

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
on

**Object Oriented Java Programming  
(23CS3PCOOJ)**

*Submitted by*

Bhoomi Suresh Kota (**1BM23CS065**)

*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 Suresh Kota (1BM23CS065)**, who is a 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.

<b>Mrs. Swathi Sridharan</b> Assistant Professor Department of CSE, BMSCE	<b>Dr. Jyothi S Nayak</b> Professor & HOD Department of CSE, BMSCE
---	--

# **Index**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	01-10-2024	Quadratic Equations	4
2	08-10-2024	SGPA Calculator	6
3	15-10-2024	Getter setter methods	9
4	22-10-2024	Abstract classes	12
5	29-10-2024	Bank Account	16
6	12-11-2024	Packages	20
7	19-11-2024	Interfaces	23
8	26-11-2024	Exceptions	26
9	3-12-2024	Threads	28

**Github Link:**

<https://github.com/BhoomiSuresh/OOJ.git>

# Program 1

## Implement Quadratic Equation

### Algorithm:

Date \_\_\_\_\_  
Page \_\_\_\_\_

(3) Quadratic Equations

```
import java.util.Scanner; import java.lang.Math;
public class QuadEquations {
    public static void main (String [] args) {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter the coefficients a,b,c");
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();

        int D = b*b - 4*a*c;
        if (a<=0) {
            System.out.println ("Not quadratic");
        } else {
            if (D<0) {
                System.out.println ("Roots are imaginary");
            } else if (D>0) {
                System.out.println ("Roots are: ");
                double r1 = (-b + Math.sqrt(D)) / (2*a);
                double r2 = (-b - Math.sqrt(D)) / (2*a);
                System.out.println ("r1 + " + r2);
            } else {
                System.out.println ("Roots are: ");
                double r1 = (-b) / (2*a);
                System.out.println ("r1 + " + r1);
            }
        }
    }
}
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

O/p :

(i) Enter the coefficients a,b,c:  
12 1 4  
Roots are imaginary

(ii) Enter the coefficients a,b,c:  
1 6 1  
Roots are:  
-0.17157 -5.8284

(iii) Enter the coefficients a,b,c:  
1 2 1  
Roots are:  
-1.0 -1.0

### Code:

```
import java.util.*;
import java.lang.Math;
public class Main{
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("Enter the coefficients a, b, c");
int a = scan.nextInt();
int b= scan.nextInt();
int c = scan.nextInt();
int D = (b*b) - (4*a*c);
if(a <= 0)
System.out.println("Not quadratic");
else {
if(D < 0){
System.out.println("Roots are imaginary");
}
else if (D > 0) {
System.out.println("Roots are: ");
double r1= (-b+Math.sqrt(D)) /(2*a);
double r2=(-b-Math.sqrt(D))/(2*a);
System.out.println(r1+" "+r2);
}
else {
System.out.println("Roots are: ");
double r1 = (- b) / (2 * a);
System.out.println ( r1+" "+r1);
}
}
}
}
}
```

# Program 2

Implement SGPA Calculator

**Algorithm:**

LAB-03

Q) 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 student.

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

    void acceptDetails() {
        Scanner student = new Scanner(System.in);
        System.out.println("Enter your name");
        name = student.nextLine();
        System.out.println("Enter usn");
        usn = student.nextLine();
        System.out.println("Enter no. of subjects");
        n = student.nextInt();
        System.out.println("Enter no. of credits in each subject");
        credits = student.nextInt[];
        System.out.println("Enter no. of marks in each subject");
        marks = student.nextInt[];
    }

    for (i=0; i<n; i++) {
        System.out.println("Enter");
        credits[i] = student.nextInt();
        System.out.println("Enter marks");
        marks[i] = student.nextInt();
    }
}
```

void calcSGPA() {
 int cred = 0;
 double total = 0;
 for (i=0; i<n; i++) {
 total += (credits[i] \* marks[i]);
 cred += credits[i];
 }
 double sgpa = total / cred;
}

System.out.println("SGPA is " + sgpa);

void displayDetails() {
 System.out.println("Name: " + name);
 System.out.println("Usn: " + usn);
 System.out.println("Credits: ");
 for (i=0; i<n; i++) {
 System.out.println(credit[i] + " " + marks[i]);
 }
}

public static void main() {
 Student s = new Student();
 s.acceptDetails();
 s.displayDetails();
 s.calcSGPA();
}

Output:

Enter no of students : 2  
Enter your name: Bhoomi  
Enter your usn: CS065  
Enter no. of subjects: 3  
Enter:  
1  
Enter marks:  
89  
Enter:  
3  
Enter marks : 98  
4 78

Name: Bhoomi  
Usn: CS065  
SGPA: 8.0

Suresh  
CS303  
3  
1 98  
3 67  
4 89

*Oh Suresh*

~~1 98~~ ~~3 67~~ ~~4 89~~ ~~89~~ 100

Name: Suresh  
Usn: CS303  
SGPA: 8.0

### Code:

```
import java.util.Scanner;

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

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

        System.out.println("Enter your name:");
        name = student.nextLine();

        System.out.println("Enter your USN:");
        usn = student.nextLine();

        System.out.println("Enter number of subjects:");
        n = student.nextInt();

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

    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + calculateSGPA());
    }

    private double calculateSGPA() {
        int totalCredits = 0;
        int totalMarks = 0;
        for (int i = 0; i < n; i++) {
            totalCredits += credits[i];
            totalMarks += marks[i];
        }
        return (double) totalMarks / totalCredits;
    }
}
```

```

System.out.println("Enter USN:");
usn = student.nextLine();

System.out.println("Enter no of subjects:");
n = student.nextInt();

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

System.out.println("Enter no of credits in each sub:");
for (int i = 0; i < n; i++) {
    credits[i] = student.nextInt();
}

System.out.println("Enter no of marks in each sub:");
for (int i = 0; i < n; i++) {
    marks[i] = student.nextInt();
}

student.close();
}

public void displayDetails() {
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("Credits and Marks:");

    for (int i = 0; i < n; i++) {
        System.out.println(credits[i] + " " + marks[i]);
    }
}

public void calcSGPA() {
    int total = 0;
    int cred = 0;

    for (int i = 0; i < n; i++) {
        total += credits[i] * marks[i];
        cred += credits[i];
    }

    double sgpa = (double) total / cred;
    System.out.println("SGPA: " + sgpa);
}

```

```

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

```

## Program 3

Implement getter setter methods

### Algorithm:

15-10-24 LAB-04

(a) Create a class Book with 4 members name, author, price, num-pages. Inside a constructor set values of members. Include the methods to get, set, and `toString()` method to display.

```

import java.util.Scanner;
class Books{
    private String name;
    private String author;
    private int price;
    private int numPages;
    public Book(String name, String author, int price,
               int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    String getName(){
        return name;
    }
    String getAuthor(){
        return author;
    }
    int getPrice(){
        return price;
    }
}

```

③ Quadratic Equations

```

import java.util.Scanner;
public class QuadEquations {
    public static void main (String [] args) {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter the coefficients a,b,c");
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();
        int D = b*b - 4*a*c;
        if (a<=0){
            System.out.println ("Not quadratic");
        } else {
            if (D<0){
                System.out.println ("Roots are imaginary");
            } else if (D>0){
                double r1 = (-b + Math.sqrt(D)) / (2*a);
                double r2 = (-b - Math.sqrt(D)) / (2*a);
                System.out.println ("Roots are: ");
                System.out.println (r1 + " " + r2);
            } else {
                System.out.println ("Roots are: ");
                double r1 = -b / (2*a);
                System.out.println (r1 + " " + r1);
            }
        }
    }
}

```

O/P :	Date _____	Page _____
(i) Enter the coefficients a,b,c:		
12 1 4		
Roots are imaginary		
(ii) Enter the coefficients a,b,c:		
1 6 1		
Roots are:		
-0.17157 -5.8284		
(iii) Enter the coefficients a,b,c:		
1 2 1		
Roots are:		
-1.0 -1.0		

### Code:

```

import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

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

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

    public String getName() {
        return name;
    }

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

    public String getAuthor() {
        return author;
    }
}

```

```

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

public double getPrice() {
    return price;
}

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

public int getNumPages() {
    return numPages;
}

@Override
public String toString() {
    return "Book{" +
        "name=\"" + name + "\" +
        ", author=\"" + author + "\" +
        ", price=" + price +
        ", numPages=" + numPages +
        '}';
}
}

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

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1));
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Author: ");
            String author = scanner.next();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
        }
    }
}

```

```

        System.out.print("Number of pages: ");
        int numPages = scanner.nextInt();

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

    System.out.println("\nBook Details:");
    for (Book book : books) {
        System.out.println(book);
    }
}
}

```

# Program 4

Implement Abstract classes

**Algorithm:**

22-10-24  
LAB-05

Create an abstract class animal with methods eat & sleep. There are 3 subclasses lion, deer & tiger that extend the animal class and implement eat & sleep method differently based on behaviour

```

abstract class Animal {
    abstract void eat();
    abstract void sleep();
}

class Lion extends Animal {
    void eat() {
        System.out.println("Lion eats meat");
    }
    void sleep() {
        System.out.println("Lion sleeps");
    }
}

class Deer extends Animal {
    void eat() {
        System.out.println("Deer eats grass");
    }
}

```

void sleep() {
 System.out.println("Deer sleeps");
}

class Tiger extends Animal {
 void eat() {
 System.out.println("Tiger eats meat");
 }
 void sleep() {
 System.out.println("Tiger sleeps");
 }
}

public class main {
 public static void main (String args[])
 {
 Lion l = new Lion();
 Deer d = new Deer();
 Tiger t = new Tiger();
 t.eat();
 t.sleep();
 d.eat();
 d.sleep();
 t.eat();
 t.sleep();
 }
}

Op: Tiger eats meat
 Tiger sleeps
 Lion eats meat
 Lion sleeps
 Deer eats grass
 Deer sleeps

22/10/24  
geen  
Op: Cg

To find area of triangle, circle and rectangle using abstract class.

```

import java.util.*;
abstract class Shape {
    abstract void area(double b, double h);
}
class Triangle extends Shape {
    void area(double b, double h) {
        double A = (b*h)/2;
        System.out.println("Area of triangle = "+A);
    }
}
class Circle extends Shape {
    void area(double b, double h) {
        double A = (3.14)*b*b;
        System.out.println("Area of circle = "+A);
    }
}
class Rect extends Shape {
    void area(double b, double h) {
        double A = b*h;
        System.out.println("Area of rectangle = "+A);
    }
}

```

Date / /  
Page / /

```

public class Main {
    public static void main (String [] args) {
        System.out.println("Enter 2 dimensions");
        Scanner sc = new Scanner();
        double b = sc.nextDouble();
        double h = sc.nextDouble();

        Triangle t = new Triangle();
        Circle c = new Circle();
        Rect r = new Rect();

        t.area(b,h);
        c.area(b,h);
        r.area(b,h);
    }
}

```

Enter 2 dimensions  
12 4  
Area of triangle = 24.0  
Area of Circle = 452.159197  
Area of Rectangle = 48.0

*Seen  
gt  
24.0*

## Code:

```
***** P1 ****/
```

```
abstract class Animal {
    abstract void eatAndSleep();
}
```

```
class Lion extends Animal {
    void eatAndSleep() {
        System.out.println("Lion: Eats meat, and sleeps in a den.");
    }
}
```

```
class Deer extends Animal {
    void eatAndSleep() {
```

```

        System.out.println("Deer: Grazes on grass, and sleeps under trees.");
    }
}

class Tiger extends Animal {
    void eatAndSleep() {
        System.out.println("Tiger: Eats meat, and rests in dense forests.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal lion = new Lion();
        Animal deer = new Deer();
        Animal tiger = new Tiger();
        System.out.println("Animal Behaviors:");
        lion.eatAndSleep();
        deer.eatAndSleep();
        tiger.eatAndSleep();
    }
}

```

\*\*\*\*\* P2 \*\*\*\*\*

```

abstract class Shape {
    int x, y;

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

```

@Override

```

void printArea() {
    System.out.println("Area of Rectangle: " + (x * y));
}
}

class Triangle extends Shape {
    Triangle(int x, int y) {
        this.x = x;
        this.y = y;
    }

    @Override
    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * x * y));
    }
}

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

    @Override
    void printArea() {
        System.out.println("Area of Circle: " + (3.14 * x * x));
    }
}

public class Main {
    public static void main(String[] args) {
        Shape rectangle = new Rectangle(5, 4);
        Shape triangle = new Triangle(6, 8);
        Shape circle = new Circle(5);

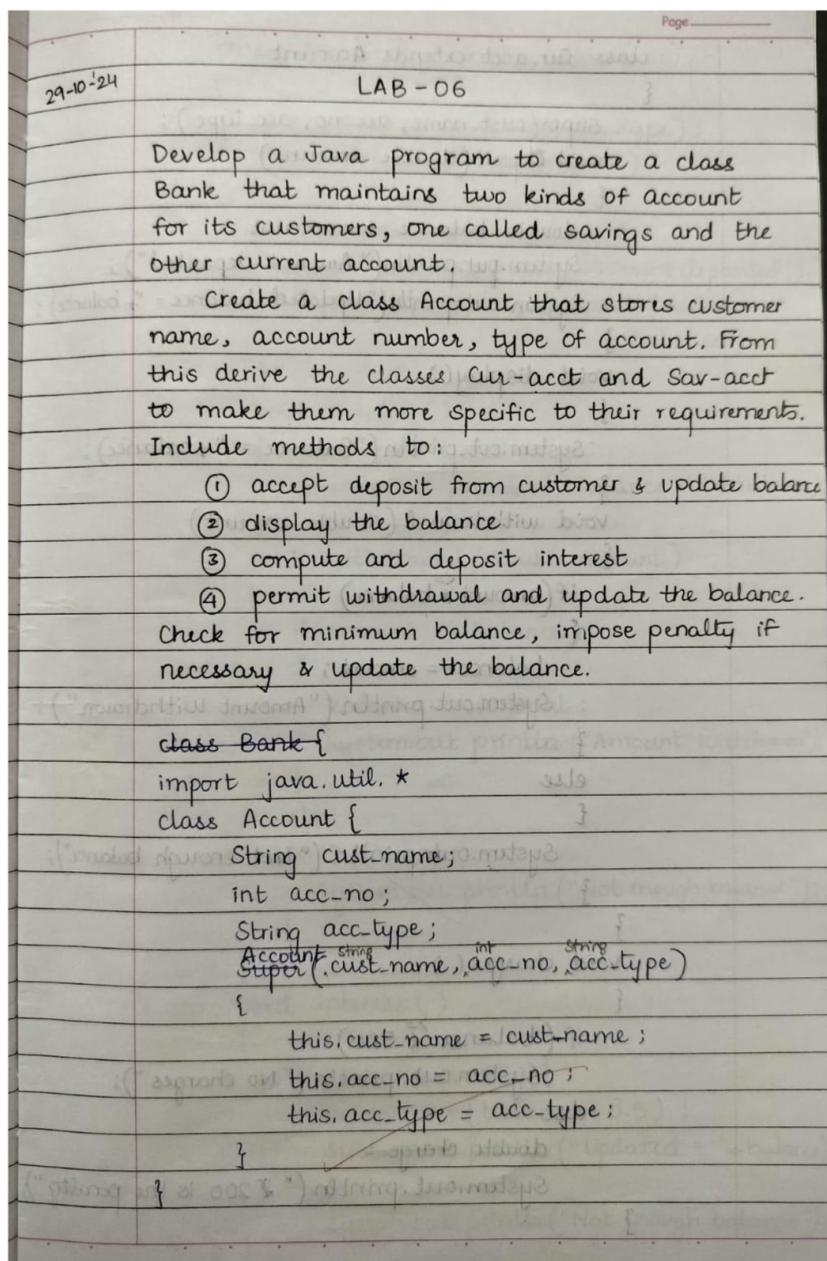
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
    }
}

```

# Program 5

Implement Bank Account

**Algorithm:**



<pre> public class Main {     public static void main(String[] args) {         System.out.println("1. Deposit \n 2. Check balance \n 3. Withdraw (from current account) \n 4. Interest (for savings account)");         System.out.print("Enter choice");         Scanner sc = new Scanner(System.in);         int ch = sc.nextInt();         System.out.println("Enter name, acc.no, acc.type");         String name = sc.nextLine();         System.out.println("Enter balance");         int acc_no = sc.nextInt();         String acc_type = sc.nextLine();         if (acc_type == "Current") {             Curr.acct c = new Curr.acct(name, acc_no, acc_type);             switch(ch) {                 case 1: System.out.println("Enter amount");                 int amount = sc.nextInt();                 c.deposit(amount);                 case 2: c.display();                 case 3: System.out.println("Enter amount");                 int amount = sc.nextInt();                 c.withdrawal(amount);                 default: System.out.println("Invalid");             }             c.charges();         }         else if (acc_type == "Savings") {             Sav.acct s = new Sav.acct(name, acc_no, acc_type);             switch(ch) {                 case 1: s.deposit(amount);                 case 2: s.display();                 case 3: s.withdrawal();                 case 4: s.interest();                 default: System.out.println("Invalid");             }         }     } } </pre>	<p><i>B</i> <i>Entered</i> <i>Proceed</i></p> <p><b>Output:</b></p> <ol style="list-style-type: none"> <li>1. Deposit</li> <li>2. Check balance</li> <li>3. Withdraw</li> <li>4. Interest</li> </ol> <p>Enter choice</p> <p>1 Enter balance</p> <p>Enter name, accno, acc type</p> <p>Bhoomi 12345 Current</p> <p>Name: Bhoomi Acc no: 12345 Acc type: Current</p> <p>Enter balance = 1000</p> <p>Enter amount: 300</p> <p>Balance = 1300</p> <p>No charges</p>
--	---

<pre> class Curr.acct extends Account {     Super(cust_name, acc_no, acc_type);     void deposit(double amount) {         double balance += amount;         System.out.println("Amount deposited");         System.out.println("Updated balance = " + balance);     }     void display() {         System.out.println("Balance = " + balance);     }     void withdrawal(double amount) {         if (amount &gt;= balance)             System.out.println("Amount withdrawn");         else             System.out.println("Not enough balance");     }     void charge() {         if (balance &lt;= 500)             System.out.println("No charges");         else             double charge = System.out.println("200 is the penalty");     } } </pre>	<pre> class Sav.acct extends Account {     double balance;     Super(cust_name, acc_no, acc_type);     void deposit(double amount) {         double balance += amount;         System.out.println("Amount deposited");         System.out.println("Updated balance = " + balance);     }     void display() {         System.out.println("Balance = " + balance);     }     void withdrawal(double amount) {         if (amount &lt;= balance)             balance -= amount;         System.out.println("Amount withdrawn");         else             System.out.println("Not enough balance");     }     void interest() {         if (balance &gt;= 500) {             balance += (balance * 8.5);             System.out.println("Updated = " + balance);         }         else             System.out.println("Not enough balance");     } } </pre>
---	---

**Code:**

```
import java.util.Scanner;

class Account {
    String name;
    int accountNumber;
    String typeOfAccount;

    Account(String name, int accountNumber, String typeOfAccount) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.typeOfAccount = typeOfAccount;
    }

    void deposit(int amount) {
    }

    void withdraw(int amount) {
    }

    void displayBalance() {
    }

    void calculateInterest() {
    }
}

class SavingsAccount extends Account {
    double interestRate;

    SavingsAccount(String name, int accountNumber, String typeOfAccount, double interestRate) {
        super(name, accountNumber, typeOfAccount);
        this.interestRate = interestRate;
    }

    @Override
    void deposit(int amount) {
        super.deposit(amount);
    }

    @Override
    void withdraw(int amount) {
        super.withdraw(amount);
    }
}
```

```

@Override
void calculateInterest() {
}

}

class CurrentAccount extends Account {
    int minimumBalance;

    CurrentAccount(String name, int accountNumber, String typeOfAccount, int minimumBalance) {
        super(name, accountNumber, typeOfAccount);
        this.minimumBalance = minimumBalance;
    }

    @Override
    void withdraw(int amount) {
        super.withdraw(amount);
    }
}

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

        String name = scanner.nextLine();
        int accountNumber = scanner.nextInt();
        String typeOfAccount = scanner.next();

        if (typeOfAccount.equalsIgnoreCase("savings")) {
            double interestRate = scanner.nextDouble();
            SavingsAccount savingsAccount = new SavingsAccount(name, accountNumber,
                    typeOfAccount, interestRate);
        } else if (typeOfAccount.equalsIgnoreCase("current")) {
            int minimumBalance = scanner.nextInt();
            CurrentAccount currentAccount = new CurrentAccount(name, accountNumber,
                    typeOfAccount, minimumBalance);
        } else {
            System.out.println("Invalid account type.");
        }

        scanner.close();
    }
}

```

# Program 6

## Implement Packages

### Algorithm:

12-11-24  
LAB - 07

Create a package CIE which has 2 classes Student and Internals with members usn, name, sem. It stores CIE marks as array of 5 subjects.

Another package SEE has class External which derives from Student. It stores an array of SEE marks.

Import 2 packages that declares final marks of n students.

```
package CIE;
class Student {
    public String usn;
    public String name;
    public int sem;
    public Student(String usn, String name, int sem) {
        this.usn = usn; this.name = name; this.sem = sem;
    }
    class Internals {
        int marks[] = new int[5];
        public Internals(int marks[]) {
            this.marks[] = marks;
        }
    }
}
```

public class FinalMarks() {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 String name = sc.nextLine();
 String usn = sc.nextLine();
 int sem = sc.nextInt();
 Internals[i] = new Internals();
 Externals[e] = new Externals();
 for(m=0; m<n; m++) {
 System.out.println("Enter name, usn, sem, see marks");
 int marks[b] = sc.nextInt();
 System.out.println("Enter CIE marks");
 for(b=0; b<5; b++) {
 int marks[b] = sc.nextInt();
 }
 E[a] = new Externals(name, usn, sem, smarks);
 if(i[a] = new Internals(marks)) {
 System.out.println("Final Marks");
 for(a=0; a<n; a++) {
 int finalC[a] = new int[5];
 for(b=0; b<5; b++) {
 finalC[b] = i.marks[a] + e.smarks[a];
 }
 System.out.println(finalC[b]);
 }
 System.out.println();
 }
 }
 }
}

10-8A

```
package SEE;
import com.CIE;
class External extends Student {
    public External(String usn, String name, int sem, int smarks[]) {
        super(usn, name, sem);
        this.smarks[] = smarks[];
    }
}
import java.util.*;
import com.CIE;
import com.SEE;
```

**Code:**

```
package CIE;

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

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

class Internals {
    int[] marks = new int[5];

    public void setMarks(int[] marks) {
        this.marks = marks;
    }
}

package SEE;

import CIE.Student;

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

    public External(String usn, String name, int sem) {
        super(usn, name, sem);
    }

    public void setMarks(int[] marks) {
        this.marks = marks;
    }
}

import CIE.*;
import SEE.*;

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

```

int n = 5;
Student[] students = new Student[n];
External[] externals = new External[n];

for (int i = 0; i < n; i++) {
    students[i] = new Student("USN" + i, "Name" + i, 5);
    externals[i] = new External("USN" + i, "Name" + i, 5);

    int[] internalMarks = {80, 75, 90, 85, 95};
    int[] externalMarks = {85, 70, 95, 80, 90};

    students[i].internals.setMarks(internalMarks);
    externals[i].setMarks(externalMarks);
}

for (int i = 0; i < n; i++) {
    System.out.println("Student " + (i + 1));
    System.out.println("USN: " + students[i].usn);
    System.out.println("Name: " + students[i].name);
    System.out.println("Semester: " + students[i].sem);

    int[] internalMarks = students[i].internals.marks;
    int[] externalMarks = externals[i].marks;

    for (int j = 0; j < 5; j++) {
        int finalMarks = (internalMarks[j] + externalMarks[j]) / 2;
        System.out.println("Course " + (j + 1) + ": " + finalMarks);
    }
    System.out.println();
}
}
}

```

# Program 7

## Implement Interfaces

## Algorithm:

M-11-24 LAB - 8 (Interface)

Program 1:  
O/p: implementation of method1

Program 2:  
O/p: Dog barks  
Dog eats bones

Program 3:  
O/p: Sedan is starting  
Sedan is driving

Program 4:  
O/p: Printing document  
Showing document preview

Program 5:

```
import java.util.*;  
interface Polygon  
{  
    public int getPerimeter();  
    public abstract int getArea();  
}  
class Triangle implements Polygon  
{  
    Scanner sc = new Scanner(System.in);  
    public int getPerimeter(){  
        int p=0;  
        int l=0;
```

```

for(i=0; i<3; i++)
{
    System.out.println("Enter side length:");
    int l = sc.nextInt();
    p+=l;
}
return p;
}

public void getArea()
{
    System.out.println("Enter side length:");
    int b = sc.nextInt();
    int h = sc.nextInt();
    int area = b*h;
    System.out.println("Area: " + area);
}

class Rectangle implements Polygon
{
    Scanner sc = new Scanner(System.in);
    public int getPerimeter()
    {
        int p=0;
        int i=0;
        for(i=0; i<4; i++)
        {
            System.out.println("Enter: ");
            int l = sc.nextInt();
            p+=l;
        }
        return p;
    }
}

public void getArea()
{
    System.out.println("Enter: ");
    int b = sc.nextInt();
    int h = sc.nextInt();
    int area = b*h;
    System.out.println("Area: " + area);
}

public class PolyMain
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        Polygon t = new Triangle();
        Polygon r = new Rectangle();
        int pr = r.getPerimeter();
        System.out.println("Perimeter: " + pr);
        int pt = t.getPerimeter();
        System.out.println("Perimeter: " + pt);
        System.out.println(r.getArea());
        System.out.println(t.getArea());
    }
}

Enter side length: 12 3 12 3
Perimeter: 30
Enter the height and breadth: 12 3
Area: 36
Enter side length: 24 4 24 4
Perimeter 52
Area: 48

```

## Code:

```

interface Polygon {
    default double getPerimeter() {
        double perimeter = 0.0;
        for (double side : getSides()) {
            perimeter += side;
        }
        return perimeter;
    }

    abstract double getArea();

    double[] getSides();
}

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[] getSides() {
    return new double[]{length, width, 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[] getSides() {
        return new double[]{2 * Math.PI * radius};
    }
}

public class Main {
    public static void main(String[] args) {
        Polygon rectangle = new Rectangle(5, 4);
        Polygon circle = new Circle(3);

        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());
        System.out.println("Rectangle Area: " + rectangle.getArea());

        System.out.println("Circle Perimeter: " + circle.getPerimeter());
        System.out.println("Circle Area: " + circle.getArea());
    }
}

```

```
}
```

# Program 8

## Implement Exception

### Algorithm:

26-1-24      LAB 9  
Exceptions

1: O/p  
ArithmeticException  $\Rightarrow$  / by zero

2: O/p  
File: test.txt is missing , Please check file name

Hi this is test file

3: O/p  
Please enter your age - Numeric value:  
10  
You are not authorized

76  
You are authorized

4: O/p:  
java.lang.ArithmaticException: / by zero  
at GFG.main (GFG.java:9)

java.lang.ArithmaticException: / by zero

java.lang.ArithmaticException: / by zero

5: O/p  
Type an integer  
23  
You typed 23  
as  
Wrapping Exception  
Exception is of type: InvalidUserInputException

6: Father Son Age Exception

```
import java.util.*;  
class Father {  
    int fage;  
    Father(int fage) {  
        this.fage = fage;  
    }  
    try {  
        if (fage > 0) System.out.println(fage);  
        else throw new Exception(" ");  
    } catch (Exception e) {  
        System.out.println("Age invalid > less than 0");  
    }  
}
```

```

class Son extends Father {
    int sage;
    Son(int sage, int fage) {
        super(fage);
        this.sage = sage;
    }
    try {
        if (fage > sage) System.out.println("Age invalid");
        else throw new Exception("Age greater");
    }
    catch (Exception e) {
        System.out.println("Age invalid => Son's age greater");
    }
}
public class AgeException {
    public static void main(String[] args) {
        Father fa = Father(-6);
        Son son = New Son(34, 12);
    }
}

Op: Age invalid => less than 0
    Age invalid => Son's age greater

```

### 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 {
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's
age.");
        }
    }
}

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

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

            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + son.age);
            System.out.println("Son's age: " + sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

## Program 9

Implement Threads

**Code:**

```

import java.util.*;
class CLG extends Thread
{
    public void run()

```

```

{
while(true)
{
System.out.println("BMS College of Engineering");
try{
Thread.sleep(10000);
}catch(InterruptedException e)
{
//error;
}
}
}

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

public class Multithreading
{
public static void main(String[] args)
{
CLG clg = new CLG();
CSE cse = new CSE();
clg.start();
cse.start();
}
}

```