

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



**LAB REPORT**  
**on**  
**Object Oriented Java Programming**  
**(23CS3PCOOJ)**

*Submitted by*

StudentName (**1BM23CS064**)

*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 **Bhavya J Makadia (1BM23CS064)**, 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.

|   |  |
|---|--|
| Geetha N<br>Assistant Professor<br>Department of CSE, BMSCE | Dr. Kavitha Sooda<br>Professor & HOD<br>Department of CSE, BMSCE |
|---|--|

# Index

| <b>Sl.<br/>No.</b> | <b>Date</b> | <b>Experiment Title</b> | <b>Page No.</b> |
|--------------------|-------------|-------------------------|-----------------|
| 1                  | 1/10/24     | Quadratic Equation      | 4-6             |
| 2                  | 8/10/24     | Calculating SGPA        | 7-11            |
| 3                  | 15/10/24    | Book Details            | 12-16           |
| 4                  | 22/10/24    | Abstract Class Shape    | 17-22           |
| 5                  | 29/10/24    | Bank Details            | 23-30           |
| 6                  | 12/11/24    | Packages                | 31-39           |
| 7                  | 19/11/24    | Interface               | 40-43           |
| 8                  | 26/11/24    | Exception Handling      | 44-48           |
| 9                  | 3/12/24     | Threads                 | 49-53           |
| 10                 | 3/12/24     | GUI – Java Swing        | 54-58           |

Github Link:

<https://github.com/Bhavya404/OOJLAB>

## **Program 1**

Implement Quadratic Equation

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

Algorithm:

```
3 import java.util.Scanner;  
  
class quad  
{  
    public static void main (String args []);  
    {  
        double a, b, c, d, r1, r2;  
        System.out.println ("Enter coefficients");  
        Scanner sc = new Scanner (System.in);  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
        c = sc.nextDouble();  
        d = b * b - 4 * a * c;  
        if (d > 0.0)  
        {  
            double r1 = (-b + Math.sqrt(d)) / (2.0 * a);  
            double r2 = (-b - Math.sqrt(d)) / (2.0 * a);  
            System.out.println ("Roots are: " + r1 + " and " + r2);  
        }  
        else if (d == 0.0)  
        {  
            double r1 = -b / (2.0 * a);  
            System.out.println ("Equal roots", + r1);  
        }  
        else  
        {  
            System.out.println ("Root don't exist");  
        }  
    }  
}
```

Q8

Output

Enter Coefficients:

3

4

5

Code:

```
import java.util.Scanner;
public class Quad
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the coefficients:");
        int a=s.nextInt();
        int b=s.nextInt();
        int c=s.nextInt();
        int d=b*b-4*a*c;
        if(d>0)
        {
            System.out.println((-b+Math.sqrt(d))/(2*a));
            System.out.println((-b-Math.sqrt(d))/(2*a));
            System.out.println("Roots are unique");
        }
        else if(d==0)
        {
            System.out.println("Roots are equal");
            System.out.println("Roots are:");
            System.out.println(-b/2*a);
        }
        else
        {
            System.out.println("No real roots");
        }
        System.out.println("Bhavya J Makadia");
        System.out.println("1BM23CS064");
    }
}
```

Output

```
D:\downloads\neww>java Quad
Enter the coefficients:
3
66
2
-0.03034488539711096
-21.969655114602887
Roots are unique
Bhavya J Makadia
1BM23CS064
```

## **Program 2**

### Calculating SGPA

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

8 | 10 | 2024

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

```
import java.util.Scanner;  
  
public class Student  
{  
    public class void()  
    {  
        string usn;  
        int name;  
        int credits;  
        int marks;  
        public void studentinfo (String usn, String name)  
        {  
            this.usn = usn;  
            this.name = name;  
        }  
        public void classandmarks (int credits, int marks)  
        {  
            scanner op = new scanner (System.in);  
            System.out.println ("Enter 5 subject's marks");  
            marks[0] = op.nextInt();  
            marks[1] = op.nextInt();  
            marks[2] = op.nextInt();  
            marks[3] = op.nextInt();  
            marks[4] = op.nextInt();  
            System.out.println ("Enter 5 credits");  
            for (int i=0; i<5; i++)  
            {  
                credit[i] = op.nextInt();  
            }  
            public void display()  
            {  
                System.out.println ("Student detail");  
                System.out.println ("student usn: " + usn);  
            }  
        }  
    }  
}
```

```
System.out.println("Student name" + name);
System.out.println("Marks:");
for(int i=0; i<5; i++)
{
```

```
    System.out.println(credits[i]);
}
```

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

```
    Student sp = new Student();
    marks[]
```

```
    sp.studentInfo("IBMA3CS000", "XYAB");
```

```
    sp.creditsAndMarks(credits, marks);
```

```
    sp.display();
```

```
    int sum;
```

```
    sum = 0;
```

```
    for(int i=0; i<5, i++)
    {
```

```
        sum += credits[i];
    }
```

```
    sum = sum / 5.00;
```

```
    System.out.println("The SGPA is: " + sum);
```

*check*

*Right*

*DP*

*Output*

Enter the 5 subject marks:

80

90

90

90

90

Code:

```
import java.util.Scanner;

public class Student
{
    String usn;
    String name;
    int[] credits = new int[5];
    int[] marks = new int[5];

    public void studentInfo(String usn, String name)
    {
        this.usn = usn;
        this.name = name;
    }

    public void enterCreditsAndMarks()
    {
        Scanner scanner = new Scanner(System.in);

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

        System.out.println("Enter credits for 5 subjects: ");
        for (int i = 0; i < 5; i++)
        {
            System.out.print("Subject " + (i + 1) + " credits: ");
            credits[i] = scanner.nextInt();
        }
    }

    public void display()
    {
        System.out.println("\nStudent Information:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Marks: ");
        for (int i = 0; i < 5; i++)
        {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }

        System.out.println("Credits: ");
        for (int i = 0; i < 5; i++)
    }
```

```

{
    System.out.println("Subject " + (i + 1) + ": " + credits[i]);
}

// Calculate GPA
int totalCredits = 0;
int weightedSum = 0;
for (int i = 0; i < 5; i++)
{
    totalCredits += credits[i];
    weightedSum += marks[i] * credits[i];
}
float gpa = (float) weightedSum / totalCredits;

System.out.println("GPA: " + gpa);
}

public static void main(String[] args)
{
    Student student = new Student();
    student.studentInfo("1BMACS001", "ABC");
    student.enterCreditsAndMarks();
    student.display();
    System.out.println("Bhavya J Makadia");
    System.out.println("1BM23CS064");

}
}

```

Output:

```

D:\downloads\neww>java Student
Enter marks for 5 subjects:
Subject 1 marks: 99
Subject 2 marks: 90
Subject 3 marks: 96
Subject 4 marks: 95
Subject 5 marks: 89
Enter credits for 5 subjects:
Subject 1 credits: 1
Subject 2 credits: 2
Subject 3 credits: 2
Subject 4 credits: 4
Subject 5 credits: 4

Student Information:
USN: 1BMACS001
Name: ABC
Marks:
Subject 1: 99
Subject 2: 90
Subject 3: 96
Subject 4: 95
Subject 5: 89
Credits:
Subject 1: 1
Subject 2: 2
Subject 3: 2
Subject 4: 4
Subject 5: 4
GPA: 92.84615

```

### **Program 3**

#### **Book Details**

Create a class Book which contains four members: name, author, price, num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

**Algorithm:**

Lab-01 import java.util.\*;

class Book

{

Scanner in = new Scanner(System.in);

String name, author;

int price, num\_pages;

Book (String n, String a, int p, int np)

{

name = n; author = a;

price = p; num\_pages = np;

}

void setDetails()

{

System.out.println("Enter Book Name:");

name = in.nextLine();

System.out.println("Enter Author's Name:");

author = in.nextLine();

System.out.println("Price of Book:");

price = in.nextInt();

System.out.println("Enter No of Pages:");

num\_pages = in.nextInt();

}

void getDetails()

{

System.out.println("Book Name: " + name);

System.out.println("Author's Name: " + author);

System.out.println("Book Price: " + price);

System.out.println("No of pages: " + num\_pages);

}

public String toString()

{

return "Book Name: " + name + "\nAuthor's Name: " + author +

"\nBook Price: " + price + "\nNo of Pages: " + num\_pages;

}

3

```
public class Lab_3
{
    public static void main (String [] args)
    {
        Scanner in = new Scanner (System.in);
        int noBook , j=0;
        String s = " ";
        System.out.println ("Enter no of Book: ");
        noBook = in.nextInt();
        Book [] bk = new Book [noBook];
        for (int i = 0 ; i < noBook ; i++)
        {
            System.out.println ("Book " + (j++));
            bk[i] = new Book (s, s, j, j);
            bk[i] = setDetails();
            j
        }
        System.out.println ("In Book Details");
        for (int k = 0 ; k < noBook ; k++)
        {
            System.out.println ();
            bk[k].getDetails();
            System.out.println (bk[k].toString());
        }
    }
}
```

3

Output

Enter number of book:

1

Book : 1

Code:

```
import java.util.*;  
  
class Book {  
    Scanner in = new Scanner(System.in);  
    String name, author;  
    int price, num_pages;  
  
    Book(String n, String a, int p, int np) {  
        name = n;  
        author = a;  
        price = p;  
        num_pages = np;  
    }  
  
    void setDetails() {  
        System.out.println("Enter Book Name: ");  
        name = in.nextLine();  
        System.out.println("Enter Author Name: ");  
        author = in.nextLine();  
        System.out.println("Price of the Book: ");  
        price = in.nextInt();  
        System.out.println("No of pages: ");  
        num_pages = in.nextInt();  
        in.nextLine();    }  
  
    void getDetails() {  
        System.out.println("Book Name: " + name);  
        System.out.println("Author Name: " + author);  
        System.out.println("Book price: " + price);  
        System.out.println("No of pages: " + num_pages);  
    }  
  
    public String toString() {  
        return "Book Name: " + name + "\nAuthor Name: " + author + "\nBook price: " + price + "\nNo  
of pages: " + num_pages;  
    }  
}  
  
public class Lab_3 {  
    public static void main(String[] args) {  
        Scanner in = new Scanner(System.in);  
        System.out.println("Enter number of books:");  
        int noBook = in.nextInt();  
        in.nextLine();  
  
        Book[] bk = new Book[noBook];
```

```
for (int i = 0; i < noBook; i++) {  
    System.out.println("Book: " + (i + 1));  
    bk[i] = new Book("", "", 0, 0);  
    bk[i].setDetails();  
}  
  
System.out.println("\nBook Details:");  
for (int k = 0; k < noBook; k++) {  
    System.out.println();  
    bk[k].getDetails();  
    System.out.println(bk[k].toString());  
}  
System.out.println("Bhavya J Makadia");  
System.out.println("1BM23CS064");  
  
}  
}
```

Output:

```
D:\downloads\neww>java Bookmain  
Enter number of books:  
1  
Book: 1  
Enter Book Name:  
Dune  
Enter Author Name:  
Frank Herbert  
Price of the Book:  
200  
No of pages:  
450  
  
Book Details:  
  
Book Name: Dune  
Author Name: Frank Herbert  
Book price: 200  
No of pages: 450  
Book Name: Dune  
Author Name: Frank Herbert  
Book price: 200  
No of pages: 450
```

#### **Program 4**

Abstract Class Shape

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

Algorithm

Q

Develop a java code to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle Triangle and circle such that each one of the classes extends the class shape. Each

class Rect

import java.util.Scanner;

abstract class Shape

{

int d1, d2;

shape (int d1, int d2)

{

this.d1 = d1;

this.d2 = d2;

}

abstract void printArea();

}

class Rectangle extends Shape

{

Rectangle (int len, int bre)

{

super(len, bre);

}

@Override

void printArea()

{

```

int area = d1 * d2;
System.out.println("Area of Rectangle : " + area);
}

```

{

```

class Triangle extends Shape
{

```

```

    Triangle (int base, int height)
    {

```

```

        super(base, height);
    }

```

@Override

```

void printAArea()
{

```

```

    double area = 0.5 * d1 * d2;

```

```

    System.out.println("Area of Triangle : " + area);
}

```

{

```

class Circle extends Shape
{

```

~~Circle (int radius)~~

2

```

    super(radius, 0);
}

```

@Override

```

void printArea()
{

```

2

```

    double area = Math.PI * d1 * d2;
}

```

```

    System.out.println("Area of circle : " + area);
}

```

{

```

public class Shapes
{
    public static void main(String[] args)
    {
        Scanner scanner sc = new Scanner(system.in);
        System.out.print("Enter length and Breath : ");
        int rectLen = sc.nextInt();
        int rectBre = sc.nextInt();
        Shape rectangle = new Rectangle(rectLen, rectBre);
        System.out.print("Enter base and height of the triangle : ");
        int triBase = sc.nextInt();
        int triHeight = sc.nextInt();
        Shape triangle = new Triangle(triBase, triHeight);
        System.out.print("Enter radius : ");
        int circleR = sc.nextInt();
        Shape circle = new Circle(circleR);
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
        scanner.close();
    }
}

```

Code:

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

```

abstract class Shape {
    int dim1, dim2;

    Shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
}
```

```
abstract void printArea();
```

```

}

class Rectangle extends Shape {
    Rectangle(int length, int breadth)
    {
        super(length, breadth);
    }

    @Override
    void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        super(base, height);
    }

    @Override
    void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0);
    }

    @Override
    void printArea()
    {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        System.out.print("Enter length and breadth of the rectangle: ");
        int rectLength = scanner.nextInt();
        int rectBreadth = scanner.nextInt();
        Shape rectangle = new Rectangle(rectLength, rectBreadth);
    }
}

```

```
System.out.print("Enter base and height of the triangle: ");
int triBase = scanner.nextInt();
int triHeight = scanner.nextInt();
Shape triangle = new Triangle(triBase, triHeight);

System.out.print("Enter radius of the circle: ");
int circleRadius = scanner.nextInt();
Shape circle = new Circle(circleRadius);

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

scanner.close();
}
}
```

Output:

```
Enter length and breadth of the rectangle: 45
12
Enter base and height of the triangle: 10
15
Enter radius of the circle: 2
Area of Rectangle: 540
Area of Triangle: 75.0
Area of Circle: 12.566370614359172
```

## **Program 5**

### **Bank Details**

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest

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

Algorithm:

Check for minimum balance and impose penalty if necessary

Class Account

{

String acc-name, acc-no, acc-type;  
double balance;

Account(String name, String no, String acc, double bal)  
{

this.acc-name = name,

this.acc-no = no;

this.acc-type = acc

this.balance = bal

}

void deposit(double amt)

{

balance = balance + amt;

System.out.print("Deposit = " + amt);

3

```
void withdraw(double amt)
```

```
{
```

```
if (amt > balance)
```

```
System.out.println ("Insufficient Balance");
```

```
else
```

```
{
```

```
balance -= amt;
```

```
System.out.println ("Withdrawal Amount = " + amt);
```

```
}
```

```
;
```

```
void checkBalance()
```

```
{
```

```
System.out.println ("Available Balance = " + balance);
```

```
}
```

```
;
```

```
class CusAcct extends Account
```

```
{
```

```
CurrAcc (String name, String no, double bal)
```

```
{
```

```
Super (name, no, "Current Account", bal);
```

```
;
```

```
void minBal()
```

```
{
```

```
if (balance < 5000)
```

```
{
```

~~System.out.println ("Min Balance in Book Account  
should be 5000\\n Service Charge of 100/-  
will be charged");~~

```
this.balance = balance - 100.0;
```

```
;
```

```
void cheque (double amt)
```

```
{
```

```
balance -= amt;
```

```
System.out.println ("Cheque Amount = " + amt);
```

```
;
```

class SavAcct extends Account

{

SavAcct (String name, String no, double bal)

{

Super (name, no, "Saving account", bal)

{

void complnd (double &) double +

{

double pri = balance;

double interest = balance \* 8 / 100.0;

pri += interest;

System.out.println("Interest Earned after 8 years for " + interest);

System.out.println("Balance after 8 years will be " + pri);

{

3

class Lab\_5

{

public static void main (String [] args)

{ CurAcct C = new CurAcct(); SavAcct S = new SavAcct();

System.out.println("Enter \n 1. To deposit in Current Account \n 2.

To withdraw from Current Account \n 3. Check balance \n 4.

To deposit in Saving Acc \n 5. Withdraw from Savings Acc \n 6.

Check balance \n 7. Check Intrest \n 8. To exist");

Scanner sc = new Scanner (System.in);

int opp = sc.nextInt();

while (opp != 0)

{

switch (opp) {

case 1: C.deposit(); break;

case 2: Scanner sc = new Scanner (System.in);

int am = sc.nextInt();

C.deposit(am); break;

case 2:

```
Scanner sc = new Scanner (System.in);
int dep = sc.nextInt();
C.withdraw(dep);
break;
```

case 3:

```
C.checkBalance()
break;
```

case 4:

```
Scanner sc = new Scanner (System.in);
int ams = sc.nextInt();
S.deposit(ams);
break;
```

case 5:

```
Scanner sc = new Scanner (System.in);
int deps = sc.nextInt(),
S.withdraw(deps);
break;
```

case 6:

```
S.checkBalance();
break;
```

case 7:

```
S.complaint()
break;
```

case 0:

```
Opp=0;
break;
```

default:

```
System.out.println("Invalid input");
```

g  
3  
g  
g  
29.10

Code:

```
import java.util.Scanner;

class Account {
    String acc_name, acc_no, acc_type;
    double balance;

    Account(String name, String no, String acc, double bal) {
        this.acc_name = name;
        this.acc_no = no;
        this.acc_type = acc;
        this.balance = bal;
    }

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

    void withdraw(double amt) {
        if (amt > balance) {
            System.out.println("Insufficient Balance");
        } else {
            balance -= amt;
            System.out.println("Withdrawal Amount = " + amt);
        }
    }

    void checkBalance() {
        System.out.println("Available Balance = " + balance);
    }
}

class CurAcct extends Account {
    CurAcct(String name, String no, double bal) {
        super(name, no, "Current Account", bal);
    }

    void minBalance() {
        if (balance < 5000) {
            System.out.println("Min Balance in Current Account should be 5000. Service charge of 100 will be charged.");
            this.balance -= 100.0;
        }
    }

    void cheque(double amt) {
        balance -= amt;
        System.out.println("Cheque Amount = " + amt);
    }
}
```

```

        }
    }

class SavAcct extends Account {
    SavAcct(String name, String no, double bal) {
        super(name, no, "Saving Account", bal);
    }

    void computeInterest(int years) {
        double initialBalance = balance;
        double interest = balance * 5 / 100.0 * years;
        initialBalance += interest;
        System.out.println("Interest earned on savings for " + years + " years is " + interest);
        System.out.println("Balance after " + years + " years will be " + initialBalance);
    }
}

public class Lab5 {
    public static void main(String[] args) {
        CurAcct currentAccount = new CurAcct("John Doe", "CA123", 10000);
        SavAcct savingAccount = new SavAcct("Jane Doe", "SA456", 15000);
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter:\n1. To deposit in Current Account\n2. To withdraw from Current Account\n3. To check balance in Current Account\n4. To deposit in Saving Account\n5. To withdraw from Saving Account\n6. To check balance in Saving Account\n7. To compute interest in Saving Account\n0. To exit");

        int option;
        do {
            System.out.print("\nChoose an option: ");
            option = sc.nextInt();

            switch (option) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double currentDeposit = sc.nextDouble();
                    currentAccount.deposit(currentDeposit);
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double currentWithdraw = sc.nextDouble();
                    currentAccount.withdraw(currentWithdraw);
                    break;
                case 3:
                    currentAccount.checkBalance();
                    break;
                case 4:
                    System.out.print("Enter amount to deposit: ");
                    double savingDeposit = sc.nextDouble();
            }
        } while (option != 0);
    }
}

```

```

savingAccount.deposit(savingDeposit);
break;
case 5:
    System.out.print("Enter amount to withdraw: ");
    double savingWithdraw = sc.nextDouble();
    savingAccount.withdraw(savingWithdraw);
    break;
case 6:
    savingAccount.checkBalance();
    break;
case 7:
    System.out.print("Enter number of years to compute interest: ");
    int years = sc.nextInt();
    savingAccount.computeInterest(years);
    break;
case 0:
    System.out.println("Exiting...");
    break;
default:
    System.out.println("Invalid input");
}
}
} while (option != 0);
}
}

```

Output:

```

Enter:
1. To deposit in Current Account
2. To withdraw from Current Account
3. To check balance in Current Account
4. To deposit in Saving Account
5. To withdraw from Saving Account
6. To check balance in Saving Account
7. To compute interest in Saving Account
0. To exit

Choose an option: 1
Enter amount to deposit: 455
Deposit = 455.0

Choose an option: 2
Enter amount to withdraw: 256
Withdrawal Amount = 256.0

Choose an option: |

```

## **Program 6**

### Packages

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

Algorithm:

package CIE;

public 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;

}

3

package CIE;

public class Internals extends Student

{

```
public int[] internalMarks;  
public Internals(String usn, String name, int sem, int[] internalMarks)
```

```
{  
    super(usn, name, sem);
```

```
    this.internalMarks = internalMarks;
```

```
}
```

```
3
```

```
package SEE;
```

```
import CIE.Student;
```

```
& public class External extends Student
```

```
{
```

```
    public int[] seeMarks;
```

```
    public Extends (String usn, String name, int sem, int[] seeMarks)
```

```
{
```

```
    super(usn, name, sem);
```

```
    this.seeMarks = seeMarks;
```

```
}
```

```
3
```

```
package
```

```
import CIE.Internals;
```

```
import CIE.Student;
```

```
import SEE.External;
```

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

```
public class Main
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
        Scanner sr = new Scanner(System.in);
```

```
        System.out.print("Enter the number of Students: ");
```

```
        int n = Scanner.nextInt();
```

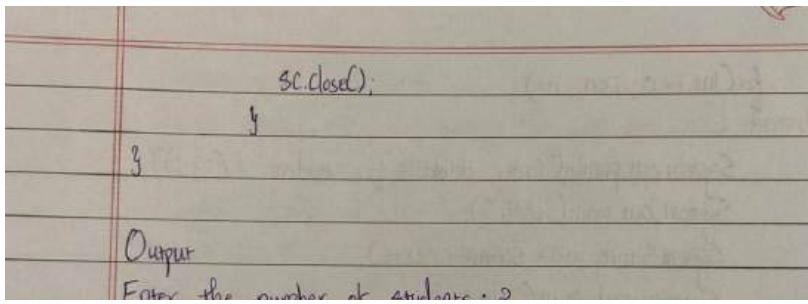
```
        Internals[] IS = new Internals[n];
```

```
        External[] ES = new External[n];
```

```

classmate
Date _____
Page _____
classmate
Date _____
Page _____
internalMarks)
for(int i=0; i<n; i++)
{
    System.out.println("Enter details for student " + (i+1));
    System.out.print("USN: ");
    String usn = scanner.nextLine();
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("Semester: ");
    int sem = scanner.nextInt();
    int[] internalMarks = new int[5];
    System.out.println("Enter internal marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        internalMark[j] = scanner.nextInt();
    }
    IS[i] = new Internal(usn, name, sem, internalMarks);
    int[] seeMarks = new int[5];
    System.out.println("Enter SEE marks for 5 courses:");
    for(int j=0; j<5; j++)
    {
        seeMark[j] = scanner.nextInt();
    }
    ES[i] = new External(usn, name, sem, seeMarks);
}
System.out.println("InFinal Mark for each student:");
for(int i=0; i<n; i++)
{
    System.out.println("Student " + (i+1) + "(" + IS[i].name + ")");
    for(int j=0; j<5; j++)
    {
        int finalMarks = ((IS[i].internalMarks[j] + ES[i].seeMarks[j])/2);
        System.out.println("Course " + (j+1) + ":" + finalMarks);
    }
    System.out.println();
}

```



Code:

```
package CIE;
```

```
public 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;
    }
}
```

```
package CIE;
import CIE.Student;
```

```
public class Internals extends Student {
    public int[] internalMarks;

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

```
package SEE;
```

```
import CIE.Student;
```

```
public class External extends Student {
    public int[] seeMarks;

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

```

        }
    }

import CIE.Internals;
import SEE.External;

import java.util.Scanner;

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

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

        Internals[] internals = new Internals[n];
        External[] externals = 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 = sc.next();

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

            System.out.print("Semester: ");
            int sem = sc.nextInt();

            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++)
            {
                internalMarks[j] = sc.nextInt();
            }
            internals[i] = new Internals(usn, name, sem, internalMarks);

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

```

```

System.out.println("\nFinal Marks for each student:");
for (int i = 0; i < n; i++)
{
    System.out.println("Student " + (i + 1) + ":");
    for (int j = 0; j < 5; j++)
    {
        int finalMarks = internals[i].internalMarks[j] + (externals[i].seeMarks[j] / 2);
        System.out.println("Course " + (j + 1) + ": " + finalMarks);
    }
}
}

```

Output:

```

Enter the number of students: 1

Enter details for student 1:
USN: esr
Name: dsfg
Semester: 3
Enter internal marks for 5 courses:
23
23
45
56
56
Enter SEE marks for 5 courses:
987
78

788
89
899

Final Marks for each student:
Student 1:
Course 1: 516
Course 2: 62
Course 3: 439
Course 4: 100
Course 5: 505

```

## **Program 7**

Interfaces

Algorithm:

Program 5

interface Shapes

{

    default public void getper(int n, int l)

{

        System.out.println("Perimeter of the regular polygon is: " + (n \* l));

    public void getarea();

}

class Rectangle implements Shapes

{

    private int a;

    private int b;

    public Rect (int a, int b)

{

        this.a = a;

        this.b = b;

}

@Override

    public void getarea()

{

        System.out.println("The area is " + (a \* b));

}

~~class Square implements Shapes~~

{

    private int a;

    private int b;

    public Square (int a, int b)

{

        this.a = a;

        this.b = b;

}

```
@Override
```

```
public void getarea()
```

```
{
```

```
    System.out.println("The area is " + (a*b));
```

```
}
```

```
}
```

```
class Triangle implements Shapes
```

```
{
```

```
    private double l, b;
```

```
    private
```

```
    public triangle(int l, int b)
```

```
{
```

```
    this.l = l;
```

```
    this.b = b;
```

```
}
```

```
@Override
```

```
public void getarea()
```

```
{
```

```
    System.out.println("The area is " + (0.5 * l * b));
```

```
}
```

```
class Main
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
        Shapes S = new Square(4, 4);
```

```
        S.getperi(4, 4);
```

```
        S.getarea();
```

```
        Shapes T = new Triangle(3, 6);
```

```
        T.getperi(3, 6);
```

```
        T.getperi();
```

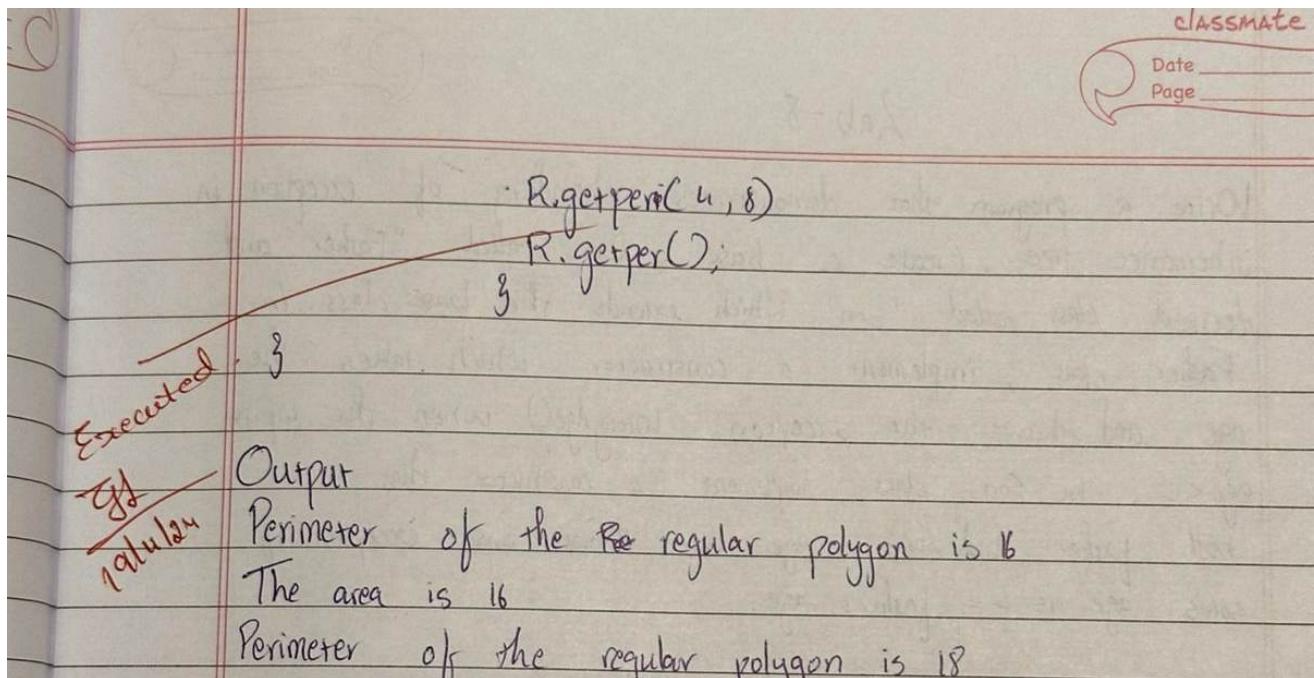
```
        Rect R = new Rect(4, 8);
```

Executed 3  
Gg  
1st line

Our  
Perio

The

Perio



Code:

```
import java.util.Scanner;

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

class Square implements Polygon {
    private double side;

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

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

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

class Triangle implements Polygon {
    private double side;

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

```

}

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

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

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

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

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

        Triangle tri = new Triangle(t);
        System.out.println("Triangle Perimeter: " + tri.getPerimeter());
        System.out.println("Triangle Area: " + tri.getArea());
        sc.close();
    }
}

```

Output:

```

Enter the length of side of square: 5
Enter the length of side of triangle: 2
Square Perimeter: 20.0
Square Area: 25.0
Triangle Perimeter: 6.0
Triangle Area: 1.7320508075688772

```

## **Program 8**

### **Exception Handling**

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

Algorithm:

## Lab - 8

Write a program that demonstrates handling of exception in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is  $\geq$  father's age.

```
class WrongAgeException extends Exception
```

```
{
```

```
    public WrongAgeException(String message)
```

```
{
```

```
    super(message);
```

```
}
```

```
3
```

```
class Father
```

```
{
```

```
    int Fatherage fa;
```

```
    public Father(int age) throws WrongAgeException
```

```
{
```

```
    if (age < 0)
```

```
{
```

throw new WrongAgeException("Age is negative");

```
}
```

~~Fatherfa = age;~~

System.out.println("Father age " + fa);

```
}
```

```
class Son extends Father
```

```
{
```

```
    int sa;
```

```
    public Son(int fa, int sa) throws WrongAgeException
```

classmate

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

exception in Father and  
class. In  
es the  
he input  
uses  
tion if

```

    i
    super(fa);
    if(sa < 0)
    {
        throw new WrongAgeException("Age is negative");
    }
    if(sa >= fa)
    {
        throw new WrongAgeException("Son's age greater than or equal to  
father's age");
    }
    this.sa = sa;
    System.out.println("Son's age is : " + sa);
}

```

3

```

public class Labpro 26
{
    public static void main(String[] args)
    {
        try
        {
            Father f = new Father(40)
            Son s = new Son(40, 15)
        }
        catch (WrongAgeException e)
        {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```

3

Code:

```

class WrongAgeException extends Exception
{

```

```

public WrongAgeException(String message)
{
    super(message);
}
}

class Father
{
    protected int age;
}

public Father(int age) throws WrongAgeException
{
    if (age < 0)
    {
        throw new WrongAgeException("Father's age is negative.");
    }
    this.age = age;
}
}

class Son extends Father
{
    public int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException
    {
        super(fatherAge);
        if (sonAge < 0)
        {
            throw new WrongAgeException("Son's age is negative.");
        }
        if (sonAge >= fatherAge)
        {
            throw new WrongAgeException("Son's age is greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class Familytree
{
    public static void main(String[] args)
    {
        try
        {
            Son son1 = new Son(40, 15);
            System.out.println("Father's age: " + son1.age);
            System.out.println("Son's age: " + son1.sonAge);
        }
    }
}

```

```
Son son2 = new Son(50, 70);
System.out.println("Father's age: " + son2.age);
System.out.println("Son's age: " + son2.sonAge);

}
catch (WrongAgeException e)
{
    System.out.println("Error: " + e.getMessage());
}
System.out.println("Bhavya J Makadia");
System.out.println("1BM23CS064");
}
}
```

Output:

```
D:\downloads\neww>java Familytree
Father's age: 40
Son's age: 15
Error: Son's age is greater than or equal to Father's age.
Bhavya J Makadia
1BM23CS064
```

## **Program 9**

Threads

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

Algorithm:

## Lab Program - 1 (Threads)

WAP which creates two threads, one thread displaying BMS  
etc college of engineering once every ten seconds and another  
display CSE once every 2 seconds

```
public class ThreadEx
```

```
{
```

```
    static class BMSSDisplayThreads extends Thread
```

```
{
```

```
        public void run()
```

```
{
```

```
            while(true)
```

```
{
```

```
                System.out.println("BMS  
Engineering College of Engineering");
```

```
            try
```

```
{
```

```
                Thread.sleep(10000);
```

```
}
```

```
            catch(InterruptedException e)
```

```
{
```

```
                System.out.println(e);
```

```
}
```

```
}
```

```
18
```

```
    class CSEDisplayThread extends Thread
```

```
{
```

```
        public void run()
```

```
{
```

```
            while(true)
```

```
{
```

```
                System.out.println("CSE");
```

```
            try
```

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

```

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

class Main
{
    public static void main(String[] args)
    {
        Thread bmsThread = new BMSDisplayThreads();
        Thread cseThread = new CSEDisplayThreads();
        bmsThread.start();
        cseThread.start();
    }
}

```

Code:

```

class ThreadEx
{
    public static class BMSDisplayThread extends Thread
    {
        public void run()
        {
            int a=0;
            while (a<5)
            {
                System.out.println("BMS College of Engineering");
                try
                {
                    Thread.sleep(200);
                }
                catch (InterruptedException e)
                {

```

```

        System.out.println(e);
    }
    a=a+1;
}
}

public static class CSEDisplayThread extends Thread
{
    public void run()
    {
        int b=0;
        while (b<5)
        {
            System.out.println("CSE");
            try
            {
                Thread.sleep(200);
            }
            catch (Exception e)
            {
                System.out.println(e);
            }
            b=b+1;
        }
    }
}
class Main
{
    public static void main(String[] args)
    {
        Thread bmsThread = new ThreadEx.BMSDisplayThread();
        Thread cseThread = new ThreadEx.CSEDDisplayThread();
        bmsThread.start();
        cseThread.start();
        System.out.println("Bhavya J Makadia");
        System.out.println("1BM23CS064");
    }
}

```

Output:

```
D:\downloads\neww>java Tmain
Bhavya J Makadia
1BM23CS064
BMS College of Engineering
CSE
BMS College of Engineering
CSE
BMS College of Engineering
CSE
CSE
BMS College of Engineering
BMS College of Engineering
CSE
```

## **Program 10**

### GUI – Java Swing

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

## Lab 2

```

2 import java.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

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

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

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

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

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

        divideButton = new JButton("Divide");

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

```

```
add(divideButton);
add(resultLabel);
add(resultField);
```

```
divideButton.addActionListener(new ActionListener() {
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            int result = num1 / num2;
            resultField.setText(String.valueOf(result));
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(DivisionApp.this, "Please enter valid integer.", "Input Error", JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ex) {
            JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero is not allowed.", "Arithmetic Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```
}
```

JOptionPane.showMessageDialog(DivisionApp.this, "Please enter valid integer.", "Input Error", JOptionPane.ERROR\_MESSAGE);

}

catch (ArithmeticException ex)

}

JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero is not allowed.", "Arithmetic Error", JOptionPane.ERROR\_MESSAGE);

}

});

}

}

Output

| Integer Division |        |
|------------------|--------|
| Num1:            | Num2:  |
| 3                | Divide |
|                  | Result |
| 2                |        |

Code:

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

public class DivisionApp extends JFrame implements ActionListener
{
    private JTextField num1Field, num2Field, resultField;
    private JButton DivisionAppdeButton;

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

JLabel nameLabel = new JLabel("Bhavya J Makadia");
JLabel usnLabel = new JLabel("1BM23CS064");

JLabel num1Label = new JLabel("Num1:");
num1Field = new JTextField(10);
JLabel num2Label = new JLabel("Num2:");
num2Field = new JTextField(10);
JLabel resultLabel = new JLabel("Result:");
resultField = new JTextField(10);
resultField.setEditable(false);
DivisionAppdeButton = new JButton("Divide");

add(nameLabel);
add(usnLabel);
add(num1Label);
add(num1Field);
add(num2Label);
add(num2Field);
add(DivisionAppdeButton);
add(resultLabel);
add(resultField);

DivisionAppdeButton.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e)
{
```

```

try
{
    int num1 = Integer.parseInt(num1Field.getText());
    int num2 = Integer.parseInt(num2Field.getText());
    int result = num1 / num2;
    resultField.setText(String.valueOf(result));
}
catch (NumberFormatException ex)
{
    JOptionPane.showMessageDialog(
        this,
        "Please enter valid integers.",
        "Input Error",
        JOptionPane.ERROR_MESSAGE
    );
}
catch (ArithmetricException ex)
{
    JOptionPane.showMessageDialog(
        this,
        "DivisionApp by zero is not allowed.",
        "Arithmetric Error",
        JOptionPane.ERROR_MESSAGE
    );
}
}

public static void main(String[] args)
{
    DivisionApp app = new DivisionApp();
    app.setVisible(true);
}
}

```

Output:

