# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB REPORT**
**on**

# Object Oriented Java Programming

# (23CS3PCOOJ)

***Submitted by***

**BHOOMI UDEDH (1BM23CS066)**

***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

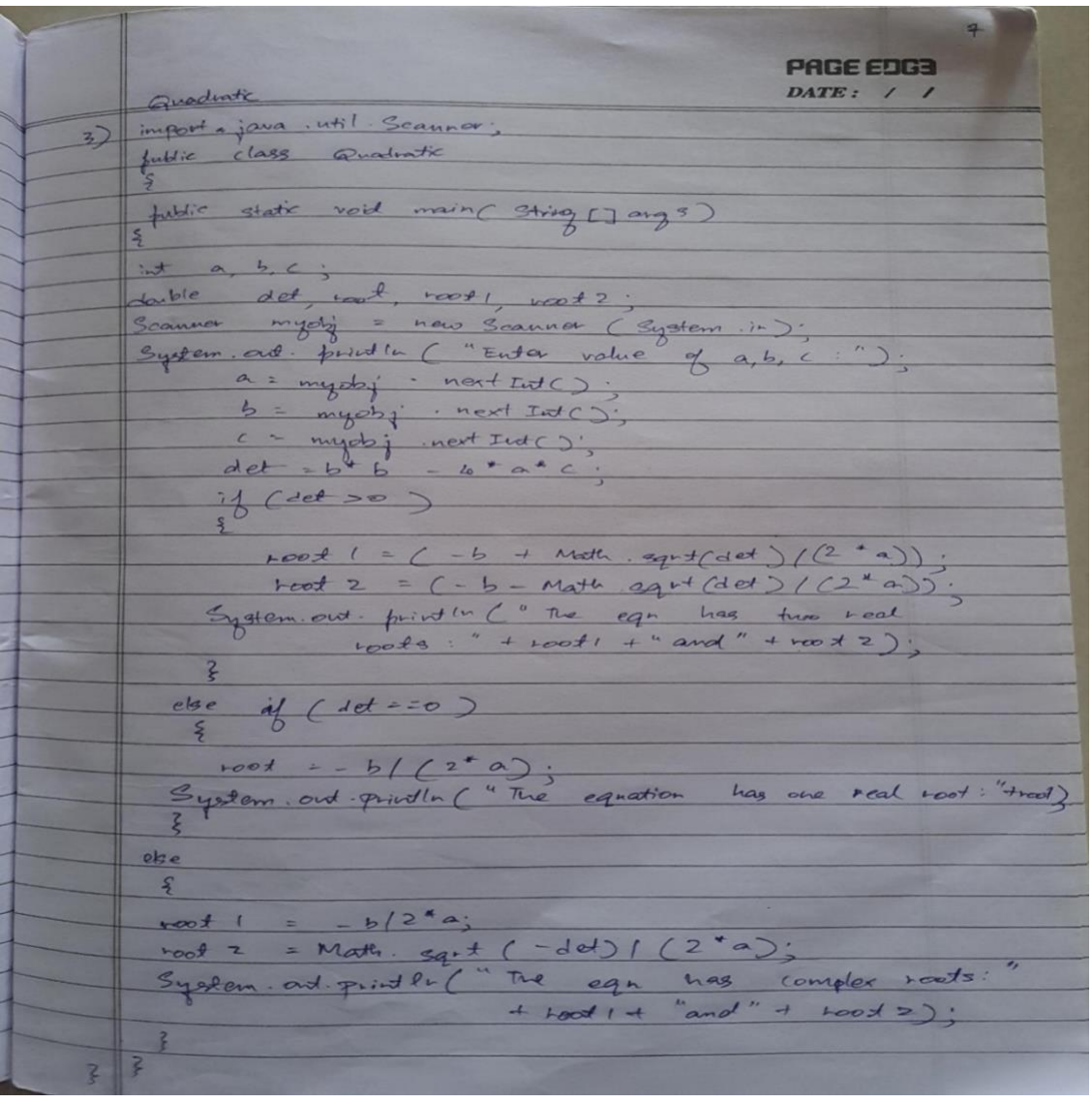
This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Bhoomi Udedh (1BM23CS066),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** 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.

| | |
|---|---|
| Lab faculty Incharge Name<br>Assistant Professor<br>Department of CSE, BMSCE | Dr. Jyothi S Nayak<br>Professor & HOD<br>Department of CSE, BMSCE |

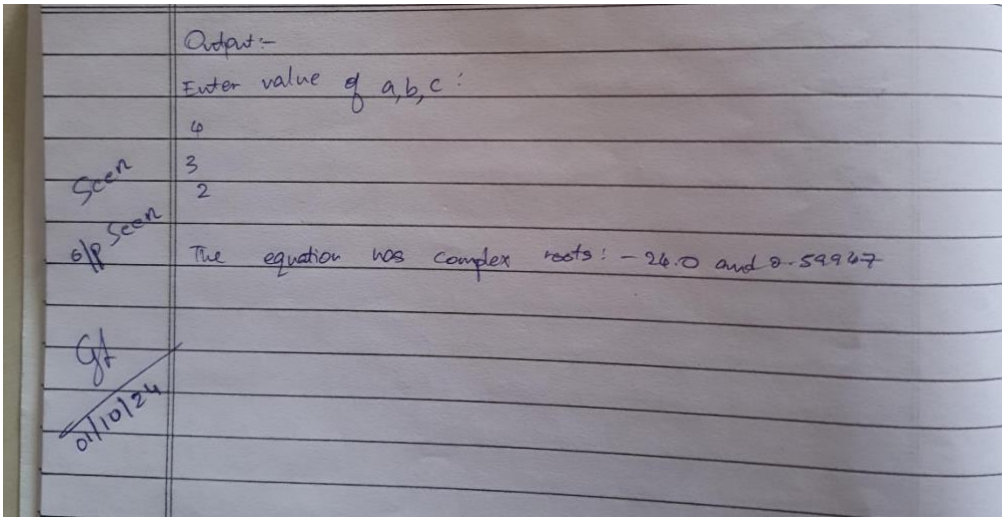
# Index

Github Link:
https://github.com/bhoomiudedh/ooj

**Program 1**
Implement Quadratic Equation
Algorithm



```
PAGE EDGE
DATE : / /

3) Quadratic
import java.util.Scanner;
public class Quadratic
{
public static void main(String[] args)
{
int a, b, c;
double det, root, root1, root2;
Scanner myobj = new Scanner(System.in);
System.out.println("Enter value of a,b,c :");
    a = myobj.nextInt();
    b = myobj.nextInt();
    c = myobj.nextInt();
    det = b*b - 4*a*c;
    if (det>0)
    {
        root1 = (-b + Math.sqrt(det)/(2*a));
        root2 = (-b - Math.sqrt(det)/(2*a));
    System.out.println("The eqn has two real
            roots: " + root1 + "and" + root2);
    }
else if (det==0)
{
    root = -b/(2*a);
System.out.println("The equation has one real root: "+root);
}
else
{
root1 = -b/2*a;
root2 = Math.sqrt(-det)/(2*a);
System.out.println("The eqn has complex roots:"
                + root1 + "and" + root2);
}
}
}
```




Code:

```
import java.util.Scanner;

public class Main
{
   public static void main(String[] args) {
   int a,b,c;
   double det,root,root1,root2;
Scanner myobj=new Scanner (System.in);

       System.out.print("Enter value of a,b,c: ");
       a = myobj.nextInt();
       b = myobj.nextInt();
       c = myobj.nextInt();

     det = b * b - 4.0* a * c;
       if (det > 0)
       {
          root1 = (-b + Math.sqrt(det)) / (2 * a);
          root2 = (-b - Math.sqrt(det)) / (2 * a);
          System.out.println("the equation has two real roots: " + root1 + " and " + root2);
       }
       else if (det == 0)
       {
          root = -b / (2.0 * a);
          System.out.println("the equation has one real root: " + root);
       }
       else
       {
          root1=-b/(2.0*a);
```

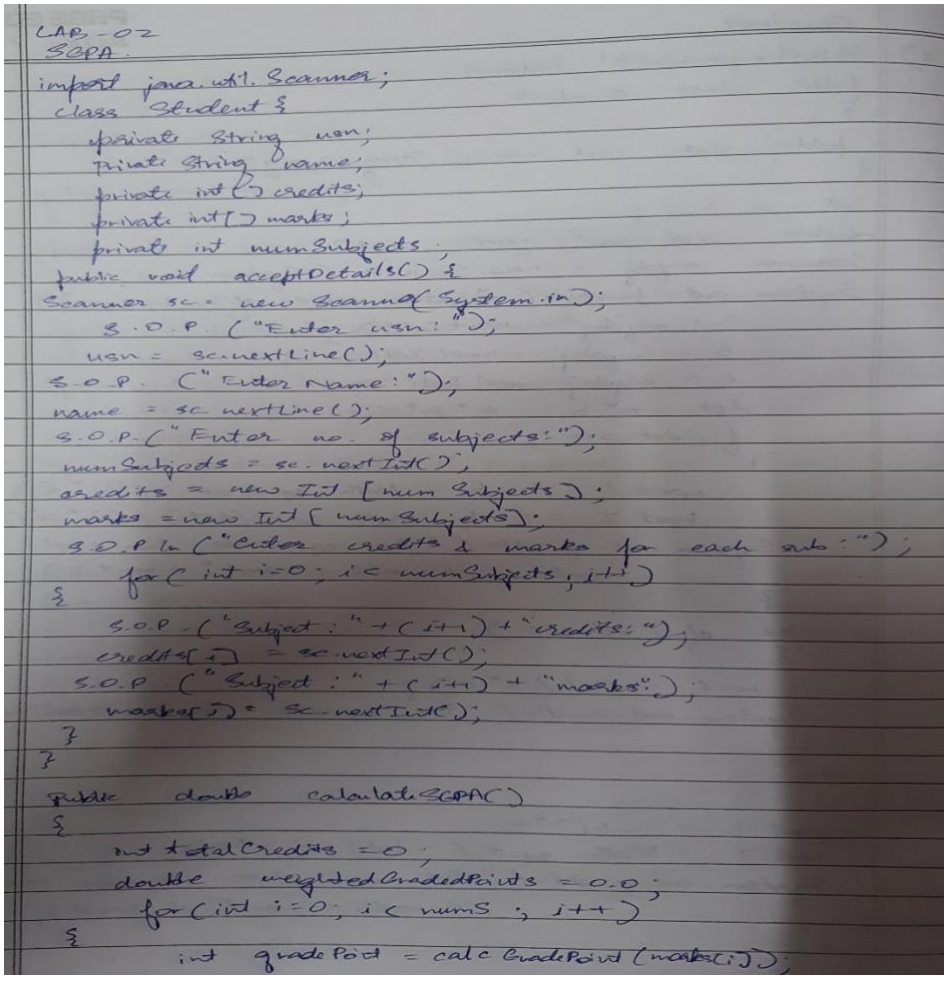
```
        root2=Math.sqrt(-det)/(2.0*a);
        System.out.println("the equation has complex roots:"+root1+"and"+root2);
      }
    }
}
```

Output

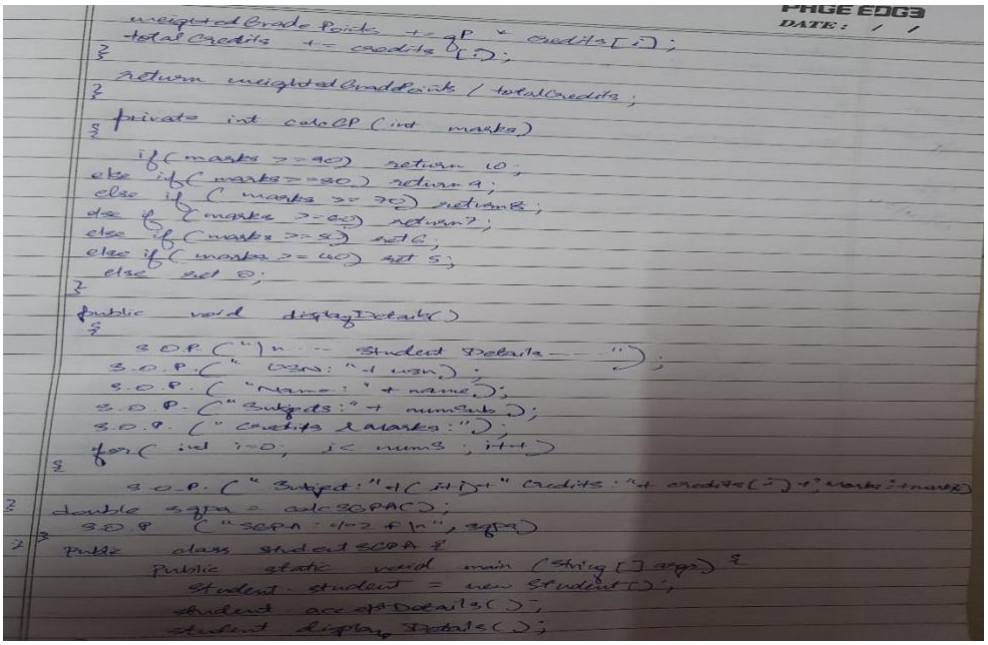
```
Enter value of a,b,c: 4 3 2
the equation has complex roots:-0.375and0.5994789404140899
```

Program 2
SGPA of student

Algorithm



```
LAB - 02
SGPA
import java.util.Scanner;
class Student {
    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;
    private int numSubjects;
public void acceptDetails() {
Scanner sc = new Scanner(System.in);
    S.O.P ("Enter usn: ");
    usn = sc.nextLine();
S.O.P ("Enter Name:");
name = sc.nextLine();
S.O.P("Enter no. of subjects:");
numSubjects = sc.nextInt();
credits = new Int [numSubjects];
marks = new Int [numSubjects];
S.O.P ln ("Enter credits & marks for each sub:");
    for(int i=0; i< numSubjects; i++)
{
    S.O.P ("Subject :" + (i+1) + "credits:");
    credits[i] = sc.nextInt();
S.O.P ("Subject :" + (i+1) + "marks:");
    marks[i] = sc.nextInt();
}
}
public double calculateSGPA()
{
    int totalCredits = 0;
    double weightedGradePoints = 0.0;
    for(int i=0; i< numS ; i++)
{
        int gradePoint = calcGradePoint(marks[i]);
```



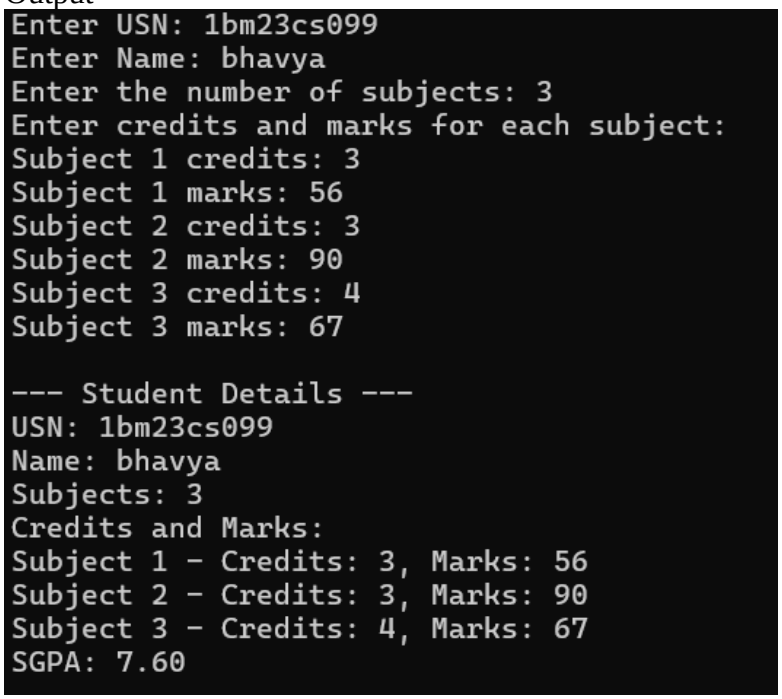
Code

```
import java.util.Scanner;

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

    public void acceptDetails() {
        Scanner sc = new Scanner(System.in);

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

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

```
    System.out.print("Enter the number of subjects: ");
    numSubjects = sc.nextInt();

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

        System.out.print("Subject " + (i + 1) + " marks: ");
        marks[i] = sc.nextInt();
    }
}
public double calculateSGPA() {
    int totalCredits = 0;
    double weightedGradePoints = 0.0;

    for (int i = 0; i < numSubjects; i++) {
        int gradePoint = calculateGradePoint(marks[i]); // Calculate grade point based on marks
        weightedGradePoints += gradePoint * credits[i];
        totalCredits += credits[i];
    }

    return weightedGradePoints / totalCredits;
}
private int calculateGradePoint(int marks) {
    if (marks >= 90) return 10;
    else if (marks >= 80) return 9;
    else if (marks >= 70) return 8;
    else if (marks >= 60) return 7;
    else if (marks >= 50) return 6;
    else if (marks >= 40) return 5;
    else return 0; // Fail
}
public void displayDetails() {
    System.out.println("\n--- Student Details ---");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Subjects: " + numSubjects);

    System.out.println("Credits and Marks:");
    for (int i = 0; i < numSubjects; i++) {
        System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);
    }
```

```
        double sgpa = calculateSGPA();
        System.out.printf("SGPA: %.2f\n", sgpa);
    }
}
public class StudentSGPA {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
    }
}
```

Output



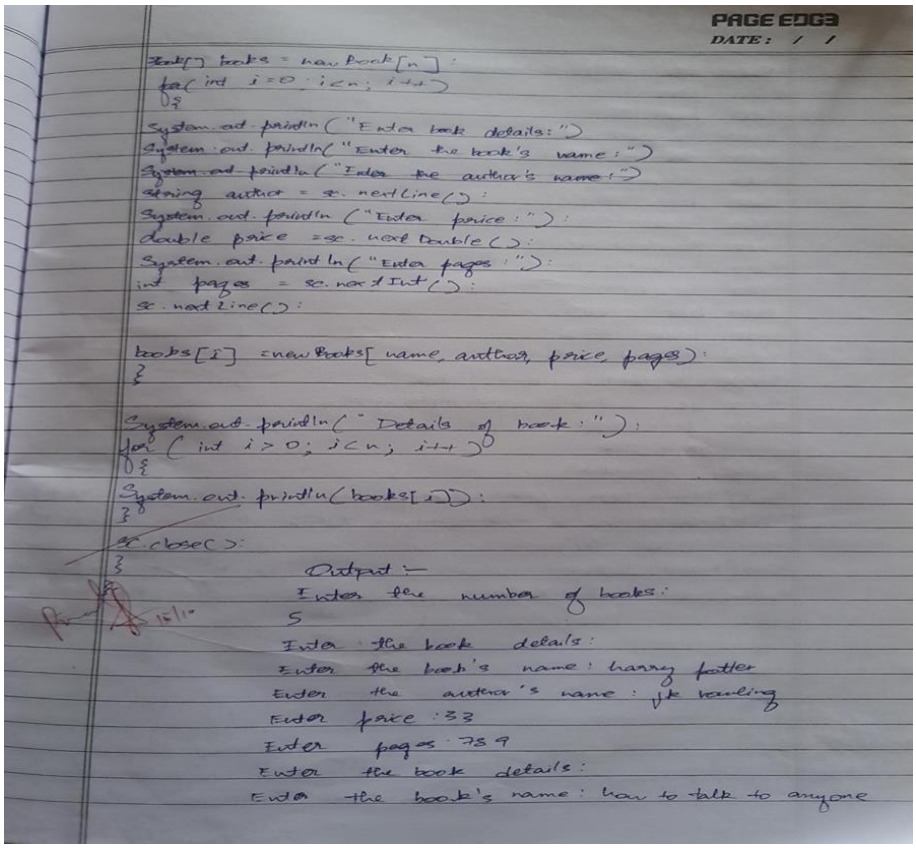

Program 3
Book information

Algorithm

15/10/2024. LAB - 3

PAGE EDGE
DATE : / /

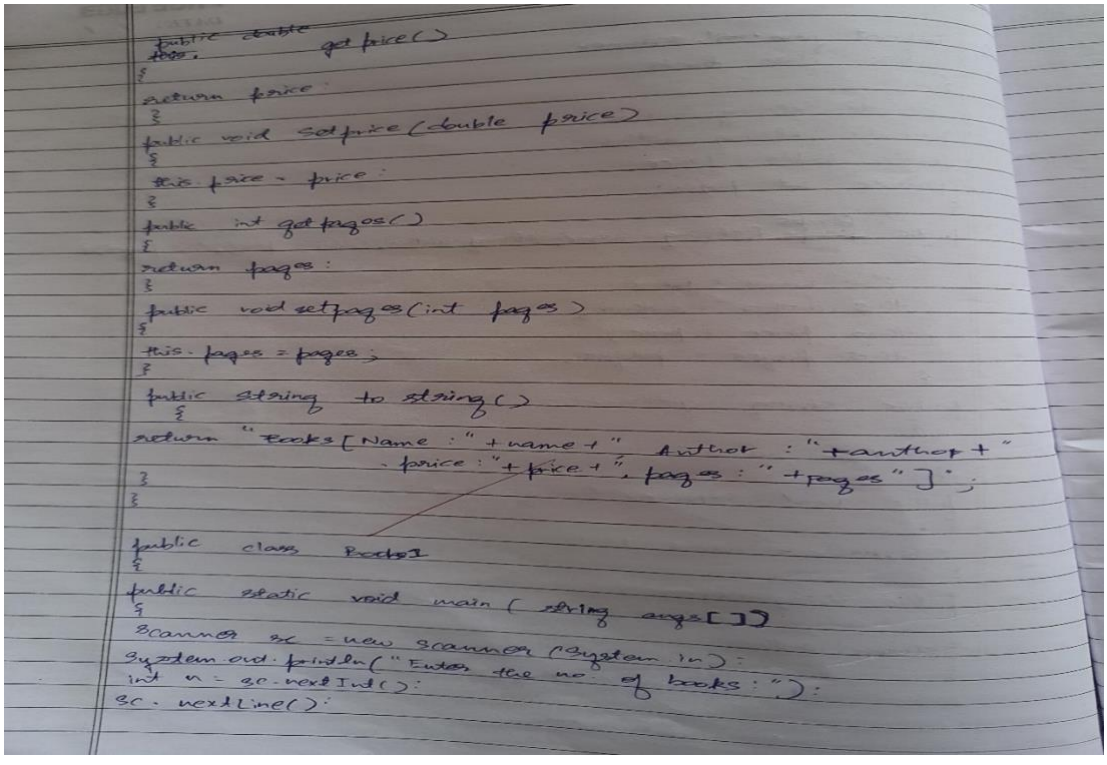
① import java.util.Scanner
public class Books
{
private string name;
private string author;
private double price;
private int pages;
}
public Books(String name, String author, double price, int pages)
{
this.name = name;
this.author = author;
this.price = price;
this.pages = pages;
}
public string getName()
{
return name;
}
public void setName(String name)
{
this.name = name;
}
public string author()
{
return author;
}
public void setAuthor (String author)
{
this.author = author;
}



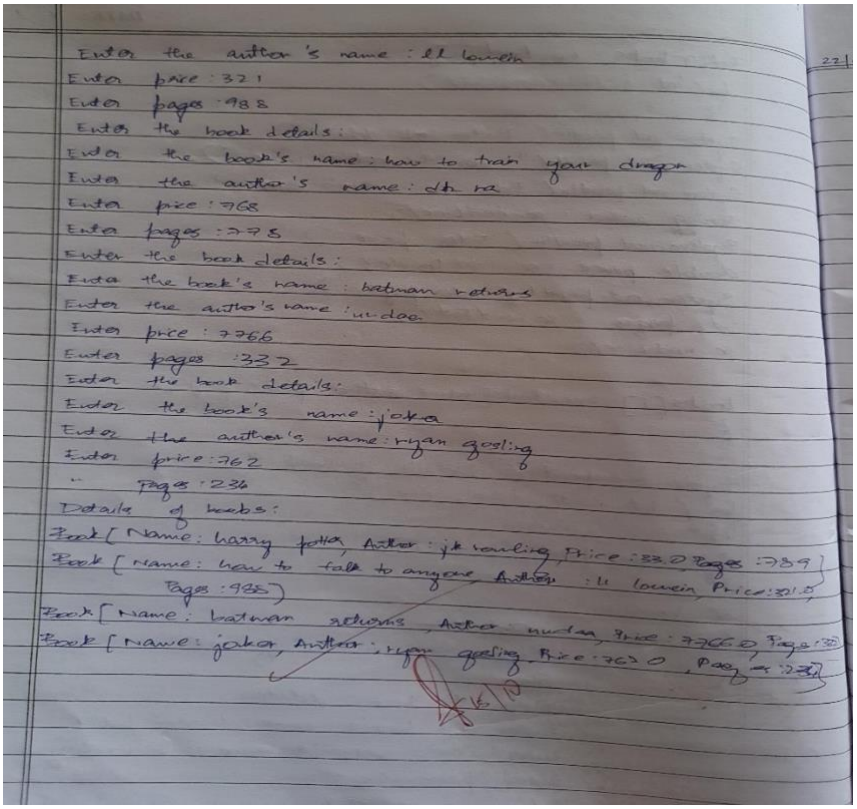
```
Book[] books = new Book[n];
for(int i=0; i<n; i++)
{
System.out.println("Enter book details:");
System.out.println("Enter the book's name:");
String author = sc.nextLine();
System.out.println("Enter price:");
double price = sc.nextDouble();
System.out.println("Enter pages:");
int pages = sc.nextInt();
sc.nextLine();

books[i] = new Books[name, author, price, pages);
}

System.out.println("Details of book:");
for(int i>0; i<n; i++)
{
System.out.println(books[i]);
}
sc.close();
}
```

Output :-

```
Enter the number of books:
5
Enter the book details:
Enter the book's name: harry potter
Enter the author's name: jk rowling
Enter price: 33
Enter pages: 789
Enter the book details:
Enter the book's name: how to talk to anyone
```



```
public double getPrice()
{
return price;
}
public void setPrice(double price)
{
this.price = price;
}
public int getPages()
{
return pages;
}
public void setPages(int pages)
{
this.pages = pages;
}
public String toString()
{
return "Books[Name :" + name + " Author :" + author +
" price :" + price + ", pages :" + pages + "]";
}
}

public class Books1
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the no. of books:");
int n = sc.nextInt();
sc.nextLine();
```




Code

```
import java.util.Scanner;

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


    public Book(String name, String author, double price, int pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.pages = 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 getPages() {
        return pages;
    }

    public void setPages(int pages) {
        this.pages = pages;
    }

    public String toString() {
        return "Book [Name: " + name + ", Author: " + author + ", Price: " + price + ", Pages: " + pages
+ "]";
    }



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

        Book[] books = new Book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter the book details:");
            System.out.print("Enter the book's name: ");
            String name = sc.nextLine();
            System.out.print("Enter the author's name: ");
            String author = sc.nextLine();
```

```
            System.out.print("Enter price: ");
            double price = sc.nextDouble();
            System.out.print("Enter pages: ");
            int pages = sc.nextInt();
            sc.nextLine(); // Consume the newline character

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

        System.out.println("Details of books:");
        for (int i = 0; i < n; i++) {
            System.out.println(books[i]);
        }
        sc.close();
    }
}
```

Output

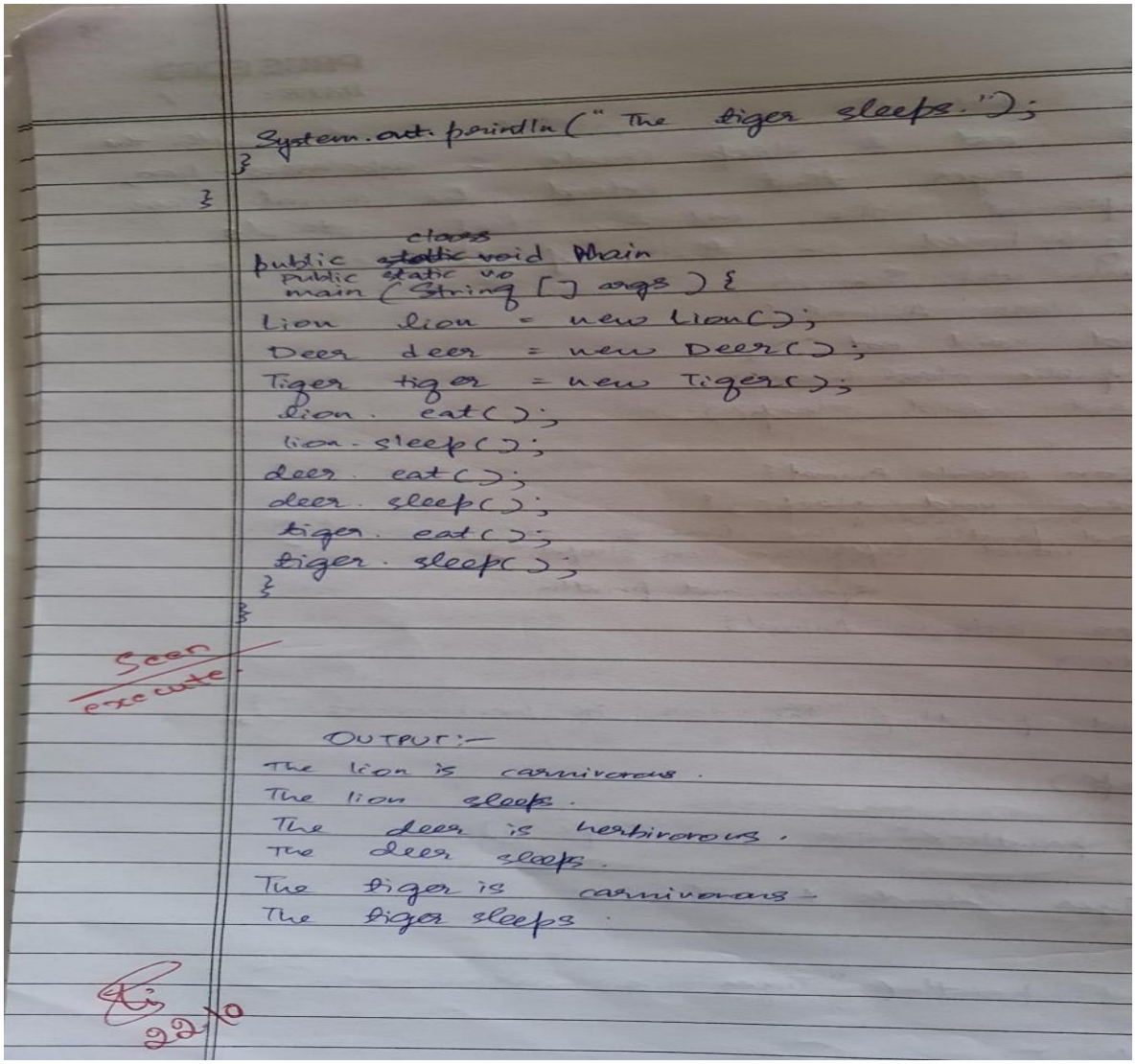
```
Enter the number of books:
5
Enter the book details:
Enter the book's name: harry potter
Enter the author's name: jk rowling
Enter price: 33
Enter pages: 789
Enter the book details:
Enter the book's name: how to talk to anyone
Enter the author's name: ll lowein
Enter price: 321
Enter pages: 988
Enter the book details:
Enter the book's name: how to train your dragon
Enter the author's name: dkra
Enter price: 768
Enter pages: 778
Enter the book details:
Enter the book's name: batman returns
Enter the author's name: uudae
Enter price: 7766
Enter pages: 332
Enter the book details:
Enter the book's name: joker
Enter the author's name: ryan gosling
Enter price: 762
Enter pages: 234
Details of books:
Book [Name: harry potter, Author: jk rowling, Price: 33.0, Pages: 789]
Book [Name: how to talk to anyone, Author: ll lowein, Price: 321.0, Pages: 988]
Book [Name: how to train your dragon, Author: dkra, Price: 768.0, Pages: 778]
Book [Name: batman returns, Author: uudae, Price: 7766.0, Pages: 332]
Book [Name: joker, Author: ryan gosling, Price: 762.0, Pages: 234]
```

Program 4
Abstract classes-Animal and Shape

22/10/2024 LAB - 04 DATE : / /

1. Create an abstract class called Animal with the methods eat & sleep. Create 3 subclasses - lion, deer & tiger that extend the animal class & implement the eat & sleep methods differently based on their specific behaviour.

```
import java.util.Scanner;
abstract class Animal {
    abstract void eat();
        public class lion()
    abstract void sleep();
}
class lion extends Animal {
        public void sleep() {
{ System.out.println("The lion is carnivorous.");
        void eat();
        void sleep("The lion sleeps.");
    }
}
                System.out.println
class Deer extends Animal {
void eat()
{
System.out.println("The deer is herbivorous.");
}
 void sleep()
 {
System.out.println("The deer sleeps.");
}
class Tiger extends Animal {
void eat()
{
System.out.println("Tiger is carnivorous.");
}
void sleep()
{
```

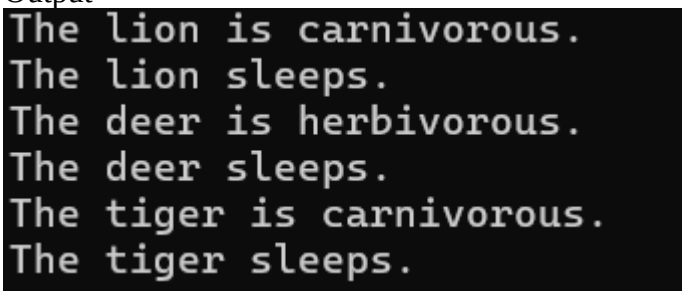



Code

```
abstract class Animal{
abstract void eat();
abstract void sleep();
}
class Lion extends Animal{
void eat()
{
System.out.println("The lion is carnivorous.");
}
void sleep()
{
System.out.println("The lion sleeps.");
}
```

```
 }
class Deer extends Animal{
void eat()
{
System.out.println("The deer is herbivorous.");
}
void sleep()
{
System.out.println("The deer sleeps.");
}
}
class Tiger extends Animal{
void eat()
{
System.out.println("The tiger is carnivorous.");
}
void sleep()
{
System.out.println("The tiger sleeps.");
}
}
public class Main{
public static void main(String[]args){
Lion lion=new Lion();
Deer deer=new Deer();
Tiger tiger=new Tiger();
lion.eat();
lion.sleep();
deer.eat();
deer.sleep();
tiger.eat();
tiger.sleep();
}
}
```

Output




Abstract class Shapes implementation

Algorithm

Program 4 :

Develop a Java program to create an abstract class named Shape that contains two integers & an empty method named printArea(). Provide three classes named Rectangle, Triangle & 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.

```
import java.util.*;
abstract class Shape{
  double x,y;
  Shape(double x, double y){
  this.x = x;
  this.y = y;
}
  abstract void printArea();
}
class Rectangle extends Shape{
  Rectangle(double l, double b){
 super(l,b);
}
  void printArea(){
  double a = x*y;
  System.out.println("Area of rectangle is:"+a+"m^2");
}
}
class Triangle extends Shape{
  Triangle(double b, double h){
  super(b,h);
}
  void printArea(){
  double at = 0.5*x*y;
```

```
        System.out.println("Area of triangle is:"+at+"m^2");
    }
}
class Circle extends Shape{
    Circle(double r, double e){
    super(r,0);
}
    void printArea(){
    double ac = 3.14 * r * r;
    System.out.println("Area of circle is:"+ac+"m^2");
    System.out.println("Bhoomi Wadekh : 1BM23CS066");
    }

public class Main {
    public static void main(String[] Args){
    Scanner sc = new Scanner(System.in);
    System.out.println("enter sides of rectangle:");
    double l = sc.nextInt();
    double b = sc.nextInt();
    Rectangle rect = new Rectangle(l, b);
    rect.printArea();
    System.out.println("Enter sides of triangle:");
    double B = sc.nextDouble();
    double h = sc.nextDouble();
    Triangle tr = new Triangle(B, h);
    tr.printArea();
    System.out.println("enter radius of circle:");
    double r = sc.nextDouble();
    Circle c = new Circle(r,0);
    c.printArea();
}
```

Output :-

```
enter sides of rectangle :
2
4
Area of rectangle is : 8.0 m^2
enter sides of triangle :
4 3
2 2
Area of triangle is : 473.0 m^2
enter radius of circle :
4 8
Area of circle is : 7234.5599 m^2
Bhoomi Wadekh : 1BM23CS066
```

Code

```
import java.util.*;
abstract class Shape{
  double x,y;
  Shape(double x,double y){
  this.x=x;
  this.y=y;
}
  abstract void printArea();
}
class Rectangle extends Shape{
  Rectangle(double l,double b){
  super(l,b);
}
  void printArea(){
  double a=x*y;
  System.out.println("Area of rectangle is:"+ a+ " m^2");
  }
}
class Triangle extends Shape{
  Triangle(double b,double h){
  super(b,h);
}
  void printArea(){
  double at=0.5*x*y;
  System.out.println("Area of triangle is:"+ at+ " m^2");
  }
}
class Circle extends Shape{
  Circle(double r,double e){
  super(r,0);
}
  void printArea(){
  double ac=3.14*x*x;
  System.out.println("Area of circle is:"+ ac+ "  m^2");
  System.out.println("Bhoomi Udedh:1BM23CS066");
  }
}
public class Main{
  public static void main(String[] Args){
    Scanner sc=new Scanner(System.in);
    System.out.println("enter sides of rectangle:");
    int l=sc.nextInt();
    int b=sc.nextInt();
    Rectangle rect=new Rectangle(l,b);
    rect.printArea();
    System.out.println("enter sides of triangle:");
```

```
    double B=sc.nextDouble();
    double h=sc.nextDouble();
    Triangle tr=new Triangle(B,h);
    tr.printArea();
    System.out.println("enter radius of circle:");
    double r=sc.nextDouble();
    Circle c=new Circle(r,0);
    c.printArea();
}
}
```

Output

```
enter sides of rectangle:
2
4
Area of rectangle is:8.0 m^2
enter sides of triangle:
43
22
Area of triangle is:473.0 m^2
enter radius of circle:
48
Area of circle is:7234.5599999999995  m^2
Bhoomi Udedh:1BM23CS066
```

Program 5
Bank class

Algorithm

a) Accept deposit from customer & update the balance
b) Display balance
c) Compute & deposit interest
d) Permit withdrawal & update balance ~~Check for min Balance, impose penalty if necessary~~

29/10/2024 LAB - 05.

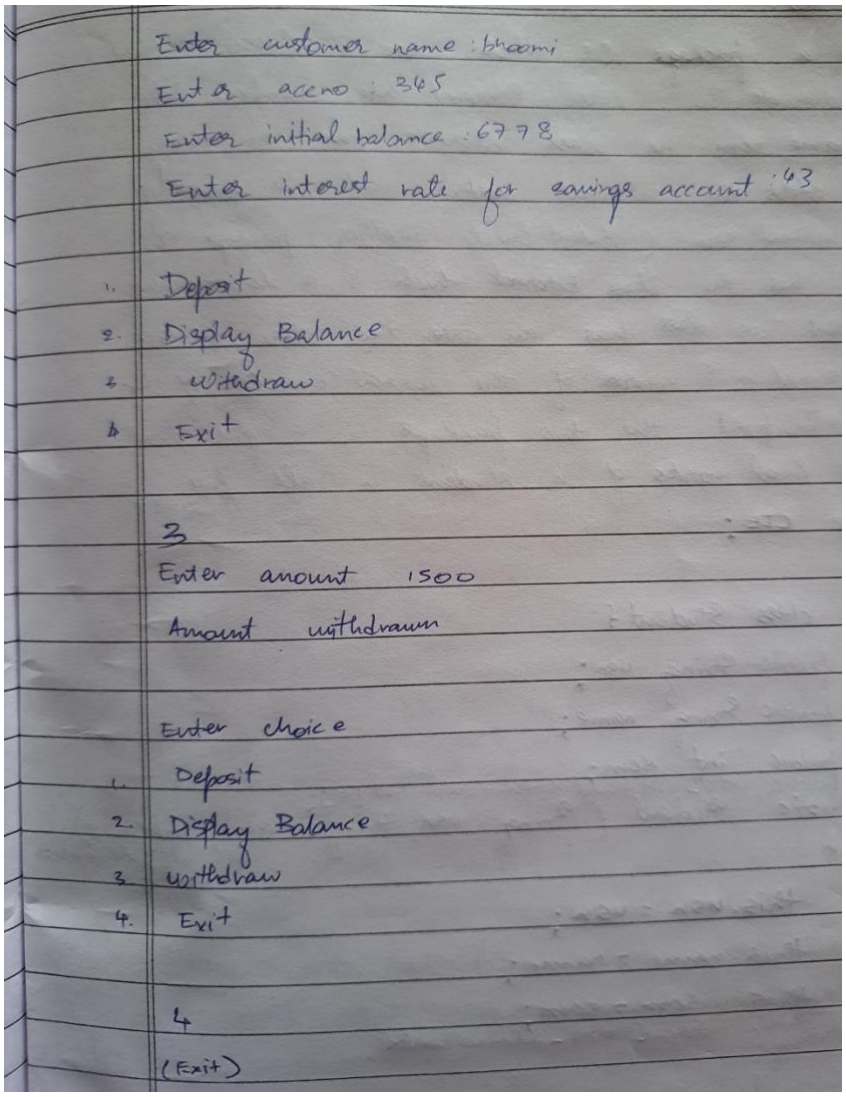
Develop a Java program to create class Bank that maintains 2 kinds of acc: for its customers, one called savings acc. & other current Acc. The Savings Acc. provides compound interest & withdrawal facilities but no cheque book facility. The current acc. provides cheque book facility but no interest. Current acc. holders should also maintain min. balance & if the balance falls below this level, service charge is imposed.
Create a class Account that stores customer name, acc. no, & type of account. From this derive the classes Cur-acct & Sav-acct to make them more specific to their requirements. Include necessary methods in order to achieve the following tasks.

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

public Account (String customerName, int accountNumber,
String accountType, double balance)
{
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;
}
    public void acceptDeposit (double amount)
{
    balance += amount;
System.out.println ("Deposit successful.
                    New balance : " + balance");
}
    public void displayBalance ()
{
  System.out.println ("Account Balance : " + balance);
}
```

```
public void withdraw (double amount)
{
if (balance >= amount)
{
    balance -= amount;
System.out.println("Withdrawal successful. New balance : " + balance);
}
else
{
  System.out.println ("Insufficient balance.");
}
}
Derived subclass for savings account
public class SavAcct extends Account {
    private final double interestRate = 0.05;

    public SavAcct (String customerName, int accountNumber, double balance)
{
super (customerName, int accountNumber, double balance) {
super (customerName, accountNumber, "Savings", balance);
}
public void compoundInterest () {
double interest = balance * interestRate;
    balance += interest;
System.out.println("Interest deposited.
    New balance : " + balance);
}
}

class Current extends Account
{
```

```
class Curacct .
private double minBalance = 500;
private double penalty = 50;

public Curacct ( String customerName, int accountNumber,
        double balance, String acctype double minbalance
                double penaltyamt)
{
  super ( customerName, accountNumber, acctype, balance);
}
```

```
    this.minbalance = minbalance;
    this.penaltyamt = penaltyamt;
}

public void withdraw(double amount) {
        if (balance >= amount)
            balance -= amount;
    System.out.println("Amount withdrawal");
        if (balance < minbalance) {
            balance -= penaltyamt;
    System.out.println("Penalty of :" + penaltyamt);
}
}
else
    System.out.println("Insufficient balance");
}

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

System.out.println("Enter balance:");
double balance = sc.nextbalance();

System.out.println("Enter account type:");
String acctype = sc.next();
```



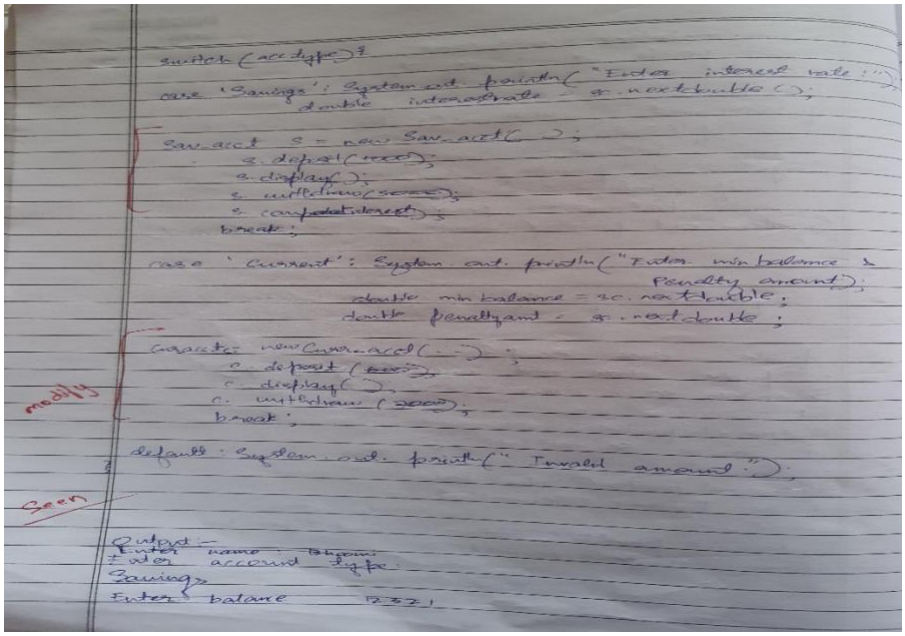
Enter customer name : bhoomi
Enter accno : 345
Enter initial balance : 6778
Enter interest rate for savings account : 43

1. Deposit
2. Display Balance
3. Withdraw
4. Exit

3
Enter amount 1500
Amount withdrawn

Enter choice
1. Deposit
2. Display Balance
3. Withdraw
4. Exit

4
(Exit)



switch (acc type)?
case 'Savings': System.out.println("Enter interest rate :");
double interestRate = sc.nextDouble();
Sav_acct s = new Sav_acct( );
s.deposit(recc);
s.display( );
s.withdraw(scan);
s.computeInterest( );
break;
case 'Current': System.out.println("Enter min balance & Penalty amount");
double minbalance = sc.nextDouble;
double penaltyamt = sc.nextdouble;
Current c = new Curr_acct( );
c.deposit(scan);
c.display( );
c.withdraw(scan);
break;
default: System.out.println("Invalid amount");

modify

Seen

Output :-
Enter name bhoomi
Enter account type
Savings
Enter balance 7777

Code

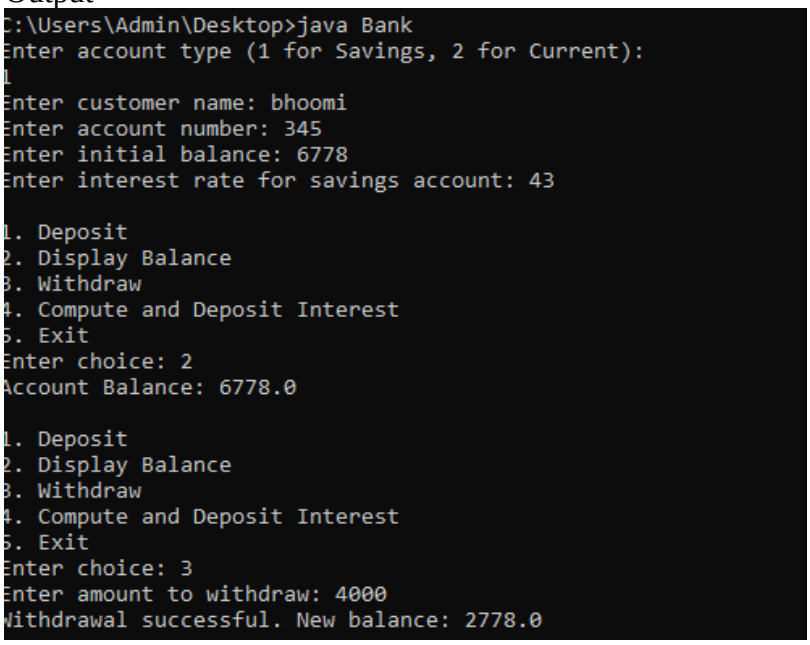
```
import java.util.Scanner;

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

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

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println("Deposit successful. New balance: " + balance);
      } else {
         System.out.println("Deposit amount must be positive.");
      }
   }

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

   public void withdraw(double amount) {
      if (amount > 0 && amount <= balance) {
         balance -= amount;
         System.out.println("Withdrawal successful. New balance: " + balance);
      } else {
         System.out.println("Invalid withdrawal amount or insufficient balance.");
      }
   }
}

class SavAcct extends Account {
   private double interestRate;

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

```
    public void computeAndDepositInterest(int years) {
        double interest = balance * Math.pow(1 + interestRate / 100, years) - balance;
        balance += interest;
        System.out.println("Interest deposited. New balance: " + balance);
    }


    public void withdraw(double amount) {
        super.withdraw(amount);
    }
}

class CurAcct extends Account {
    private double minimumBalance;
    private double serviceCharge;

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


    public void withdraw(double amount) {
        if (balance - amount < minimumBalance) {
            System.out.println("Balance would fall below minimum. Service charge of " + serviceCharge
+ " will be applied.");
            balance -= (amount + serviceCharge);
        } else {
            balance -= amount;
        }
        System.out.println("Withdrawal successful. New balance: " + balance);
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter account type (1 for Savings, 2 for Current): ");
        int accountType = sc.nextInt();
        sc.nextLine();

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

        System.out.print("Enter account number: ");
```

```
        String accountNumber = sc.nextLine();

        System.out.print("Enter initial balance: ");
        double initialBalance = sc.nextDouble();

        Account account;
        if (accountType == 1) {
            System.out.print("Enter interest rate for savings account: ");
            double interestRate = sc.nextDouble();
            account = new SavAcct(customerName, accountNumber, initialBalance, interestRate);
        } else {
            System.out.print("Enter minimum balance for current account: ");
            double minimumBalance = sc.nextDouble();

            System.out.print("Enter service charge for current account: ");
            double serviceCharge = sc.nextDouble();

            account = new CurAcct(customerName, accountNumber, initialBalance, minimumBalance,
serviceCharge);
        }

        int choice;
        do {
            System.out.println("\n1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Withdraw");
            if (account instanceof SavAcct) {
                System.out.println("4. Compute and Deposit Interest");
            }
            System.out.println("5. Exit");
            System.out.print("Enter choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = sc.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    account.displayBalance();
                    break;
                case 3:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = sc.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
```

```
            case 4:
                if (account instanceof SavAcct) {
                    System.out.print("Enter number of years to calculate interest: ");
                    int years = sc.nextInt();
                    ((SavAcct) account).computeAndDepositInterest(years);
                } else {
                    System.out.println("Invalid choice for current account.");
                }
                break;
            case 5:
                System.out.println("Thank you for banking with us.");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 5);
  }
}
```

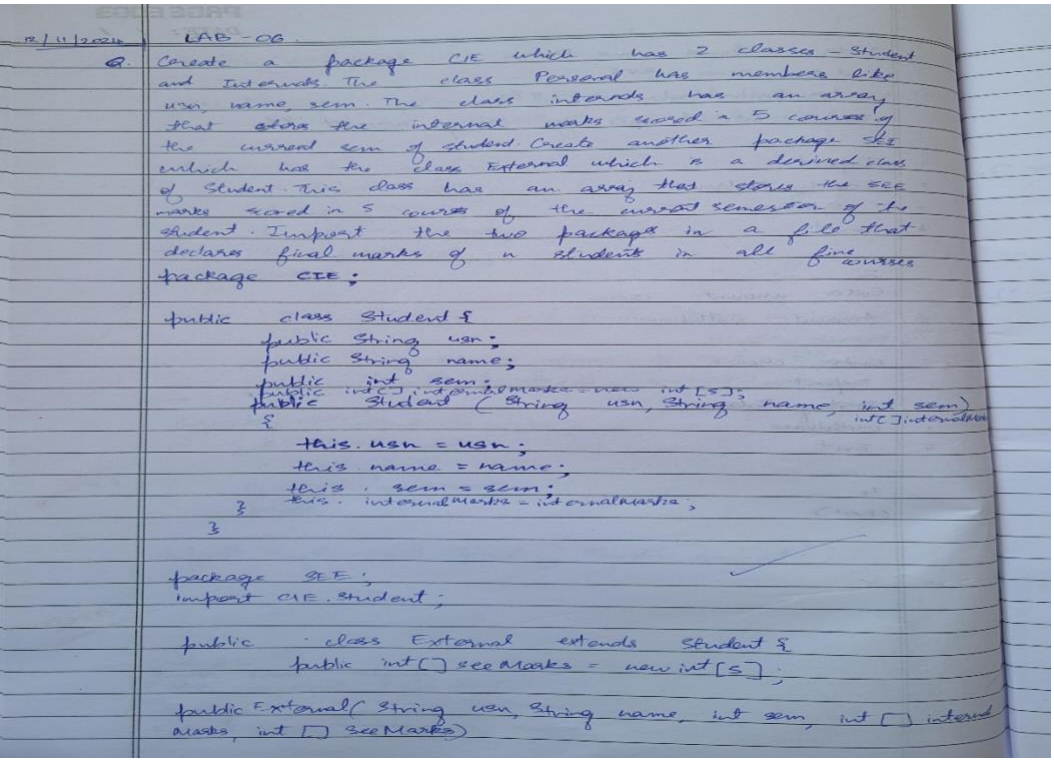
Output



```
1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 4
Enter number of years to calculate interest: 3
Interest deposited. New balance: 8123.447045999998

1. Deposit
2. Display Balance
3. Withdraw
4. Compute and Deposit Interest
5. Exit
Enter choice: 5
Thank you for banking with us.

C:\Users\Admin\Desktop>
```

Program 6
Implementation of packages

Algorithm

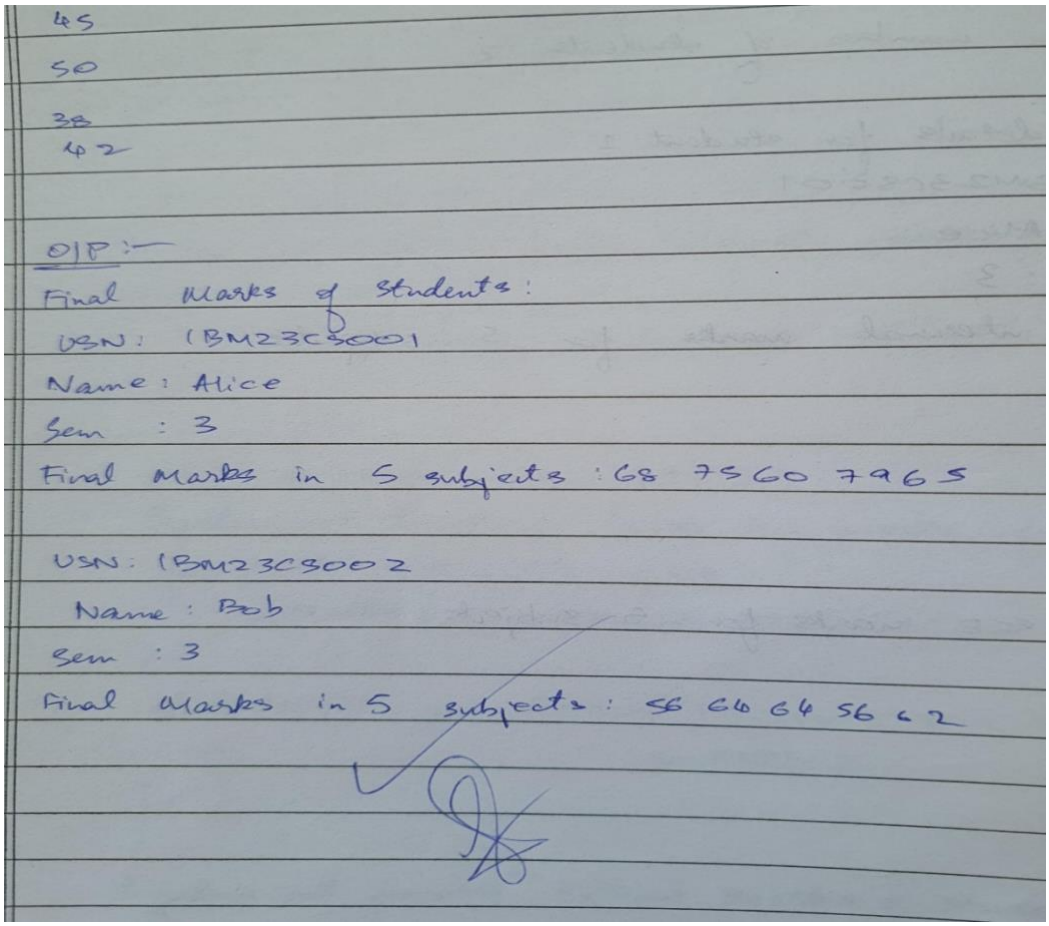


DATE : / /

```
{
    super(usn, name, sem, internalMarks);
    this.seeMarks = seeMarks;
}

public int[] getFinalMarks() {
    int[] finalMarks = new int[5];
    for(int i=0; i<5; i++)
    {
        finalMarks[i] = internalMarks[i] + seeMarks[i];
    }
    return finalMarks;
}
}

import CIE.Student;
import SEE.External;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter no. of students:");
    int n = scanner.nextInt();

    External[] students = new External[n];

    for(int i=0; i<n; i++) {
        System.out.println("\n Enter details for student "+(i+1));
        System.out.print("USN:");
        String usn = scanner.next();
        System.out.print("Name:");
```

```
String name = scanner.next();
System.out.print("Semester: ");
int sem = scanner.nextInt();

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

int[] seeMarks = new int[5];
System.out.println("Enter SEE marks for 5 subjects:");
for(int j=0; j<5; j++) {
    seeMarks[j] = scanner.nextInt();
}
students[i] = new External(usn, name, sem, internalMarks,
                           seeMarks);
}

System.out.println("\nFinal marks of students:");
for(External student : students) {
    System.out.println("USN: "+student.usn);
    System.out.println("Name: "+student.name);
    System.out.println("Semester: "+student.sem);
    int[] finalMarks = student.getFinalMarks();
    System.out.print("Final marks in 5 subjects:");
    for(int mark : finalMarks) {
        System.out.print(mark+" ");
    }
    System.out.println("\n");
}
scanner.close();
}
}
```

```
O/P:-
Enter number of students: 2

Enter details for student 1
USN: 1BM23CS001
Name: Alice
Semester: 3
Enter internal marks for 5 subjects:
18
20
15
19
17
Enter SEE marks for 5 subjects:
50
55
45
60
48

Enter details for student 2
USN: 1BM23CS002
Name: Bob
Semester: 3
Enter internal marks for 5 subjects:
16
19
14
18
20
Enter SEE marks for 5 subjects:
40
```




Code

```
//code done in separate folder(CIE)
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;
    public int[] internalMarks = new int[5];

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

```
//code done in a separate folder(SEE)

package SEE;
import CIE.Student;

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

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

    public int[] getFinalMarks() {
        int[] finalMarks = new int[5];
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + seeMarks[i];
        }
        return finalMarks;
    }
}


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

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

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

        External[] students = new External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1));

            System.out.print("USN: ");
            String usn = scanner.next();

            System.out.print("Name: ");
            String name = scanner.next();

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

```
            int sem = scanner.nextInt();

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

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

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

        System.out.println("\nFinal Marks of Students:");
        for (External student : students) {
            System.out.println("USN: " + student.usn);
            System.out.println("Name: " + student.name);
            System.out.println("Semester: " + student.sem);
            int[] finalMarks = student.getFinalMarks();
            System.out.print("Final Marks in 5 subjects: ");
            for (int mark : finalMarks) {
                System.out.print(mark + " ");
            }
            System.out.println("\n");
        }
    }
}
```

Output

```
Enter number of students: 2

Enter details for student 1
USN: 1BM23CS001
Name: Alice
Semester: 3
Enter internal marks for 5 subjects:
18
20
15
19
17
Enter SEE marks for 5 subjects:
50
55
45
60
48

Enter details for student 2
USN: 1BM23CS002
Name: Bob
Semester: 3
Enter internal marks for 5 subjects:
16
19
14
18
20
Enter SEE marks for 5 subjects:
40
45
50
38
42

Final Marks of Students:
USN: 1BM23CS001
Name: Alice
Semester: 3
Final Marks in 5 subjects: 68 75 60 79 65

USN: 1BM23CS002
Name: Bob
Semester: 3
Final Marks in 5 subjects: 56 64 64 56 62
```

Program 7
Interface

Algorithm

4) class Document

output :-

Printing document

Showing document preview

5) We have created an interface named Polygon. It includes a default method getPerimeter() and an abstract method getArea().

We can calculate the perimeter of all polygons in the same manner so we implemented the body of getPerimeter() in Polygon.

Now, all polygons that implement Polygon can use getPerimeter() to calculate perimeter.

However, the rule for calculating the area is different for different polygons.

Hence, getArea() is included without implementation.

Any class that implements Polygon must

DATE : / /

provide an implementation of getArea().

```
public interface Polygon {
    double getPerimeter(double... sideLengths)
    {
        double perimeter = 0;
        for (double side : sideLengths)
        {
            perimeter += side;
        }
        return perimeter;
    }
    double getArea();
}

class Rectangle implements Polygon
{
    double length;
    double width;
}
    getArea() {
        return length * width;
}

    double calculatePerimeter()
    {
        return getPerimeter(length, width, length, width);
}

class Triangle implements Polygon
{
    double base;
```

DATE : / /

```
Rectangle rectangle = new Rectangle (length, width);
Sys.out.println("Rectangle Area:" + rectangle.getArea());
Sys.out.println("Rectangle Perimeter:" + rectangle.calculateP());
Sys.out.println("\n enter base of triangle:");
double base = scanner.nextDouble();
System.out.println("Enter height of triangle:");
double height = scanner.nextDouble();
System.out.println("enter three sides of triangle:");
double side1 = scanner.nextDouble();
double side2 = scanner.nextDouble();
double side3 = scanner.nextDouble();

Triangle triangle = new Triangle(base, height, side1, s2,
    s3);
    Sys.out.println("Area of D'c : "+ triangle.getArea());
    SOP("D'e Perimeter:" + triangle.calcPerimeter());
}
}
```

O/P :-

```
Enter length of rectangle:
3
Enter width :
2
Area rectangle: 6.0
Rectangle perimeter: 10.0

Enter base of triangle:
4 5
        height
3
        3 sides
1 2 3
4
3 2
```

```
Triangle Area : 67.5
Triangle Perimeter : 159.0
```

Code

```
import java.util.Scanner;
 interface Polygon {
       default double getPerimeter(double... sideLengths) {
       double perimeter = 0;
       for (double side : sideLengths) {
          perimeter += side;
       }
       return perimeter;
   }
   double getArea();
}

 class Rectangle implements Polygon {
   private double length;
   private double width;

   public Rectangle(double length, double width) {
      this.length = length;
      this.width = width;
   }
   public double getArea() {
      return length * width;
   }

   public double calculatePerimeter() {
      return getPerimeter(length, width, length, width);
   }
}
 class Triangle implements Polygon {
   double base;
    double height;
    double side1, side2, side3;

    public Triangle(double base, double height, double side1, double side2, double side3) {
      this.base = base;
      this.height = height;
      this.side1 = side1;
      this.side2 = side2;
      this.side3 = side3;
   }
   public double getArea() {
      return 0.5 * base * height;
   }
```

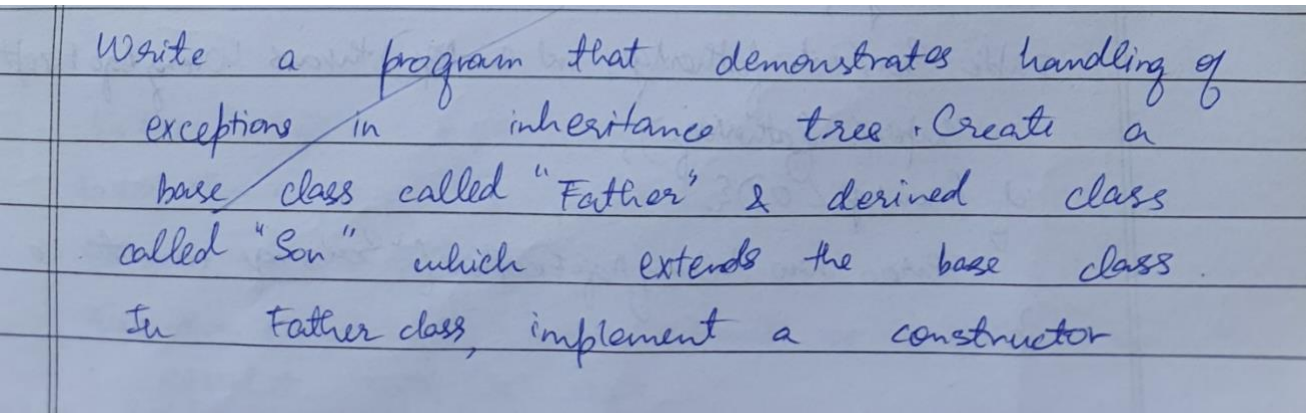
```
    public double calculatePerimeter() {
        return getPerimeter(side1, side2, side3);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the length of the rectangle:");
        double length = scanner.nextDouble();
        System.out.println("Enter the width of the rectangle:");
        double width = scanner.nextDouble();

        Rectangle rectangle = new Rectangle(length, width);
        System.out.println("Rectangle Area: " + rectangle.getArea());
        System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());


        System.out.println("\n Enter the base of the triangle:");
        double base = scanner.nextDouble();
        System.out.println("Enter the height of the triangle:");
        double height = scanner.nextDouble();
        System.out.println("Enter the three sides of the triangle:");
        double side1 = scanner.nextDouble();
        double side2 = scanner.nextDouble();
        double side3 = scanner.nextDouble();

        Triangle triangle = new Triangle(base, height, side1, side2, side3);
        System.out.println("Triangle Area: " + triangle.getArea());
        System.out.println("Triangle Perimeter: " + triangle.calculatePerimeter());
    }
}
```

Output

```
Enter the length of the rectangle:
3
Enter the width of the rectangle:
2
Rectangle Area: 6.0
Rectangle Perimeter: 10.0

 Enter the base of the triangle:
45
Enter the height of the triangle:
3
Enter the three sides of the triangle:
123
4
32
Triangle Area: 67.5
Triangle Perimeter: 159.0

C:\Users\Bhoomi\OneDrive\Desktop>
```

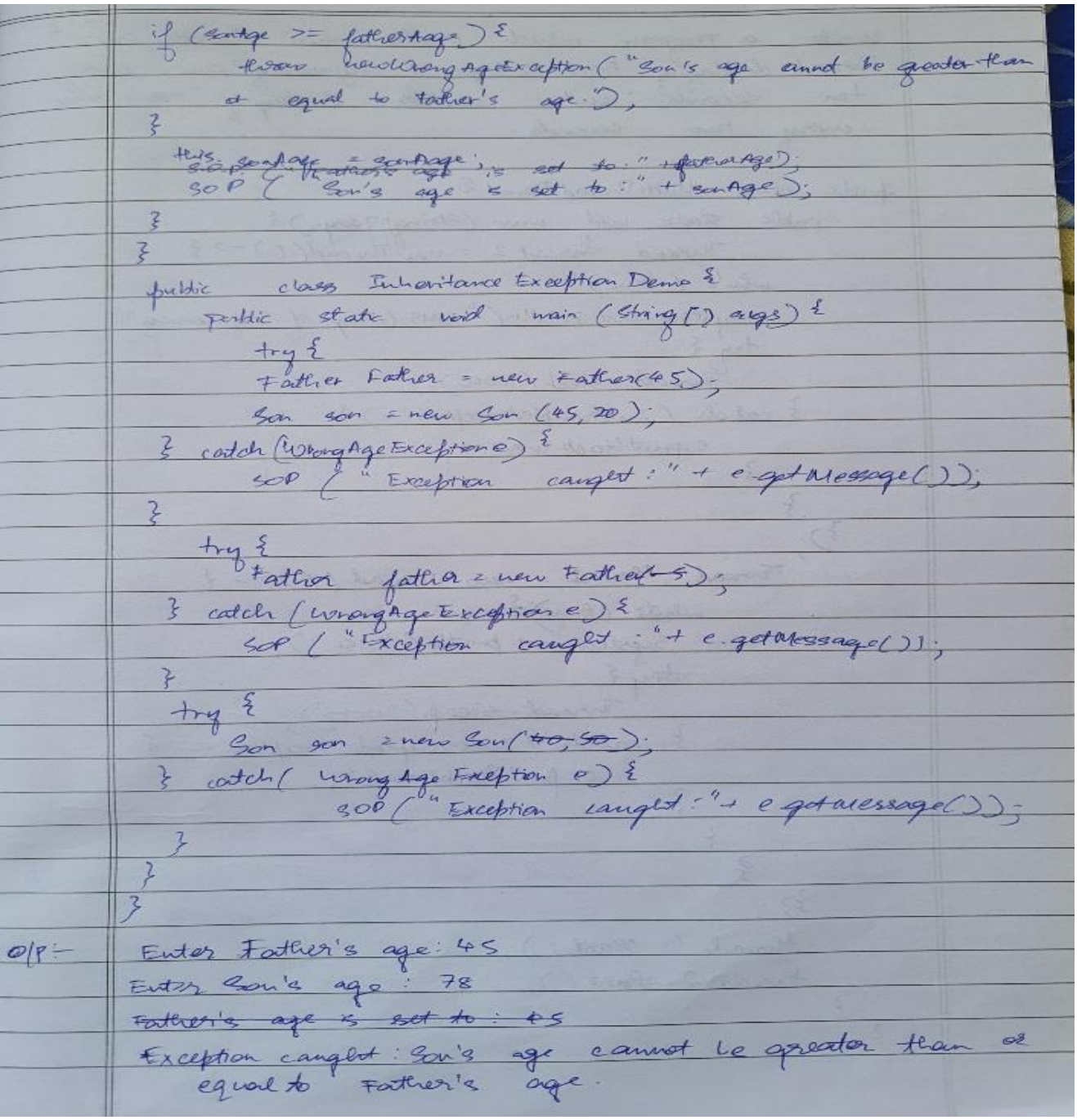
Program 08
Handling of exceptions

Algorithm



which takes the age & throws exception WrongAge() when the input age < 0.
In Son class, implement a constructor that uses both father & son's age & throws an exception if son's age is >= father's age.

```
import java.util.*;
class WrongAgeException extends Exception {
    public WrongAgeException (String message) {
        super(message);
    }
}

class Father {
    int fatherAge;

public Father(int age) throws WrongAgeException {
    if (age < 0) {
        throw new WrongAgeException ("Father's age cannot be neg");
    }
    this.fatherAge = age;
    SOP("Father's age is set to: " + age);
}
}

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException ("Son's age cannot be neg");
        }
```



```
if (sonAge >= fatherAge) {
    throw new WrongAgeException("Son's age cannot be greater than
    or equal to father's age.");
}
    this.sonAge = sonAge;
    SOP("Son's age is set to: " + sonAge);
}
}
public class InheritanceExceptionDemo {
    public static void main(String[] args) {
        try {
        Father father = new Father(45);
        Son son = new Son(45, 20);
    } catch (WrongAgeException e) {
        SOP("Exception caught: " + e.getMessage());
}
    try {
        Father father = new Father(-5);
    } catch (WrongAgeException e) {
        SOP("Exception caught: " + e.getMessage());
}
    try {
        Son son = new Son(50, 50);
    } catch (WrongAgeException e) {
            SOP("Exception caught: " + e.getMessage());
    }
}
}
```

O/P:-

```
Enter Father's age: 45
Enter Son's age: 78
Father's age is set to: 45
Exception caught: Son's age cannot be greater than or
    equal to Father's age.
```

Code

```
import java.util.Scanner;

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

```
class Father {
    int fatherAge;

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

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age is set to: " + sonAge);
    }
}

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

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

            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();

            Father father = new Father(fatherAge); // Validates father's age
            Son son = new Son(fatherAge, sonAge); // Validates son's age

        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Invalid input! Please enter numeric values.");
        }
```

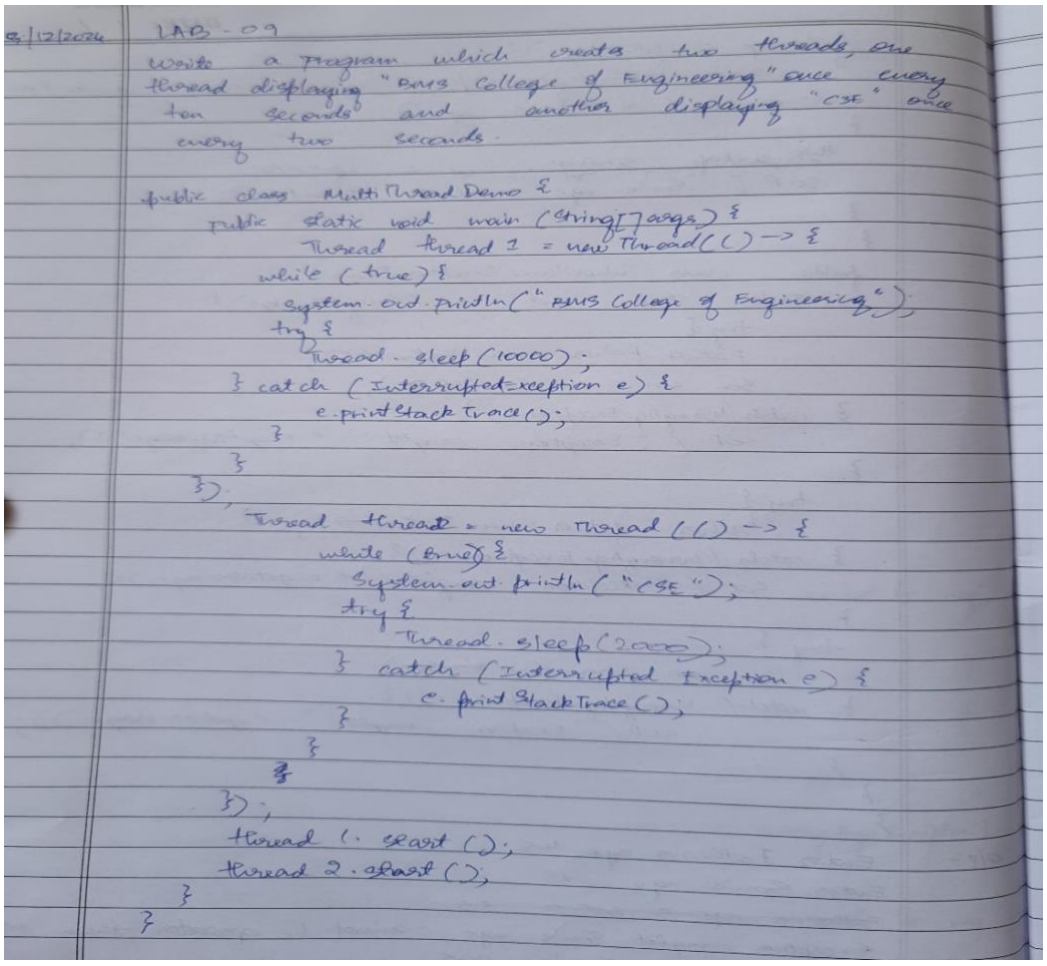
```
    }
  }
```

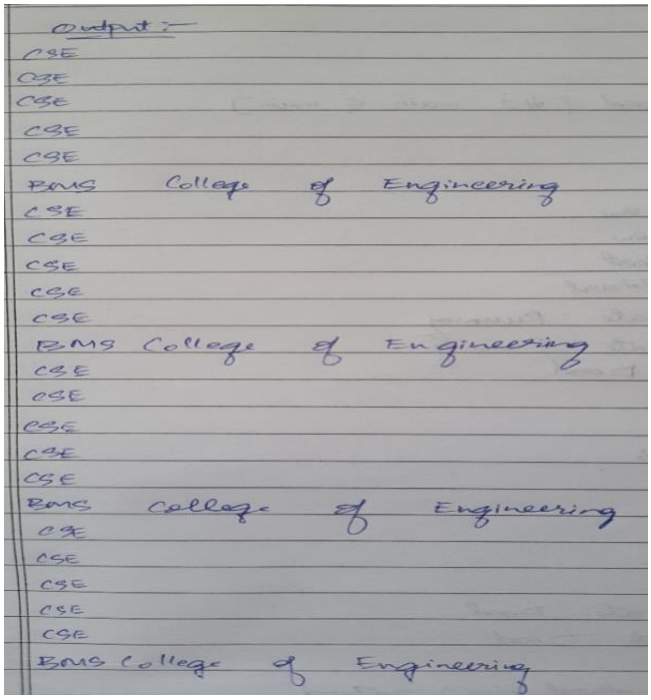
Output

```
Enter Father's age: 45
Enter Son's age: 78
Exception caught: Son's age cannot be greater than or equal to Father's age.
```

Program 09
Thread handling

Algorithm



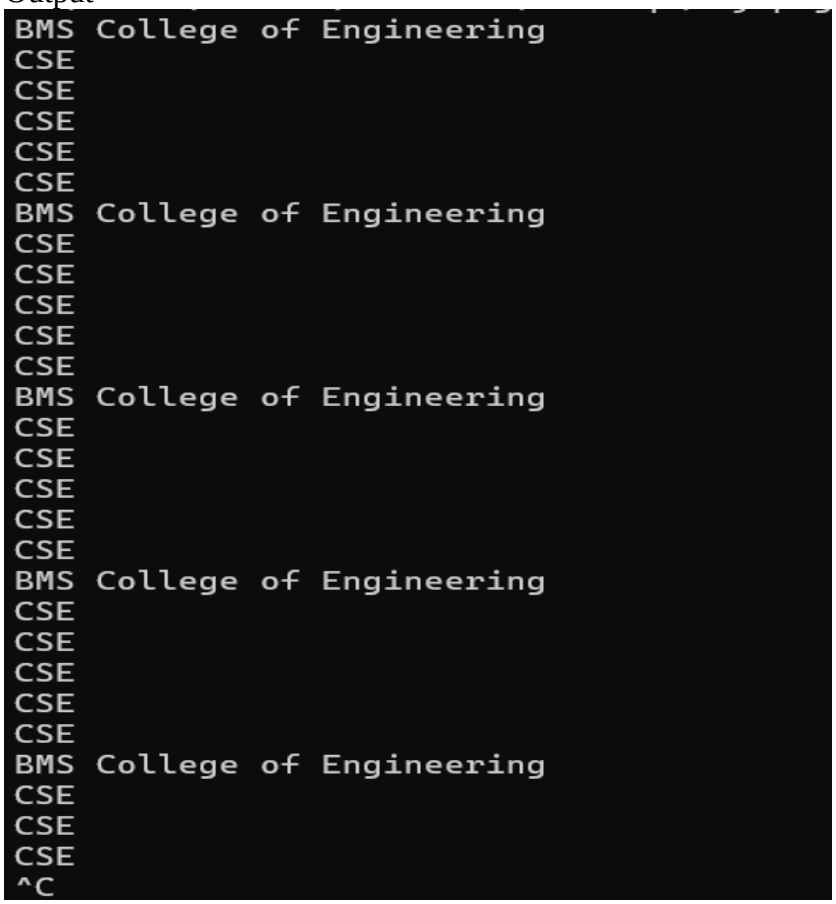



Code

```
public class MultiThreadDemo {

    public static void main(String[] args) {
            Thread thread1 = new Thread(() -> {
          while (true) {
             System.out.println("BMS College of Engineering");
             try {
                Thread.sleep(10000); // Pause for 10 seconds
             } catch (InterruptedException e) {
                e.printStackTrace();
             }
          }
       });

            Thread thread2 = new Thread(() -> {
          while (true) {
             System.out.println("CSE");
             try {
                Thread.sleep(2000); // Pause for 2 seconds
             } catch (InterruptedException e) {
                e.printStackTrace();
             }
          }
       });
```

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

Output




Program 10
Open end exercise
Algorithm

2) WAP that creates user interface to perform integer divisions. The user enter 2 nos in the text field, Num1 & Num2. Div. of Num1 & Num2 is displayed in Result field when ÷ button is clicked. If Num1 / Num2 were not an integer, the pgrm would throw a NumberFormatExeption. If Num2 = 0, throws an ArithmeticException Display the exception in a message dialog box.

```
import java.util.Scanner;
public class SimpleDivision {
    public static void main (String args[])
{
Scanner scanner = new Scanner (System.in);
try
{
S.O.P.(" enter 1st number (Num1): ");
int num1 = Integer.parseInt (scanner.nextLine());
S.O.P. (" enter 2nd no: ");
int num2 = Integer.parseInt(scanner.nextLine());
int result = num1 / num2;
S.O.P ("Result of division :" + result);
}
catch (NumberFormatException e)
{
S.O.P. ("Invalid input! Please enter integers only.");
}
catch ( ArithmeticException e)
{
S.O.P ("Error: Division by zero is not allowed.");
}
}
}
```

O/P:-

Enter the first number: 3

Enter the second numba: 0

Error: Division by zero is not allowed

Code

```
import java.util.Scanner;

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

        try {
            System.out.print("Enter the first number (Num1): ");
            int num1 = Integer.parseInt(scanner.nextLine());
            System.out.print("Enter the second number (Num2): ");
            int num2 = Integer.parseInt(scanner.nextLine());
            int result = num1 / num2;
            System.out.println("Result of division: " + result);
        } catch (NumberFormatException e) {
            // Handle invalid number input
            System.out.println("Invalid input! Please enter integers only.");
        } catch (ArithmeticException e) {
            // Handle division by zero
            System.out.println("Error: Division by zero is not allowed.");
        }
    }
}
```

Output