

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

# Object Oriented Java Programming (23CS3PCOOJ)

*Submitted by*

Himika Kakhani (**1BM23CS112**)

*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 **Himika Kakhani (1BM23CS112)**, 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 Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

## Index

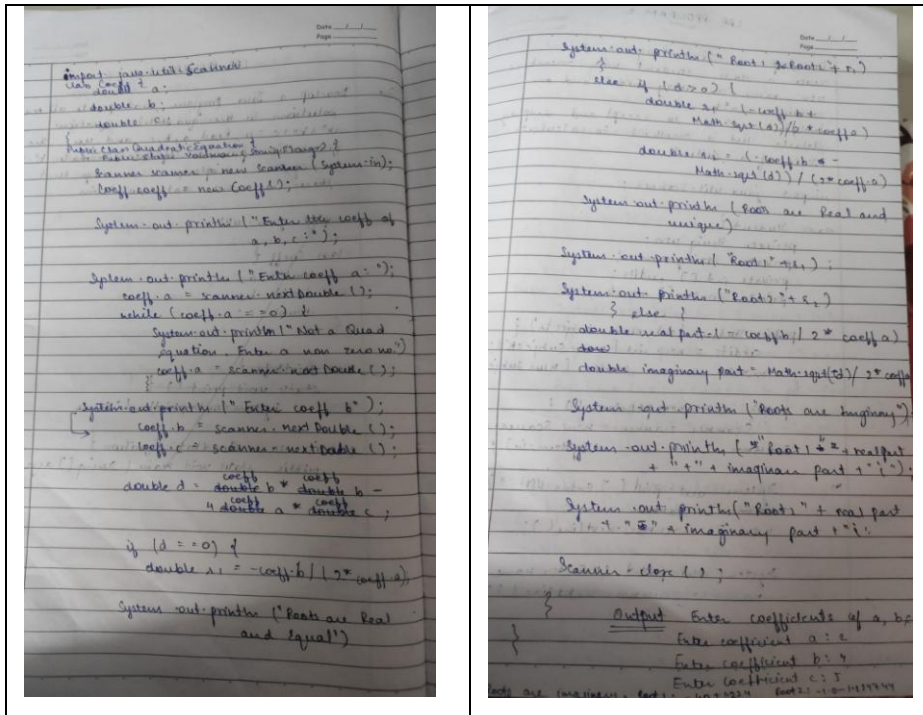
Sl. No.	Date	Experiment Title	Page No.
1	01/10 /24	Quadratic Equation	4-6
2	08/10/24	Student SGPA	7-10
3	15/10/24	Book Details	11-13
4	22/10/24	Area of the Shape	14-17
5	29/10/24	Bank	18-23
6	12/11/24	Package	24-29
7	19/11/24	Interface	30-33
8	26/11/24	Exception Handling Inheritance	34-36
9	03/12/24	Threads	37-38
10	03/12/24	Swing Demo	39-42

Github Link: <https://github.com/himika03/Java/blob/main/Main.java>

## **Program 1**

### Quadratic Equation

#### **Algorithm:**



#### **Code:**

```
import java.util.Scanner;
```

```
class Coeff {  
    double a;  
    double b;  
    double c;  
}
```

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

```

Scanner scanner = new Scanner(System.in);
Coeff coeff = new Coeff();

System.out.println("Enter the coefficients of a, b, c:");

System.out.print("Enter coefficient a: ");
coeff.a = scanner.nextDouble();
while (coeff.a == 0) {
    System.out.println("Not a quadratic equation. Please enter a non-zero value for a:");
    coeff.a = scanner.nextDouble();
}

System.out.print("Enter coefficient b: ");
coeff.b = scanner.nextDouble();
System.out.print("Enter coefficient c: ");
coeff.c = scanner.nextDouble();

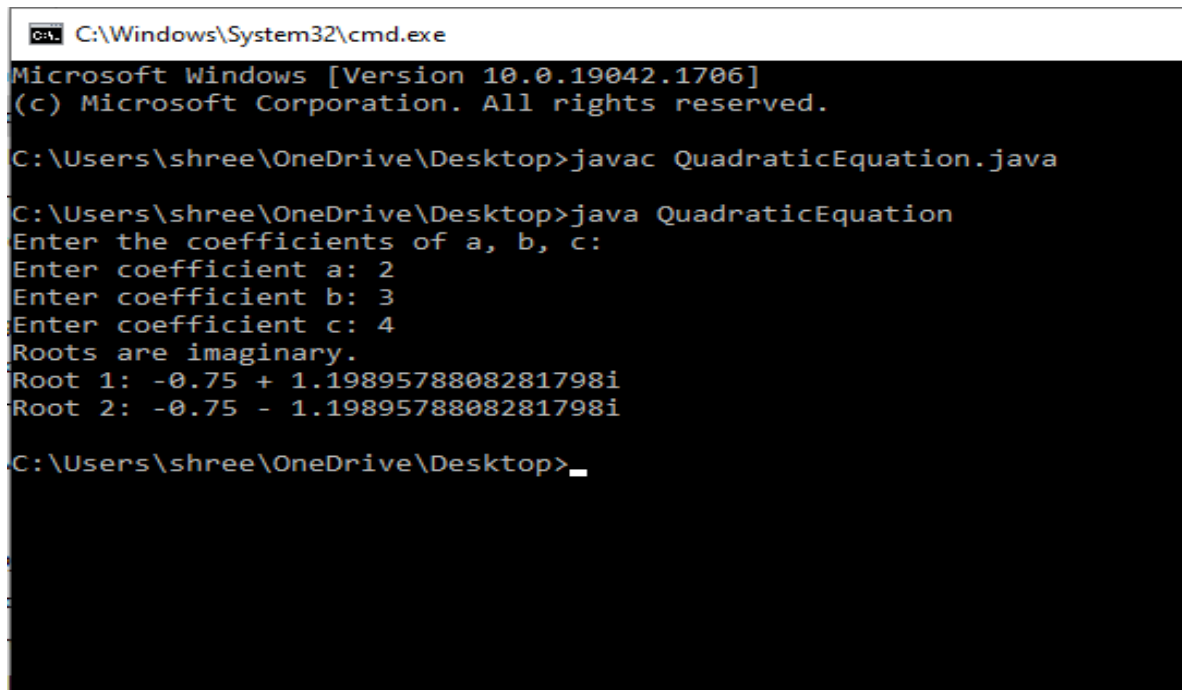
double d = coeff.b * coeff.b - 4 * coeff.a * coeff.c;

if (d == 0) {
    double r1 = -coeff.b / (2 * coeff.a);
    System.out.println("Roots are real and equal.");
    System.out.println("Root 1 and Root 2: " + r1);
} else if (d > 0) {
    double r1 = (-coeff.b + Math.sqrt(d)) / (2 * coeff.a);
    double r2 = (-coeff.b - Math.sqrt(d)) / (2 * coeff.a);
    System.out.println("Roots are real and unique.");
    System.out.println("Root 1: " + r1);
    System.out.println("Root 2: " + r2);
} else {
    double realPart = -coeff.b / (2 * coeff.a);
    double imaginaryPart = Math.sqrt(-d) / (2 * coeff.a);
    System.out.println("Roots are imaginary.");
    System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
    System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
}

```

```
    }  
    scanner.close();  
}  
}
```

### Output:

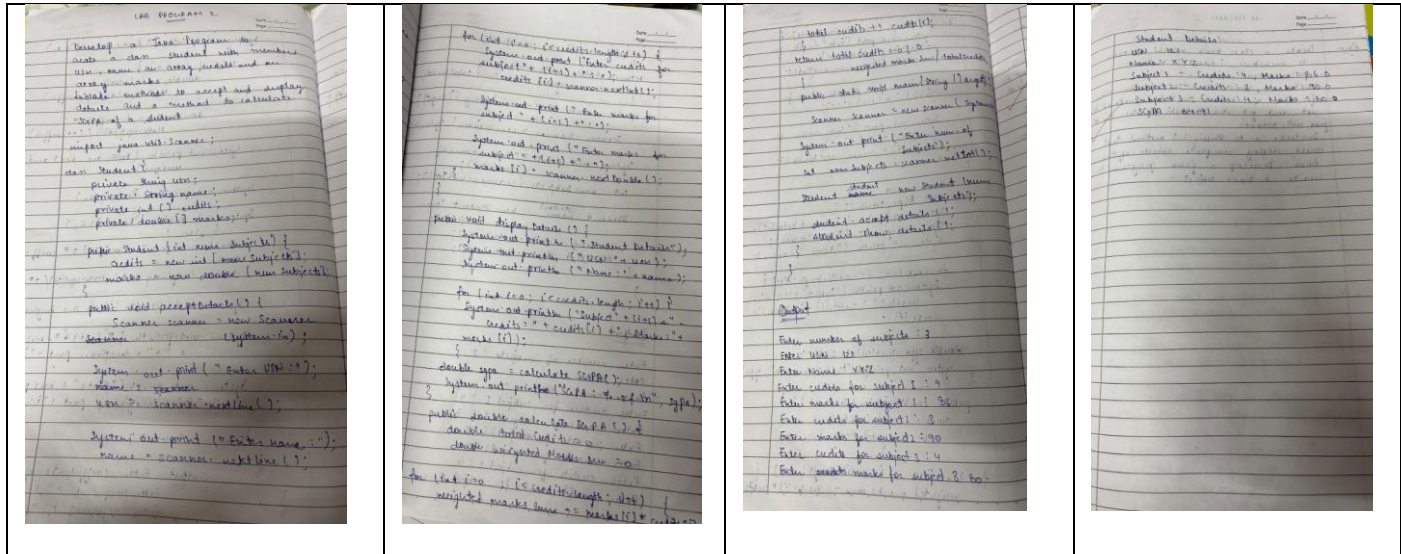


```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.19042.1706]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\shree\OneDrive\Desktop>javac QuadraticEquation.java  
  
C:\Users\shree\OneDrive\Desktop>java QuadraticEquation  
Enter the coefficients of a, b, c:  
Enter coefficient a: 2  
Enter coefficient b: 3  
Enter coefficient c: 4  
Roots are imaginary.  
Root 1: -0.75 + 1.1989578808281798i  
Root 2: -0.75 - 1.1989578808281798i  
  
C:\Users\shree\OneDrive\Desktop>_
```

## Program 2

### Student SGPA

#### Algorithm:



#### Code:

```
import java.util.Scanner;
```

```
class Subject {
    int grade;
    int credits;
}
```

```
class Student {
    String usn;
```

```
String name;  
double SGPA;  
Subject[] subjects;
```

```
Student() {  
    subjects = new Subject[8];  
    for (int i = 0; i < 8; i++) {  
        subjects[i] = new Subject();  
    }  
}
```

```
void getDetails(Scanner sc) {  
    System.out.println("Enter USN:");  
    usn = sc.nextLine();  
    System.out.println("Enter name:");  
    name = sc.nextLine();  
}
```

```
void getMarks(Scanner sc) {  
    double totalScore = 0;  
    int totalCredits = 0;
```

```
    System.out.println("Enter marks for 8 subjects:");  
    for (int j = 0; j < 8; j++) {  
        System.out.println("Enter marks for subject " + (j + 1) + ":");  
        int marks = sc.nextInt();  
        System.out.println("Enter the credits for subject " + (j + 1) + ":");  
        int credits = sc.nextInt();
```

```
        int grade = (marks / 10) + 1;  
        if (grade > 10) grade = 10;
```

```
        subjects[j].credits = credits;  
        subjects[j].grade = grade;
```

```
        totalScore += grade * credits;  
        totalCredits += credits;  
    }
```



```

        SGPA = totalScore / totalCredits;
    }

    void displaySGPA() {
        System.out.println("SGPA of student " + name + " (" + usn + "): " + SGPA);
    }
}

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

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Entering details for student " + (i + 1));
            students[i] = new Student();
            students[i].getDetails(sc);
            students[i].getMarks(sc);
            students[i].displaySGPA();
        }

        sc.close();
    }
}

```

**Output:**

---

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shree\OneDrive\Desktop>javac Student.java

C:\Users\shree\OneDrive\Desktop>java Student
Enter number of subjects: 3
Enter USN: 1bm001
Enter Name: abc
Enter credits for subject 1: 3
Enter marks for subject 1: 50
Enter credits for subject 2: 3
Enter marks for subject 2: 50
Enter credits for subject 3: 3
Enter marks for subject 3: 50

Student Details:
USN: 1bm001
Name: abc
Subject 1 - Credits: 3, Marks: 50.0
Subject 2 - Credits: 3, Marks: 50.0
Subject 3 - Credits: 3, Marks: 50.0
SGPA: 50.00

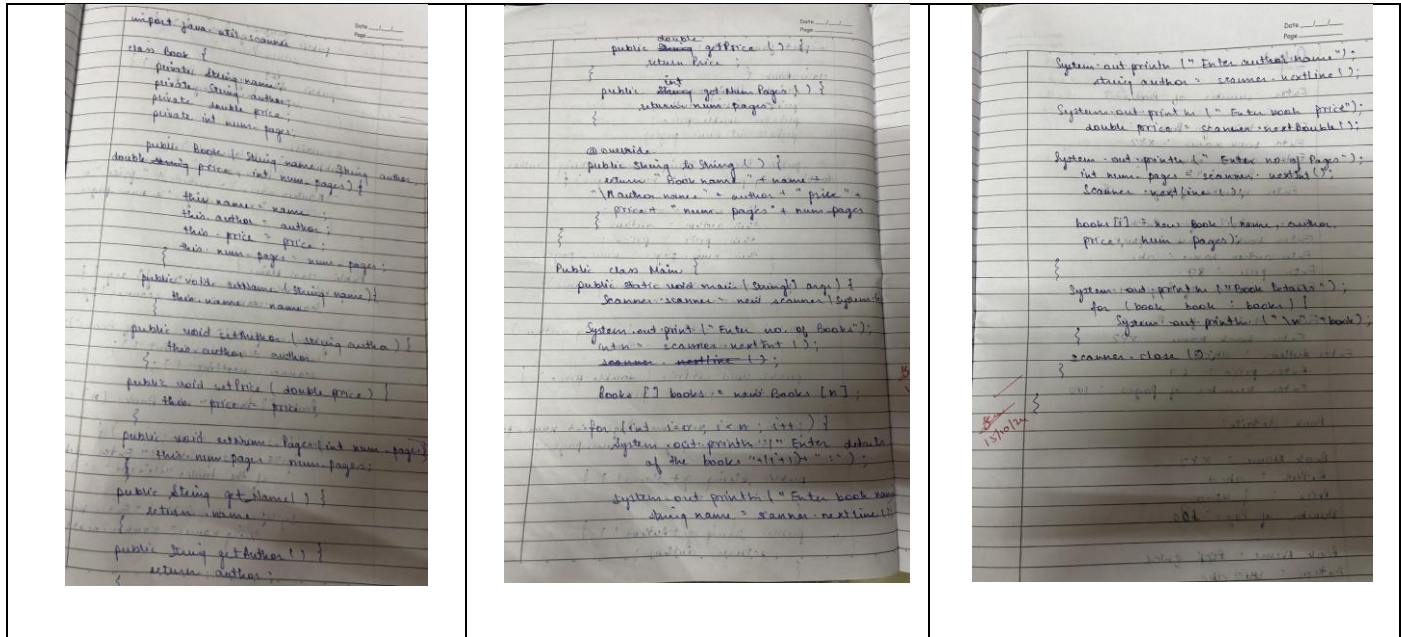
C:\Users\shree\OneDrive\Desktop>
```

---

### Program 3

#### Book Details

#### Algorithm:



#### Code:

```
import java.util.Scanner;

class Books{
    String name;
    String author;
    int price;
    int numPages;
    Books(String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public void setName (String name) {
        this.name = name;
    }
    public void setAuthor (String author) {
        this.author = author;
    }
    public void setPrice (double price) {
        this.price = price;
    }
    public void setNumPages (int numPages) {
        this.numPages = numPages;
    }
    public String getName () {
        return name;
    }
    public String getAuthor () {
        return author;
    }
}
```

```

        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString()

    {

        String name, author, price, numPages;

        name = "Book name: " + this.name + "\n";

        author = "Author name: " + this.author + "\n";

        price = "Price: " + this.price + "\n";

        numPages = "Number of pages: " + this.numPages + "\n";

        return name + author + price + numPages;

    }
}
class main{

    public static void main(String args[])

    {

        Scanner s = new Scanner(System.in);

        int n;
        int i;
        String name;
        String author;
        int price;
        int numPages;
        n=s.nextInt();
        Books b[];
        b=new Books[n];

```

```

for(i=0;i<n;i++){
    System.out.println("enter book name");
    name = s.next();
    System.out.println("enter author name");
    author = s.next();
    System.out.println("enter the price");
    price=s.nextInt();
    System.out.println("enter number of pages");
    numPages = s.nextInt();
    b[i] = new Books(name,author,price,numPages);
}
for(i=0;i<n;i++){
    System.out.println("Book Details");
    System.out.println(b[i].toString());
}
s.close();
}
}

```

**Output:**

```

Enter the number of books: 3

Enter details for Book 1:
Enter book name: XYZ
Enter author name: abcd
Enter price: 45
Enter number of pages: 180

Enter details for Book 2:
Enter book name: qwer
Enter author name: abc
Enter price: 89
Enter number of pages: 100

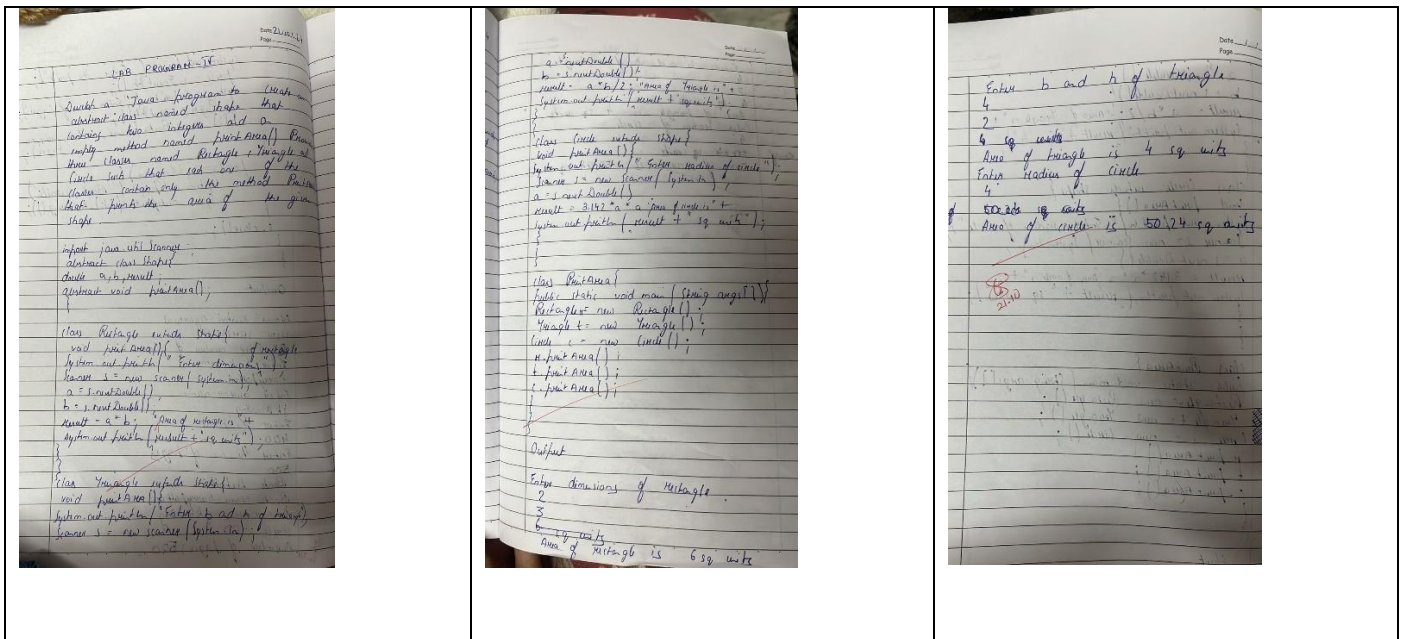
Enter details for Book 3:
Enter book name: XYZ
Enter author name: abc
Enter price: 67
Enter number of pages: 100

```

## Program 4

### Area of the Shape

#### Algorithm:



#### Code:

```
import java.util.Scanner;
```

```
abstract class shape{  
    int dim1;  
    int dim2;
```

```

        abstract void printarea();
    }

class rectangle extends shape{
    public rectangle(){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);
        System.out.println("enter the l and b");
        dim1=s.nextInt();
        dim2=s.nextInt();

        int area=dim1*dim2;
        System.out.println("area of rectangle: "+area);
    }
}

class triangle extends shape{
    public triangle(){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);

        System.out.println("enter the l and b");
        dim1=s.nextInt();
        dim2=s.nextInt();
        double area=(dim1*dim2)/2;
        System.out.println("area of triangle: "+area);
    }
}

class circle extends shape{
    final double Pi=3.14;
    public circle(){
        this.dim1=dim1;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);

```

```

        System.out.println("enter the radius");
        dim1=s.nextInt();

        double area=Pi*dim1*dim1;
        System.out.println("area of circle: "+area);
    }
}
public class main{
    public static void main (String [] args){

```

```

        rectangle R =new rectangle();

        R.printarea();

        triangle T = new triangle();
        T.printarea();

        circle C = new circle();
        C.printarea();
    }
}

```



## Output:

---

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shree\OneDrive\Desktop>javac Shape1.java

C:\Users\shree\OneDrive\Desktop>java Shape1
enter the l and b
12 14
area of rectangle: 168
enter the l and b
12 14
area of triangle: 84.0
enter the radius
12
area of circle: 452.15999999999997

C:\Users\shree\OneDrive\Desktop>_
```

**Algorithm:**

```
this.name = name;
this.age = age;
this.balance = balance;
```

```
{ public void deposit (double amount) {  
    if (amount > 0) {  
        balance += amount;  
        System.out.println ("Deposit balance  
new balance " + balance);  
    }  
    else {  
        System.out.println ("Amount is not positive")  
    }  
}  
  
{ public void withdraw (double amount) {  
    if (amount <= 0 || amount >= balance) {  
        System.out.println ("Balance is negative or  
public void display_balance () {  
    System.out.println ("Balance = " + balance);  
}  
  
public void withdraw (double amount) {  
    if (amount <= 0 || amount >= balance) {  
        System.out.println ("Withdrawal successful  
new balance = " + balance);  
    }  
    else {  
        System.out.println ("Insufficient Balance");  
    }  
}  
  
// Withdraw extends Account {  
    double withdraw ();  
}
```

```
double balance, double interest;
// (name, accno, balance)
this.interest = interest;
}
public void computedInterest(int year)
{
    double interest = balance * Math.pow(
        1 + (int) year - balance);
    balance += interest;
    System.out.println("Interested Amount: " +
        new balance + " balance");
}
public void withdraw(double amount)
{
    super.withdraw(amount);
}
}
// This interface extends Account
double minBalance;
double service charge;
public interface SavingAccName, Saving
double balance, double minBalance, double
service charge;
}
// (name, accno, balance)
this.minBalance = minBalance;
this.service charge = service charge;
}
public void withdraw (double amount)
```

```

g) balance amount < min balance)
system.out.println("Balance below
min balance, charge service charge") +
service charge);
balance -= (interest + service charge);
}

// public class bank {
    // balance = amount
    system.out.println("Withdrawing successful
    New balance " + balance);
}

public class bank {
    public static void main (String [] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount type
        1) Saving 2) Current");
        int acc_type = (sc.nextInt());
        sc.nextLine();
        System.out.println("Enter name");
        String name = (sc.nextLine());
        System.out.println("Enter acno");
        String acno = (sc.nextLine());
        System.out.println("Enter the balance");
        double balance = (sc.nextDouble());
        Account account =
            if (acc_type == 1) {
                System.out.println("Enter deposit");
                double d = (sc.nextDouble());

```

```

account = new SavAcc (name, acc no,
balance, interest);
}
else
{
    System.out.println ("Enter your balance");
    numBalance = sc.nextInt();
    System.out.println ("Service charge");
    service charge = sc.nextInt();
    account = new SavAcc (name, acc no,
    balance, numBalance, service charge);
}
int choice;
do
{
    System.out.println ("Deposit");
    System.out.println ("Display balance");
    System.out.println ("Withdraw amount");
    if (account instanceof savacc)
    {
        System.out.println ("compute and deposit");
    }
    System.out.println ("Exit");
    System.out.println ("Enter your choice");
    int choice = sc.nextInt();
    switch (choice)
    {
        case 1:
            System.out.println ("Enter the amount to deposit");
            double deposit amount = sc.nextInt();
            account.deposit (deposit amount);
            break;
    }
}

```

```

case 2:
    account.displayBalance();
    break;
case 3:
    System.out.println ("Enter amount to withdraw");
    double withdraw amount = sc.nextInt();
    account.withdraw (withdraw amount);
    break;
case 4:
    if (account instanceof savacc)
    {
        System.out.println ("Enter year of year");
        int year = sc.nextInt();
        (savacc) amount - compute and deposit,
        interest (year);
    }
    else
    {
        System.out.println ("Invalid");
    }
    break;
case 5:
    System.out.println ("Thank you for choosing");
    break;
default:
    System.out.println ("Invalid choice");
}
while (choice != 5);
sc.close();
}

```

## Code:

```
import java.util.Scanner;
```

```

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

```

```

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

```

```

        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

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

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

    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

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

    public void compoundDeposit() {
        double interest = balance * (interestRate / 100);
        deposit(interest);
        System.out.println("Interest of " + interest + " deposited");
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn amount is: " + amount);
        } else {
            System.out.println("Insufficient amount for withdrawal.");
            return;
        }
        displayBalance();
    }
}

```

```

class CurAcct extends Account {
    private static final double minBalance = 1000.0;
    private static final double serviceCharge = 50.0;

    public CurAcct(String customerName, String accountNumber) {
        super(customerName, "current", accountNumber);
    }

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn amount is: " + amount);
        } else {
            System.out.println("Insufficient amount for withdrawal.");
            return;
        }

        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained");
            System.out.println("Service charge of: " + serviceCharge + " included");
        }
        displayBalance();
    }
}

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

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your account type (savings/current):");
        String accountType = scanner.nextLine();
        System.out.println("Enter account number:");
        String accountNumber = scanner.nextLine();
        System.out.println("Enter your name:");
        String customerName = scanner.nextLine();

        Account account;
        if (accountType.equals("savings")) {
            System.out.println("Enter the interest rate:");

```

```

        double interestRate = scanner.nextDouble();
        account = new SavAcct(customerName, accountNumber, interestRate);
    } else {
        account = new CurAcct(customerName, accountNumber);
    }

    while (true) {
        System.out.println("1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.println("Enter amount to deposit:");
                double depositAmount = scanner.nextDouble();
                account.deposit(depositAmount);
                break;
            case 2:
                System.out.println("Enter amount to withdraw:");
                double withdrawAmount = scanner.nextDouble();
                account.withdraw(withdrawAmount);
                break;
            case 3:
                account.displayBalance();
                break;
            case 4:
                System.out.println("Exit");
                scanner.close();
                return;
            default:
                System.out.println("Try again");
        }
    }
}

```

## Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shree\OneDrive\Desktop>javac Bank.java
```

```
C:\Users\shree\OneDrive\Desktop>java Bank
Enter your account type (savings/current):
savings
Enter account number:
1234
Enter your name:
abc
Enter the interest rate:
```

```
7
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Enter amount to deposit:
10000
Deposited amount is: 10000.0
Current balance is: 10000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
3
```

```
C:\Windows\System32\cmd.exe
```

```
3. Display Balance
4. Exit
2
Enter amount to withdraw:
3000
Withdrawn amount is: 3000.0
Current balance is: 7000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
3
Current balance is: 7000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
4
Exiting...

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

## Program 6

### Packages

#### Algorithm:

LAB PROGRAM - 6

Create a package CIE which has two classes Student and Internal. The class Student has members like user, name, sem. The class Internal has an array that stores the internal marks stored score in five courses of the current semester of the student. Create another package SEE which has the class External which is derived class of Student. This class has an array that stores SEE marks of the current sem of student. Import the two packages in a file that declares final marks of students in all 5 courses.

```
package CIE;
import java.util.Scanner;

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

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        Scanner (System.in);
        System.out.print("Enter User:");
        this.user = s.nextLine();
        System.out.print("Enter Name:");
        this.name = s.nextLine();
        System.out.print("Enter Semester:");
        this.sem = s.nextInt();
    }
}
```

```
package CIE;
import java.util.Scanner;

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

    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the marks of 5 courses:");
        for (int i = 0; i < 5; i++) {
            System.out.print("course " + (i+1) + ":");
            marks[i] = s.nextInt();
        }
    }

    public void displayCIEmarks() {
        System.out.print("Internal Marks");
        for (int i = 0; i < 5; i++) {
            System.out.print("course " + (i+1) + ": " + marks[i]);
        }
    }
}
```

```
package SEE;
import CIE.Student;
import CIE.Internal;
import java.util.Scanner;

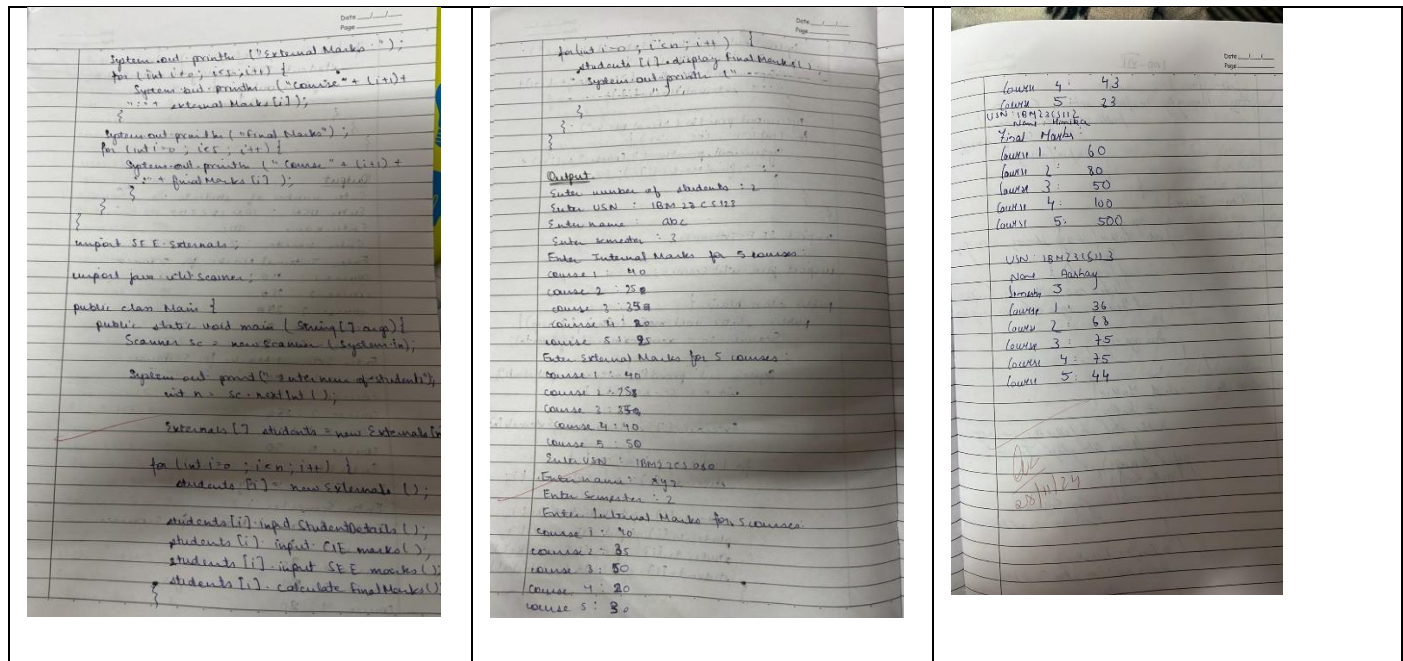
public class External extends Internal {
    protected int externalMarks[] = new int[5];
    protected int finalMarks[] = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter External Marks for 5 courses:");
        for (int i = 0; i < 5; i++) {
            System.out.print("course " + (i+1) + ":");
            externalMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        displayCIEmarks();
    }
}
```





**Code: CIE**  
*Internals.java*

package CIE;

import java.util.Scanner;

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

public void inputCIEMarks() {  
 Scanner s = new Scanner(System.in);  
 System.out.println("Enter Internal Marks for 5 courses: ");  
 for (int i = 0; i < 5; i++) {  
 System.out.print("Course " + (i + 1) + ": ");  
 marks[i] = s.nextInt();  
 }  
 }

public void displayCIEMarks() {  
 System.out.println("Internal Marks: ");  
 for (int i = 0; i < 5; i++) {  
 System.out.println("Course " + (i + 1) + ": " + marks[i]);  
 }  
 }

```
    }  
  }  
}
```

*Student.java*

```
package CIE;
```

```
import java.util.Scanner;
```

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

```
    public void inputStudentDetails() {  
        Scanner s = new Scanner(System.in);  
        System.out.print("Enter USN: ");  
        this.usn = s.nextLine();  
        System.out.print("Enter Name: ");  
        this.name = s.nextLine();  
        System.out.print("Enter Semester: ");  
        this.sem = s.nextInt();  
    }  
    public void displayStudentDetails() {  
        System.out.println("USN: " + usn);  
        System.out.println("Name: " + name);  
        System.out.println("Semester: " + sem);  
    }  
}
```

**SEE:**

*Student.java*

```
package SEE;
```

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

```
public class Externals extends Internals {  
    protected int externalMarks[] = new int[5];  
    protected int finalMarks[] = new int[5];
```

```

public Externals() {    externalMarks =
new int[5];
    finalMarks = new int[5];
}
public void inputSEEmarks() {
Scanner s = new Scanner(System.in);
    System.out.println("Enter External Marks for 5 courses: ");
    for (int i = 0; i < 5; i++) {
        System.out.print("Course " + (i + 1) + ": ");
externalMarks[i] = s.nextInt();
    }
}
public void calculateFinalMarks() {
for (int i = 0; i < 5; i++) {
    finalMarks[i] = marks[i] + externalMarks[i];
}
}
public void displayFinalMarks() {
displayStudentDetails();
displayCIEmarks();

    System.out.println("External Marks: ");
    for (int i = 0; i < 5; i++) {
        System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
    }

    System.out.println("Final Marks: ");
for (int i = 0; i < 5; i++) {
    System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
}
}
}

```

*Main.java*

```

import SEE.Externals;

import java.util.Scanner;

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

```

```

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

Externals[] students = new Externals[n];

    for (int i = 0; i < n; i++) {
students[i] = new Externals();
students[i].inputStudentDetails();
students[i].inputCIEMarks();
students[i].inputSEEMarks();
        students[i].calculateFinalMarks();
    }
        for (int i = 0; i < n; i++) {
students[i].displayFinalMarks();
        System.out.println(" ..... ");
    }
    }
}

```

## Output:

```
C:\Users\STUDENT\Desktop\1BMK>javac -d . CIE/Student.java
C:\Users\STUDENT\Desktop\1BMK>javac -d . CIE/Internals.java
C:\Users\STUDENT\Desktop\1BMK>javac -d . SEE/Externals.java
C:\Users\STUDENT\Desktop\1BMK>javac Main.java

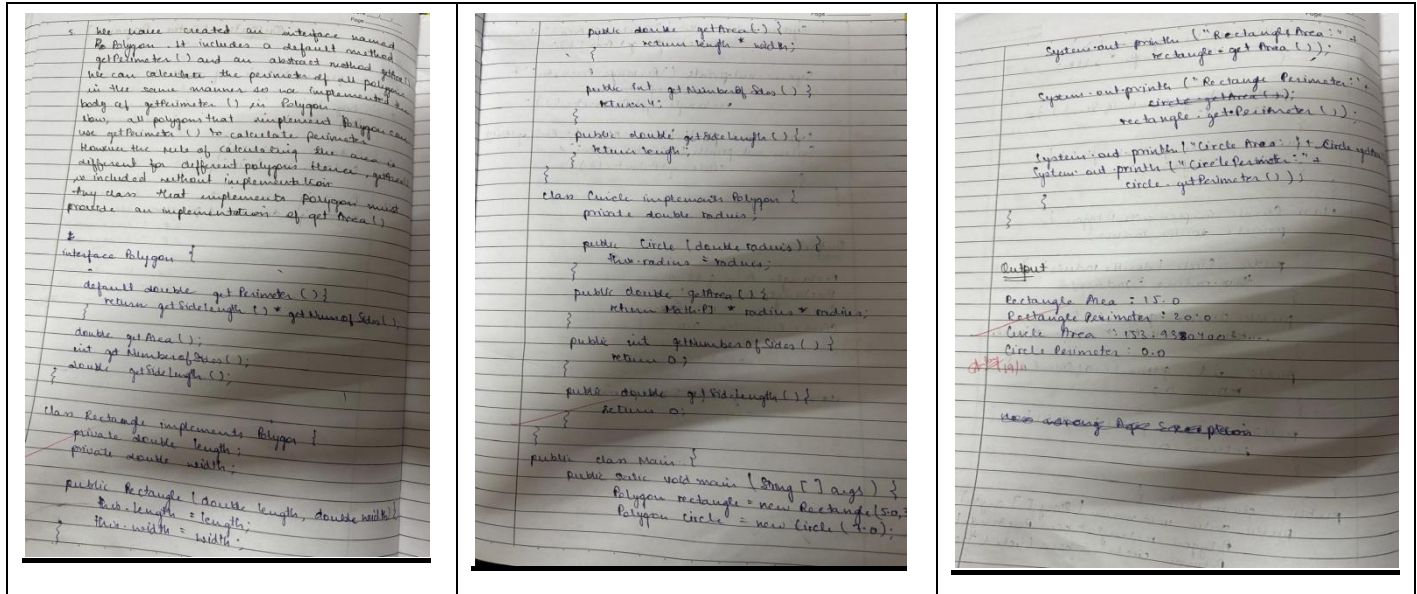
C:\Users\STUDENT\Desktop\1BMK>java Main
Enter number of students: 2
Enter USN: 1bm23cs121
Enter Name: himika
Enter Semester: 3
Enter Internal Marks for 5 courses:
Course 1: 30
Course 2: 40
Course 3: 20
Course 4: 50
Course 5: 30
Enter External Marks for 5 courses:
Course 1: 30
Course 2: 40
Course 3: 30
Course 4: 50
Course 5: 20
Enter USN: 1bm23cs122
Enter Name: chiraiya
Enter Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 12
Course 2: 23
Course 3: 43
Course 4: 32
Course 5: 21
Enter External Marks for 5 courses:
Course 1: 24
Course 2: 45
Course 3: 32
Course 4: 43
Course 5: 23
USN: 1bm23cs121
Name: himika
Semester: 3
Internal Marks:
Course 1: 30
Course 2: 40
Course 3: 20
Course 4: 50
Course 5: 30
```

```
Course 5: 21
Enter External Marks for 5 courses:
Course 1: 24
Course 2: 45
Course 3: 32
Course 4: 43
Course 5: 23
USN: 1bm23cs121
Name: himika
Semester: 3
Internal Marks:
Course 1: 30
Course 2: 40
Course 3: 20
Course 4: 50
Course 5: 30
External Marks:
Course 1: 30
Course 2: 40
Course 3: 30
Course 4: 50
Course 5: 20
Final Marks:
Course 1: 60
Course 2: 80
Course 3: 50
Course 4: 100
Course 5: 50
-----
USN: 1bm23cs122
Name: chiraiya
Semester: 2
Internal Marks:
Course 1: 12
Course 2: 23
Course 3: 43
Course 4: 32
Course 5: 21
External Marks:
Course 1: 24
Course 2: 45
Course 3: 32
Course 4: 43
Course 5: 23
Final Marks:
Course 1: 36
Course 2: 68
Course 3: 75
Course 4: 75
Course 5: 44
-----
```

## Program

### Interfaces

### Algorithm:



### Code:

```
interface Polygon {
```

```
    default double getPerimeter() {  
        return 0.0;  
    }  
}
```

```
    double getArea();  
}
```

```
class Rectangle implements Polygon {
```

```
    private double length;  
    private double width;
```

```
    public Rectangle(double length, double width) {
```

```
    this.length = length;
    this.width = width;
}
```

```
@Override
public double getArea() {
    return length * width;
}
```

```
@Override
public double getPerimeter() {
    return 2 * (length + width);
}
}
```

```
class Circle implements Polygon {
    private double radius;
```

```
    public Circle(double radius) {
        this.radius = radius;
    }
```

```
@Override
public double getArea() {
    return Math.PI * radius * radius;
}
```

```
@Override
public double getPerimeter() {
    return 2 * Math.PI * radius;
}
```

```
}
```

```
class Triangle implements Polygon {  
    private double side1, side2, side3;
```

```
  
    public Triangle(double side1, double side2, double side3) {  
        this.side1 = side1;  
        this.side2 = side2;  
        this.side3 = side3;  
    }  
  
    @Override
```

```
    public double getArea() {  
        double s = (side1 + side2 + side3) / 2;  
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));  
    }  
  
    @Override
```

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

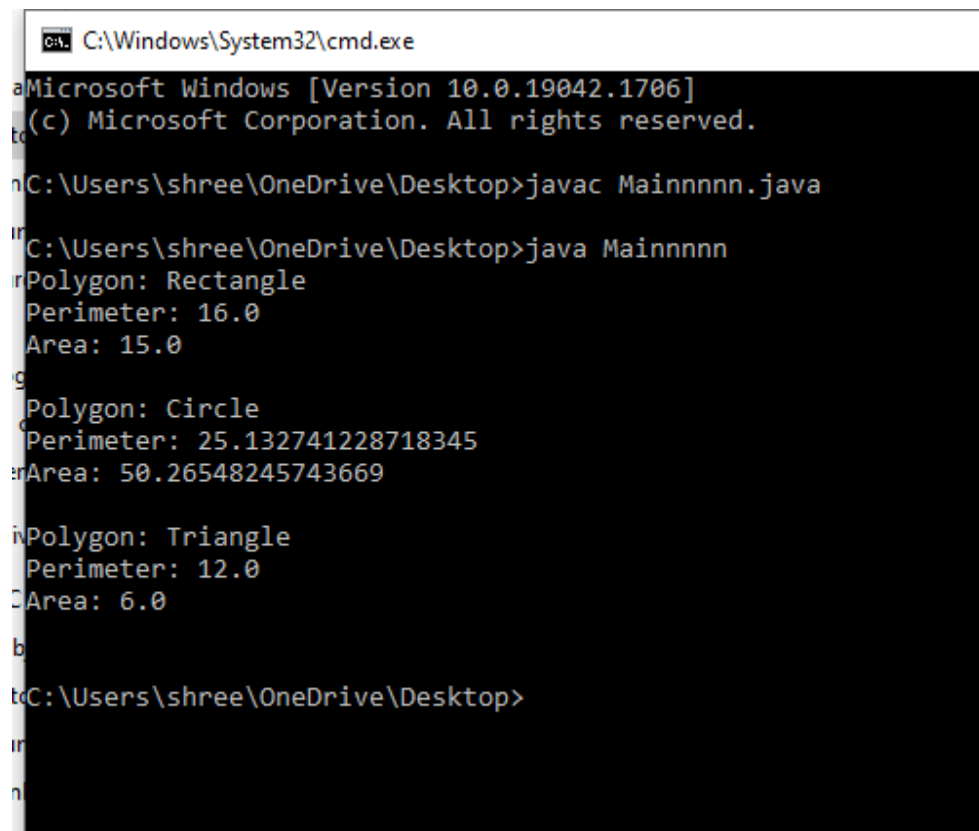
```
public class Mainnnnn {  
    public static void main(String[] args) {
```

```
  
        Polygon[] polygons = {  
            new Rectangle(5, 3),  
            new Circle(4),  
            new Triangle(3, 4, 5)  
        };  
  
        for (Polygon polygon : polygons) {
```



```
        System.out.println("Polygon: " + polygon.getClass().getSimpleName());
        System.out.println("Perimeter: " + polygon.getPerimeter());
        System.out.println("Area: " + polygon.getArea());
        System.out.println();
    }
}
}
```

### **Output:**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shree\OneDrive\Desktop>javac Mainnnnn.java

C:\Users\shree\OneDrive\Desktop>java Mainnnnn
Polygon: Rectangle
Perimeter: 16.0
Area: 15.0

Polygon: Circle
Perimeter: 25.132741228718345
Area: 50.26548245743669

Polygon: Triangle
Perimeter: 12.0
Area: 6.0

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

## Program 7

### Exception Handling Inheritance

#### Algorithm:

Ques. 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 an exception WrongAge() when input age < 0. In Son class, implement a constructor that uses father's age and son's age and throws exception if son's age > father's age.

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

class Father {
    int fatherAge;
    public Father(int fatherAge) {
        throws (WrongAgeException) {
            throws new WrongAgeException("Father's age can't be -ve");
        }
        this.fatherAge = fatherAge;
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) {
        super(fatherAge);
        if (sonAge < 0) {
            throws new WrongAgeException("Son's age can't be -ve");
        }
        if (sonAge > fatherAge) {
            throws new WrongAgeException("Father and son's age can't be equal and greater than father's age");
        }
        this.sonAge = sonAge;
        System.out.println("Son's Age: " + sonAge);
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            System.out.println("Test Case 1:");
            Son son = new Son(40, 20);
            System.out.println("Test Case 2:");
            Son invalidSon = new Son(20, 40);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output

Test Case 1:  
Father's Age: 40  
Son's Age: 20

Test Case 2:  
Father's Age: 20  
Exception: Son's age can't be equal or greater than father's age

Test Case 3:  
Exception: Father's age is negative

#### Code:

```
import java.util.Scanner;
```

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

```
class Father {
```

```

int age;

public Father(int age) throws WrongAgeException {
    if (age < 0) {
        throw new WrongAgeException("Father's age cannot be negative.");
    }
    this.age = 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;
    }
}

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

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

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

```

```

        Son son = new Son(fatherAge, sonAge);
        System.out.println("Father's age: " + son.age);
        System.out.println("Son's age: " + son.sonAge);

    } catch (WrongAgeException e) {
        System.out.println("Error: " + e.getMessage());
    } catch (Exception e) {
        System.out.println("Unexpected error: " + e.getMessage());
    } finally {
        scanner.close();
    }
}
}

```

### Output:

```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91889\OneDrive\Desktop\BMS PDF>javac ExceptionHandlingInheritance.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java ExceptionHandlingInheritance
Enter Father's age: 18
Enter Son's age: 5
Father's age: 18
Son's age: 5

C:\Users\91889\OneDrive\Desktop\BMS PDF>15
'15' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\91889\OneDrive\Desktop\BMS PDF>javac ExceptionHandlingInheritance.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java ExceptionHandlingInheritance
Enter Father's age: 15
Enter Son's age: 16
Error: Son's age cannot be greater than or equal to Father's age.

```

## Program 8

### Threads

### Algorithm:

LAB PROGRAM

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class BMSDisplayThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS college of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted: " + e.getMessage());
        }
    }
}

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

public class DisplayThreads {
 public static void main (String[] args) {
 BMSDisplayThread bmsThread = new BMSDisplayThread();
 CSEDisplayThread cseThread = new CSEDisplayThread();

 bmsThread.start();
 cseThread.start();
 }
}

Output: Printing every message

BMS College of Engineering

CSE

CSE

CSE

CSE

BMS College of Engineering

### Code:

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

```

    }

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

    public static void main(String[] args) {
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();

        bmsThread.start();
        cseThread.start();
    }
}

```

### Output:

```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

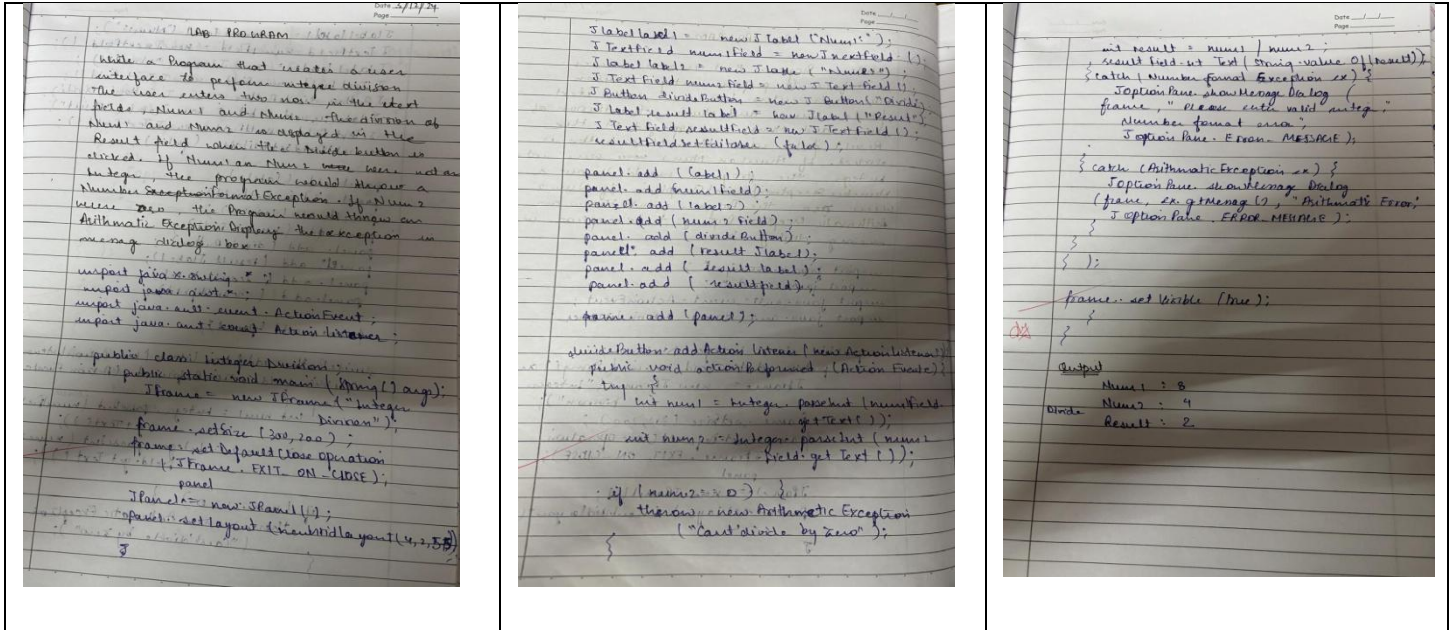
C:\Users\91889\OneDrive\Desktop\BMS PDF>javac Main.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE

```

## Swing Demo

### Algorithm:



**Code:**

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

```
class SwingDemo {
    SwingDemo() {
```

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

```

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

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

JButton button = new JButton("Calculate");

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
jfrm.add(err);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());

            int ans = a / b;
            alab.setText("A = " + a);
            blab.setText("B = " + b);

```



```

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

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

```

## Output:

