fatherAge);
        }
        this.fatherAge = fatherAge;
        System.out.println("Father's age set to " + this.fatherAge);
    }
}

class Son extends Father {
    int sonAge;

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

        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative: " + sonAge);
        }
```

```
        if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age: " + sonAge);
        }

        this.sonAge = sonAge;
        System.out.println("Son's age set to " + this.sonAge);
    }
}

public class labexception {
    public static void main(String[] args) {
        try {
            System.out.println("Test case 1:");
            int father1 = 40;
            int son1 = 15;
            int father2 = 40;
            int son2 = -5;

            try {
                Son s1 = new Son(father1, son1);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            try {
                Son s2 = new Son(father2, son2);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            System.out.println("\nTest case 2:");
            int father3 = -30;
            int son3 = 10;

            try {
                Son s3 = new Son(father3, son3);
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

            System.out.println("\nTest case 3:");
            int father4 = 40;
            int son4 = 50;

            try {
                Son s4 = new Son(father4, son4);
```

```
            } catch (WrongAge e) {
                System.out.println("Error: " + e.getMessage());
            }

        } finally {
            System.out.print("Chethan K S\n1BM23CS074");
        }
    }
}
```

Output:

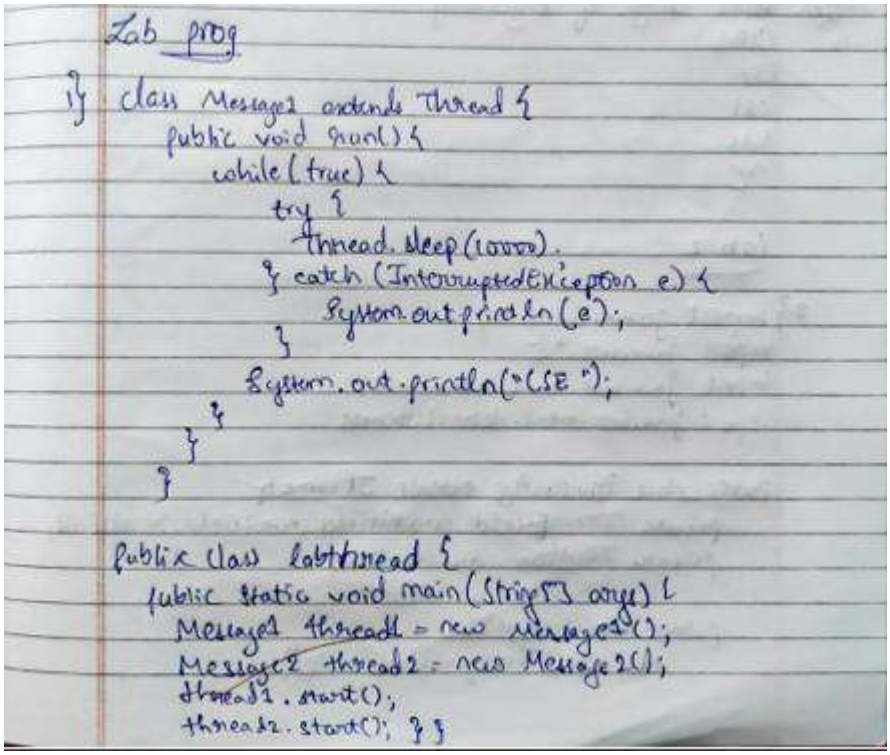


Program 9

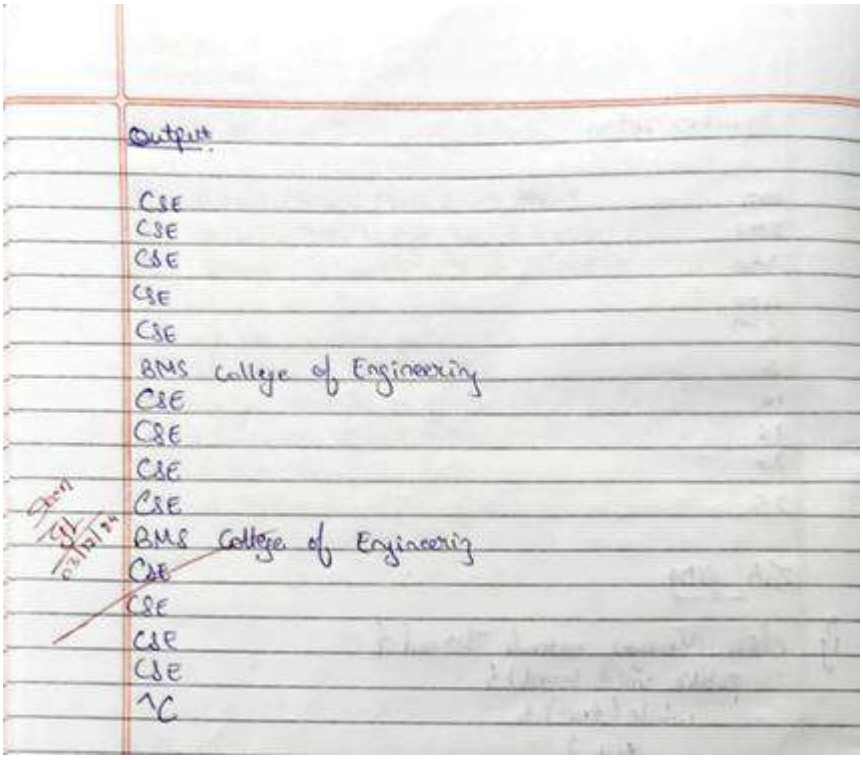
Threads

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:



```
Lab prog
class Message2 extends Thread {
    public void run() {
        while(true) {
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
            System.out.println("CSE");
        }
    }
}

Public class Labthread {
    public static void main(String[] args) {
        Message1 thread1 = new Message1();
        Message2 thread2 = new Message2();
        thread1.start();
        thread2.start(); } }
```

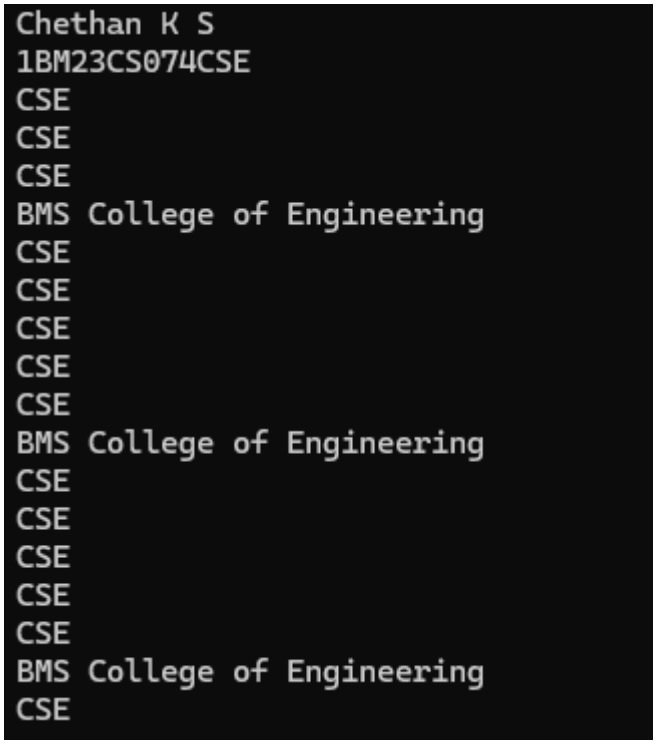


Code:

```
class Message1 extends Thread {
   public void run() {
      while (true) {
         try {
            Thread.sleep(10000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
         System.out.println("BMS College of Engineering");
      }
   }
}

class Message2 extends Thread {
   public void run() {
      while (true) {
         try {
            Thread.sleep(2000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
         System.out.println("CSE");
      }
   }
```

```
}

public class labthread {
    public static void main(String[] args) {
        Message1 thread1 = new Message1();
        Message2 thread2 = new Message2();
        System.out.print("Chethan K S\n1BM23CS074");

        thread1.start();
        thread2.start();
    }
}
```

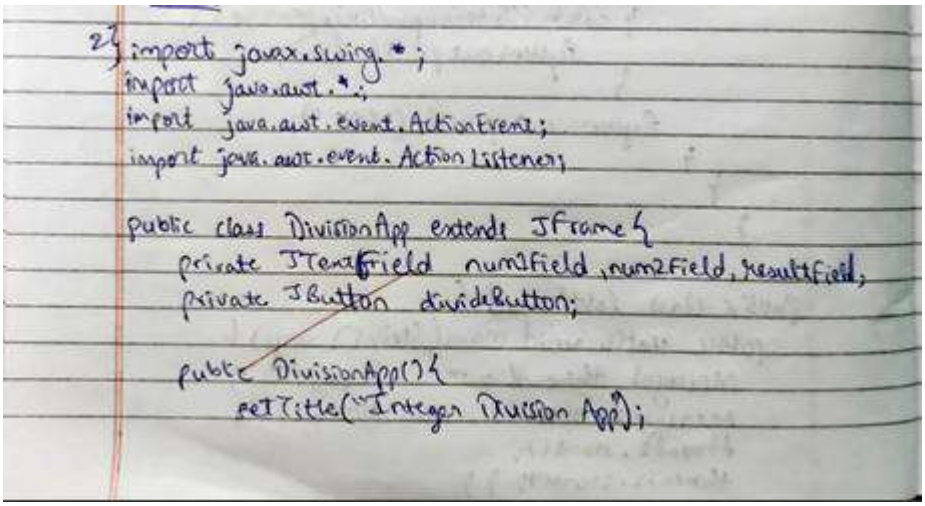
Output:

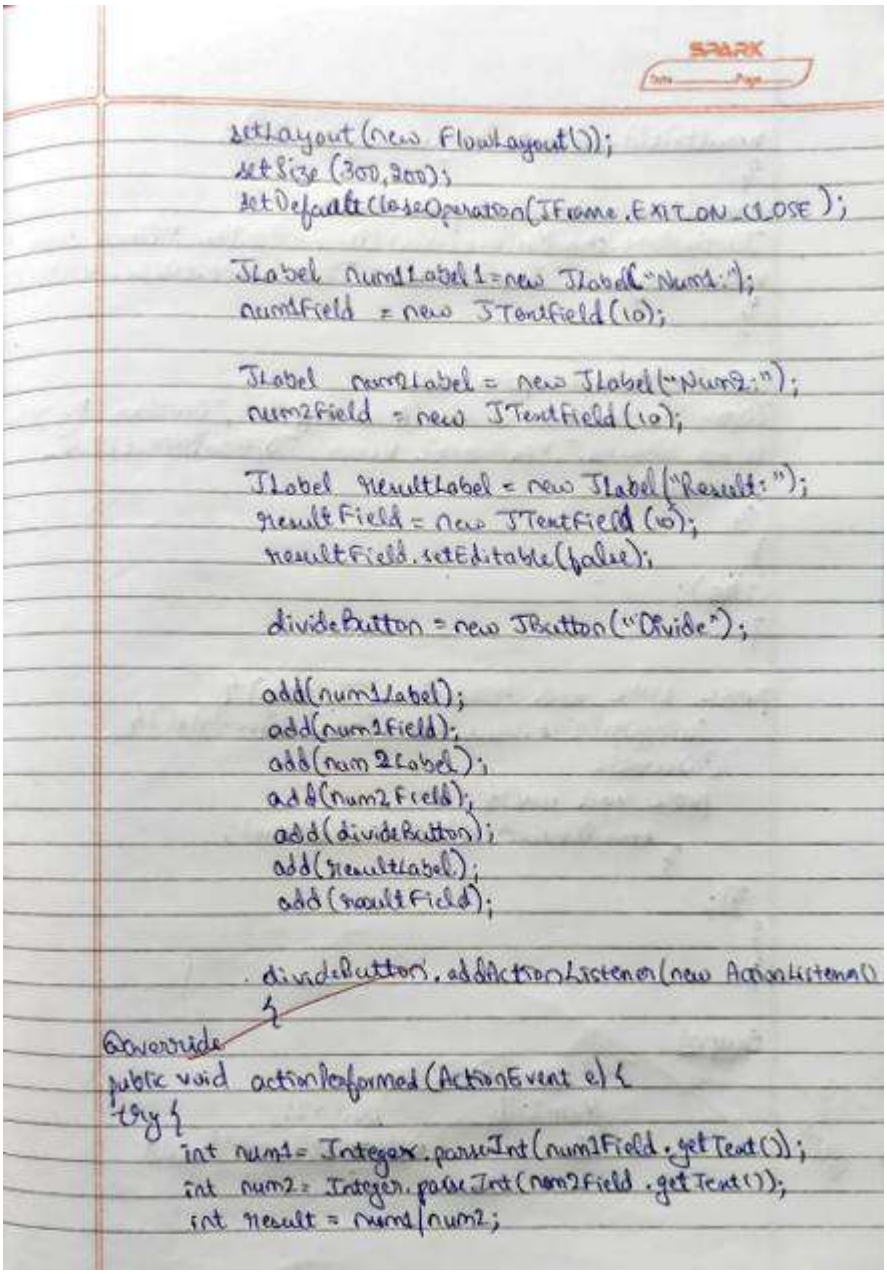


Program 10

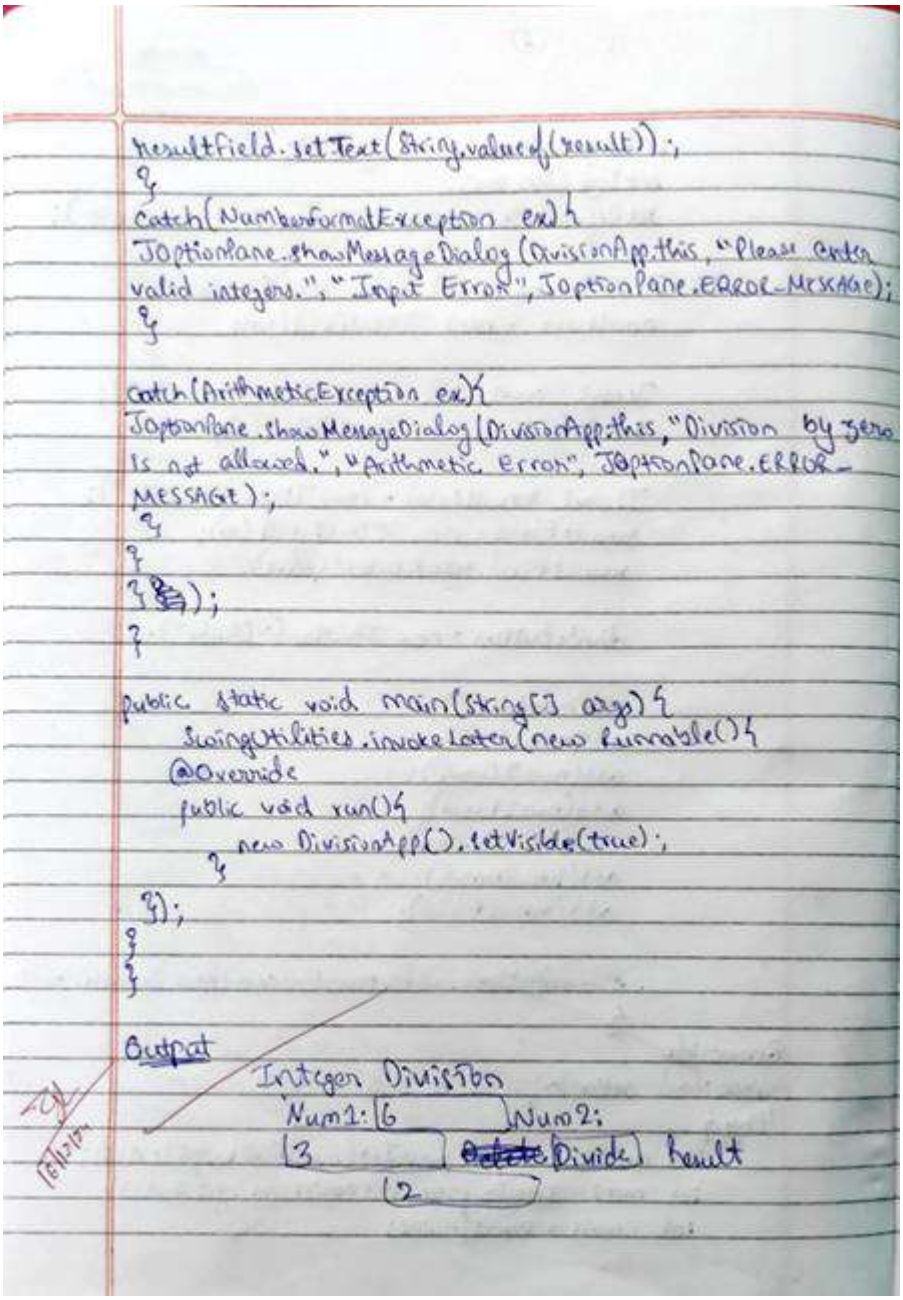
GUI – Java Swing

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.





```
setLayout(new FlowLayout());
setSize(300,200);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JLabel num1Label1=new JLabel("Num1:");
num1Field = new JTextField(10);

JLabel num2Label = new JLabel("Num2:");
num2Field = new JTextField(10);

JLabel resultLabel = new JLabel("Result:");
resultField = new JTextField(10);
resultField.setEditable(false);

divideButton = new JButton("Divide");

add(num1Label);
add(num1Field);
add(num2Label);
add(num2Field);
add(divideButton);
add(resultLabel);
add(resultField);

divideButton.addActionListener(new ActionListener()
{
@Override
public void actionPerformed(ActionEvent e){
try{
    int num1= Integer.parseInt(num1Field.getText());
    int num2= Integer.parseInt(num2Field.getText());
    int result = num1/num2;
```



Code:

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

public class DivisionApp extends JFrame {
    private JTextField num1Field, num2Field, resultField;
    private JButton divideButton;
```

```
    public DivisionApp() {
        setTitle("Integer Division App");
        setLayout(new FlowLayout());
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField(10);

        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField(10);

        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField(10);
        resultField.setEditable(false);

        divideButton = new JButton("Divide");

        add(num1Label);
        add(num1Field);
        add(num2Label);
        add(num2Field);
        add(divideButton);
        add(resultLabel);
        add(resultField);

        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));

                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(DivisionApp.this, "Please enter valid integers.",
"Input Error", JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero is not allowed.",
"Arithmetic Error", JOptionPane.ERROR_MESSAGE);
                } finally {

                    System.out.println("Chethan K S\n1BM23CS074");
                }
            }
```

```
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new DivisionApp().setVisible(true);
            }
        });
    }
}
```

Output: