

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

### Object Oriented Java Programming

**(23CS3PCOOJ)**

*Submitted by*

Parth Jain(1BM23CB033)

*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**

**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 **Parth Jain(1BM23CB033)**, 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 Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	12/11/24	Quadratic Equation	
2	19/11/24	Calculating SGPA	
3	26/11/24	Book Details	
4	26/11/24	Abstract Class Shape	
5	3/12/24	Bank Details	
6	3/12/24	Packages	
7	3/12/24	Interface	
8	3/12/24	Exception Handling	
9	3/12/24	Threads	
10	3/12/24	GUI – Java Swing	

Github Link: <https://github.com/parthjain21108/JAVA>

### 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:

12/11/24  
LAB-1  
Quadratic Eq.

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

```
7) import java.util.Scanner;
public class Main {
    public static void main (String [] args) {
        Scanner myobj = new Scanner (System.in);
        System.out.println ("Enter variables a,b,c for
            quadratic egn  $a^2x^2 + bx + c = 0$ ");
        double int a = myobj.nextInt();
        double int b = myobj.nextInt();
        double int c = myobj.nextInt();
        double int d = (b * b) - (4 * a * c);
        double float d1 = (float) Math.sqrt (d);
        if (d > 0)
            double x1 = (c - b + d1) / (2 * a);
            double x2 = (-b - d1) / (2 * a);
            System.out.println ("The roots are real and
                distinct : ");
            System.out.println ("x1 = " + x1);
            System.out.println ("x2 = " + x2);
        else if (d == 0)
            double r1 = -b / (2 * a);
            System.out.println ("There is one real
                root (repeated) : ");
    }
}
```

System.out.println ("x1 = x2 = " + x1);

else {

System.out.println ("There are no real solutions.");

}

### Output

Enter the variables a,b and c for the quadratic  
equation  $ax^2+bx+c=0$

1

1

1

~~There are no real solutions.~~

**Code:**

```
import java.util.Scanner;

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

        System.out.print("Enter coefficient a: ");
        double a = sc.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = sc.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = sc.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
            System.out.println("The equation has no real solutions.");
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The equation has one real solution: " + root);
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The equation has two real solutions:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        }
    }
}
```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 1>java main.java
Enter coefficient a: 1
Enter coefficient b: 1
Enter coefficient c: 1
The equation has no real solutions.
```

## 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.

### Algorithm:

13/11/24  
LAB - 2

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.

```
→ import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        Scanner myObj = new Scanner(System.in);
        System.out.print("Enter the name ");
        String name = myObj.nextLine();
        System.out.print("Enter USN ");
        int usn = myObj.nextInt();
        System.out.print("Enter no. of subjects ");
        int subjects = myObj.nextInt();

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

        System.out.println("Enter credits for subjects");
        for (int i = 0; i < subjects; i++)
        {
            credits[i] = myObj.nextInt();
        }

        System.out.println("Enter marks for subjects");
        for (int i = 0; i < subjects; i++)
        {
            marks[i] = myObj.nextInt();
        }
    }
}
```

```
double sgpa = calculateSGPA(credits, marks)
{
    System.out.println("Student Details");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("SGPA is: " + sgpa);
}
```

public static double calculateSGPA(int[] credits,  
int[] marks)

```
{  
    double totalCredits = 0;  
    double weightedSum = 0;  
    for (int i = 0; i < credits.length; i++) {  
        double gradePoint = getGradePoint(marks[i]);  
        weightedSum += gradePoint * credits[i];  
        totalCredits += credits[i];  
    }  
    return weightedSum / totalCredits;  
}
```

public static double getGradePoint(int marks)

```
{  
    if (marks >= 90)  
        return 10.0;  
    else if (marks >= 80)  
        return 9.0;  
    else if (marks >= 70)  
        return 8.0;  
}
```

return 8.0;

}

else if (marks >= 60)

{

return 7.0;

}

else if (marks >= 50)

{

return 5.0;

}

else

return 0.0;

}

}

### Output

Enter USN : 33

Enter Name : Path

Enter No of subjects : 5

Enter credits for subjects :

4

4

3

3

1

Enter marks for subjects :-

97

88

77

97

88

Student Details:

USN: 33

Name: Parth

SGPA is : 9.2667

Parth

### Code:

```
import java.io.*;
import java.util.Scanner;

public class main {
    Scanner sc = new Scanner(System.in);
    String usn;
    String name;
    int[] credit;
    int[] marks;

    public main(int numsubjects) {
        credit = new int[numsubjects];
        marks = new int[numsubjects];
    }

    void accept() {
        System.out.println("Enter USN: ");
        usn = sc.nextLine();
        System.out.println("Enter Name: ");
        name = sc.nextLine();

        for (int i = 0; i < credit.length; i++) {
            System.out.printf("Enter credit of subject %d: ", i + 1);
            credit[i] = sc.nextInt();
        }

        for (int j = 0; j < marks.length; j++) {
            System.out.printf("Enter marks of subject %d: ", j + 1);
            marks[j] = sc.nextInt();
        }
    }

    void display() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);

        for (int i = 0; i < credit.length; i++) {
            System.out.printf("Credit of subject %d = %d\n", i + 1, credit[i]);
        }

        for (int j = 0; j < marks.length; j++) {
            System.out.printf("Marks of subject %d = %d\n", j + 1, marks[j]);
        }
    }
}
```

```

void calc() {
    int totalCredit = 0;
    int wmarks = 0;

    for (int i = 0; i < credit.length; i++) {
        int mmars = (marks[i] / 10) + 1;
        totalCredit = totalCredit + credit[i];
        wmarks = wmarks + mmars * credit[i];
    }

    if (totalCredit == 0) {
        System.out.println("GPA cannot be calculated, total credits are zero.");
        return;
    }

    double gpa = (double) wmarks / totalCredit;
    System.out.println("GPA = " + gpa);
}

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

    Main s1 = new Main(numSubjects);
    sc.nextLine();

    s1.accept();
    s1.display();
    s1.calc();
}
}

```

### Output -

```

USN: 33
Name: Parth
Credit of subject 1 = 4
Credit of subject 2 = 4
Credit of subject 3 = 3
Credit of subject 4 = 3
Credit of subject 5 = 1
Marks of subject 1 = 97
Marks of subject 2 = 88
Marks of subject 3 = 77
Marks of subject 4 = 97
Marks of subject 5 = 88
GPA = 9.266666666666667

```

```
D:\COLLEGE\2nd year\github\java\Program 2>
```

### 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-3

8) 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 details of objects. Include a `toString()` method that could display complete details of the book. Java code to create n books.

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

    public Books (String name, String author, int price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public String getName() {
        return name;
    }

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

```
public string getAuthor()
{
    return author;
}

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

public int getPrice()
{
    return price;
}

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

public int getNumPages()
{
    return numPages;
}

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

public string ToString()
{
    return "Name: " + name + " Author: " + author +
        "\n Price: " + price +
        "\n Num pages: " + NumPages;
}
```

```
public class Bookstore {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter no of books :");  
    int n = sc.nextInt();  
    Books[] books = new Books[n];  
    for (int i=0; i<n; i++)  
        {  
            System.out.println("Enter name of the book :");  
            name = sc.nextLine();  
            System.out.println("Enter name of author :");  
            author = sc.nextLine();  
            System.out.println("Enter the price :");  
            price = sc.nextInt();  
            System.out.println("Enter no of pages :");  
            num_pages = sc.nextInt();  
            books[i] = new Book(name, author, price,  
                num_pages);  
        }  
    for (int i=0; i<n; i++)  
        {  
            System.out.println(books[i].toString());  
        }  
}
```

### Output

Enter the number of books : 1  
Enter the detail of BOOK 1:  
Enter name of Book : Python  
Enter name of author : xyz  
Enter price of book : 200  
Enter number of pages : 300

Printing details :-

Name of the book : Python

Name of author : xyz

price of book : 200

Number of pages : 300

26/11

### Code:

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

    public Book(String name, String author, double price, int num_pages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
    public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name=name;
    }
    public String getAuthor()
    {
        return author;
    }
    public void setAuthor(String author)
    {
        this.author=author;
    }
    public double getPrice()
    {
        return price;
    }
    public void setPrice(double price)
    {
        this.price=price;
    }
    public int getNumPages()
    {
        return num_pages;
    }
    public void setNumPages(int num_pages)
    {
```

```

        this.num_pages=num_pages;
    }
    @Override
    public String toString()
    {
        return "\nName:" + name + "\nAuthor:" + author + "\nPrice:" + price + "\nNumber of pages:"
        + num_pages;
    }
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        System.out.print("\n Enter the number of books:");
        int n = input.nextInt();
        input.nextLine();
        Book[] books =new Book[n];
        for(int i=0;i<n;i++)
        {
            System.out.print("\n Enter the name of the book:");
            String name=input.nextLine();
            System.out.print("\n Enter the author name:");
            String author = input.nextLine();
            System.out.print("\n Enter the price of the book:");
            double price = input.nextDouble();
            System.out.print("\n Enter the number of pages:");
            int num_pages = input.nextInt();
            input.nextLine();
            books[i] = new Book (name,author,price,num_pages);
        }
        for (Book book:books)
        {
            System.out.println(book.toString());
        }
        System.out.print("Chethan K S\n1BM23CS074");
        input.close();
    }
}

```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 3>java Book.java

Enter the number of books:1

Enter the name of the book:1

Enter the author name:Python

Enter the price of the book:200

Enter the number of pages:300

Name:1
Author:Python
Price:200.0
Number of pages:300
Chethan K S
1BM23CS074
D:\COLLEGE\2nd year\github\java\Program 3>
```

#### 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:

LAB - 4

8) Develop a Java program to create an abstract class named shape that contains two int and an empty class 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 method printArea() that prints area of given shape.

→ public abstract class shape  
{  
 public abstract printArea();  
 double l, b, r;  
}  
class Rectangle extends shape {  
 double l;  
 double b;  
 public Rectangle (double l, double b)  
 {  
 this.l = l;  
 this.b = b;  
 }  
 public void printArea()  
 {  
 double area = l \* b;  
 System.out.println ("The area of Rectangle : " + area);  
 }  
}

class Triangle extends Shape {

double l;

double b;

public Triangle (double l, double b)

this.l = l;

this.b = b;

public void printArea()

double area = 0.5 \* b \* l;

System.out.println ("The area of triangle: ")

class Circle extends Shape

{

double r;

public Circle (double r)

this.r = r;

public void printArea()

double area = 3.142 \* r \* r;

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

public class Area

{ public static void main (String args[])

shape = new Rectangle (10, 20);

Ques 6)

```
rec.printArea();  
shape tri = new triangle(10, 20);  
tri.printArea();  
shape cir = new circle(20);  
cir.printArea();
```

3  
3

### Output

Area of rectangle is : 200

Area of triangle is : 100

Area of circle is : 1256.8

triangle: ?

rectangle: ?

circle: ?

### Code:

```
import java.util.Scanner;

abstract class Shape {
    int l,b,r;
    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.l = length;
        this.b = breadth;
    }
    void printArea() {
        System.out.println("Area of Rectangle: " + l* b);
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.l = base;
        this.b = height;
    }
    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * l * b));
    }
}

class Circle extends Shape {
    Circle(int radius) {
        this.r = radius;
    }
    void printArea() {
        System.out.println("Area of Circle: " + (3.142 * r * r));
    }
}

public class area{
    public static void main(String[] args) {
        Rectangle r1=new Rectangle(10,20);
        r1.printArea();
        Triangle t1=new Triangle(10,20);
        t1.printArea();
    }
}
```

```
        Circle c1=new Circle(20);
        c1.printArea();
    }
}
```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 4>java area.java
Area of Rectangle: 200
Area of Triangle: 100.0
Area of Circle: 1256.8
```

## 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:

The image shows handwritten code on lined paper. The code defines three classes: Account, Cur-acct, and Sav-acct. The Account class has attributes Name, acc-no, and balance. It includes methods for deposit, display, and withdraw. The Cur-acct class extends Account and adds a calc-interest method. The Sav-acct class extends Account and adds a calc-interest method with a different implementation. There are also some notes on the left side of the page.

```
class account
{
    String Name;
    int acc-no;
    double balance;
    public Account (String Name, int acc-no, double balance)
    {
        this.Name = Name;
        this.acc-no = acc-no;
        this.balance = balance;
    }
    public void deposit (double amount)
    {
        balance += amount;
        System.out.println ("Deposit successful, current
balance = " + balance);
    }
    public void display()
    {
        System.out.println ("Account balance " + balance);
    }
}
class sav-acct extends account
{
    public sav-acct (String Name, int acc-no, double balance)
    {
        super (Name, acc-no, balance);
    }
    public void calc-interest (double amount)
    {
        double interest-rate = 0.05;
        double interest = interest-rate * balance;
        balance += interest;
    }
}
```

System.out.println("Interest is applied to  
balance = " + balance);

3  
public void withdraw(double amount)

{  
if (amount >= balance)

balance -= amount;

System.out.println("The amount" + amount

+ "has been withdrawn.

Balance is =" + balance);

4

else System.out.println("Insufficient balance");

5

System.out.println("The balance is insufficient");

6  
public class Curr-Acc extends Account {

public void currAcc(String Name, int acc-no,  
double balance) {

super(Name, acc-no, balance);

7

int minBal = 1000;

int penalty = 50;

public void withdraw(double amount) {

if (amount >= minBalance)

balance -= amount;

balance -= penalty;

System.out.println("Penalty applied

Current balance =" + balance);

is applied to  
balance,

amount)

t" + amount  
withdrawn.  
" + balance).

is insufficient;

at acc-no,

"balance?

else

balance -= amount;

System.out.println ("withdrawal successful  
current Balance = " + balance);

class Bank {

public static void main (String args[])

{

Scanner sc = new Scanner (System.in);

int x;

int i=0;

sav-acc sav = new Sav-Acc ("Path", 203, 1000)

curr-acc cur = new Cur-Acc ("Jain", 103, 2000)

while (i < 0)

{

System.out.println ("Enter 1 for saving & 2  
for current acc");

x = sc.nextInt();

if (x == 1)

{

System.out.println ("1 → balance, 2 → withdraw,  
3 → deposit, 4 → interest");

x = sc.nextInt();

if (x == 1)

{

sc.display();

{

else if (x == 2)

{

System.out.println ("Enter amt :");

int y = sc.nextInt();

sc.withdraw(y);

```
else if (x == 3)
{
    system.out.println ("Enter amount");
    int y = sc.nextInt();
    sav_acc.deposit (int y);
}

else if (x == 4)
{
    sc.calInterest ();
}

else
{
    break;
}

else if (x == 2)
{
    system.out.println ("1 → balance 2 → withdraw 3 → deposit");
    int y = sc.nextInt();
    if (ly == 1)
        sav_acc.display ();
    else if (ly == 2)
        {
            system.out.println ("Enter amount");
            int zy = sc.nextInt();
            sav_acc.withdraw (int zy);
        }
    else if (ly == 3)
        system.out.println ("Enter amount");
        int z = sc.nextInt();
```

sav.acc \* deposit (intg);

3

else

2

break;

4

3

04/3/12

### Output

Enter 1 → Saving acc 2 → Current acc

1

Enter 1 → Balance; 2 → Deposit; 3 → Withdraw 4 → Interest

2

Enter amount:

2000

Deposit successful current balance = 3000.0

Enter 1 → Saving acc 2 → Current acc

1

Enter 1 → Balance; 2 → Deposit; 3 → Withdraw; 4 → Interest

4

Interest applied current balance = 3150.0

**Code:**

```
import java.util.Scanner;

class Account{

    String Name;
    int accno;
    double balance;

    public Account(String Name,int accno,double balance){
        this.Name=Name;
        this.accno=accno;
        this.balance=balance;
    }

    public void deposite(double amount){
        balance+=amount;
        System.out.println("Deposite successfull current balance = "+balance);
    }

    public void display(){
        System.out.println("Current balance = "+balance);
    }
}

class Savacct extends Account{

    public Savacct(String Name,int accno,double balance){
        super(Name,accno,balance);
    }

    public void calinter(){
        double interrate=0.05;
        double interest =interrate*balance;
        balance+=interest;
        System.out.println("Interest applied Current balance = "+balance);
    }

    public void withdraw(double amount){
        if(amount<=balance){
            balance-=amount;
            System.out.println("Withdraw successfull ; Current Balance = "+balance);
        }
    }
}
```

```

        }
    else{
        System.out.println("Insufficient balance");
    }
}
class Curacct extends Account{

public Curacct(String Name,int accno,double balance){
super(Name,accno,balance);
}

int minbalance=1000;
int pan=50;

public void withdraw(double amount){
if(amount<balance){
    balance-=amount;
    System.out.println("Withdraw successfull ; Current Balance = "+balance);
}
else{
    balance-=amount;
    balance-=pan;
    System.out.println("Low balance warring Panalty of 50 applied ; Current
Balance = "+balance);
}
}
}

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

Savacct sav=new Savacct("Parth",203,1000);
Curacct cur=new Curacct("Jain",203,1000);

int i=0;
int x;
int y;
double z;

while(i==0){
    System.out.println("Enter 1-> Saving acc ; 2 -> Current acc");
    x=sc.nextInt();
    if(x==1){
        System.out.println("Enter 1-> Balance ; 2-> Deposite ; 3 -> Withdraw ;
4-> Interest ; other to exit");
    }
}
}
}

```

```

y=sc.nextInt();
if(y==1){
    sav.display();
}
else if(y==2){
    System.out.println("Enter the amount : ");
    z=sc.nextInt();
    sav.deposite(z);
}
else if(y==3){
    System.out.println("Enter the amount : ");
    z=sc.nextInt();
    sav.withdraw(z);
}
else if(y==4){
    sav.calinter();
}
else{
    System.out.println("exiting");
    break;
}
}

else if(x==2){
    System.out.println("Enter 1-> Balance ; 2-> Deposite ; 3 -> Withdraw;
other to exit");
    y=sc.nextInt();
    if(y==1){
        cur.display();
    }
    else if(y==2){
        System.out.println("Enter the amount : ");
        z=sc.nextInt();
        cur.deposite(z);
    }
    else if(y==3){
        System.out.println("Enter the amount : ");
        z=sc.nextInt();
        cur.withdraw(z);
    }
    else{
        System.out.println("exiting");
        break;
    }
}
else{

```

```
        break;  
    }  
  
}  
}  
}
```

**Output:**

```
Enter 1-> Saving acc ; 2 -> Current acc  
1  
Enter 1-> Balance ; 2-> Deposite ; 3 -> Withdraw ; 4-> Interest ; other to exit  
2  
Enter the amount :  
2000  
Deposite successfull current balance = 3000.0  
Enter 1-> Saving acc ; 2 -> Current acc  
1  
Enter 1-> Balance ; 2-> Deposite ; 3 -> Withdraw ; 4-> Interest ; other to exit  
4  
Interest applied Current balance = 3150.0
```

## **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:**

### LAB-6

- e) Create a package CIE which has two classes - Personal and Internals. This class Personal has members like USN, Name and sem. This class Internals has an array that stores the marks made by student in 5 courses of current sem at student. Import package in a file marks of n students.

package CIE;

import java.util.Scanner;

public class Student

{

protected String USN = new String();

protected String name = new String();

protected int sem;

3

public void input()

{

Scanner s = new Scanner(System.in);

System.out.println("Enter USN, name, sem:");

USN = s.next();

sem = s.nextInt();

7

public void disp()

{

System.out.println("Student details");

System.out.println("USN" + USN);

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

System.out.println("Sem" + sem);

3

internal → C:\Users\asus\OneDrive

```
package cse;
import java.util.Scanner;
public class Internal extends Student
```

```
{ protected int marks = new int[5];
  public void inputmarks()
```

```
Scanner s = new Scanner(System.in);
System.out.println("Enter marks of 5 courses");
for(int i=0; i<5; i++)
{
```

marks[i] = s.nextInt()

External

```
package SEE
```

```
import cse.Internal;
```

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

```
public class External extends Internal
```

```
{ protected int seeMarks[];
```

```
protected int finalMarks[];
```

```
public External(int[] seeMarks, int[] finalMarks)
```

```
{
```

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

```
finalMarks = new int[5];
```

```
public void inputmain()
```

```
{  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter 5 numbers");  
    for (int i=0; i<5; i++)
```

```
{  
    scemain[i] = s.nextInt();  
}
```

```
3  
public void finalmain()
```

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

```
{  
    finalmain[i] = scemain[i] + main[i];  
}
```

```
4  
public void display final main()
```

```
display;
```

```
System.out.print("Enter main");
```

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

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

```
y
```

~~main.java~~

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

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

```
class main
```

```
{
```

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

```
{  
    int n;
```

```
public void inputmain()
```

```
{  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter 5 numbers");  
    for (int i=0; i<5; i++)
```

```
{  
    scemain[i] = s.nextInt();  
}
```

```
3  
public void finalmain()
```

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

```
{  
    finalmain[i] = scemain[i] + main[i];  
}
```

```
4  
public void display final main()
```

```
display;
```

```
System.out.print("Enter main");
```

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

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

```
y
```

~~main.java~~

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

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

```
class main
```

```
{
```

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

```
{  
    int n;
```

2 void nuorange(int a) throw my exception

{

if (a < 0) {

throw new exception();

}

else

{

System.out.println("Valid age");

}

3 class son extends Father

{

int sage;

son (int page, int sage)

{

super(sage)

this.sage = sage;

}

void check age () throws my except

{

if (sage > page)

throw new my except();

}

else

{

System.out.println("Valid son age") + sage);

}

}

### Code:

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

public class Internals extends Student {
    protected int[] cieMarks;
    int n = 5;

    public Internals(String usn, String name) {
        super();
        this.usn = usn;
        this.name = name;
        this.cieMarks = new int[n];
    }

    public void get_CIE_marks() {
        Scanner s2 = new Scanner(System.in);
        System.out.println("Enter Marks for CIE");
        for (int i = 0; i < n; i++) {
            cieMarks[i] = s2.nextInt();
        }
    }

    public void display_CIE_marks() {
        System.out.println("CIE Marks for the student " + name + ":");
        for (int i = 0; i < n; i++) {
            System.out.println("Subject " + (i + 1) + ": " + cieMarks[i]);
        }
    }
}

package CIE;
import java.util.Scanner;

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

    public Student() {
        this.usn = "";
        this.name = "";
        this.sem = 0;
    }

    public void get_details() {
        Scanner s1 = new Scanner(System.in);
```

```

System.out.println("Enter USN NAME SEM");
usn = s1.nextLine();
name = s1.nextLine();
sem = s1.nextInt();
}

public void display_details() {
    System.out.println("Student Details");
    System.out.println("USN - " + usn);
    System.out.println("NAME - " + name);
    System.out.println("Sem - " + sem);
}
}

public static void main(String[] args) {
    Student student = new Student();
    student.get_details();
    student.display_details();
}
}

```

```

package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int[] seeMarks;
    protected int[] finalMarks;
    int x = 5;

    public Externals(String usn, String name) {
        super(usn, name);
        this.seeMarks = new int[x];
        this.finalMarks = new int[x];
    }

    public void get_SEE_marks() {
        Scanner s3 = new Scanner(System.in);
        System.out.println("Enter Marks for SEE");
        for (int i = 0; i < x; i++) {
            seeMarks[i] = s3.nextInt();
        }
    }

    public void calc_final_marks() {
        for (int i = 0; i < x; i++) {
            finalMarks[i] = cieMarks[i] + seeMarks[i];
        }
    }
}

```

```

}

public void display_final_marks() {
    display_details();
    System.out.println("Final Marks for each subject: ");
    for (int i = 0; i < x; i++) {
        System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
    }
}

public static void main(String[] args) {
    Externals externals = new Externals("33", "Parth");
    externals.get_details();
    externals.get_CIE_marks();
    externals.get_SEE_marks();
    externals.calc_final_marks();
    externals.display_final_marks();
}
}

import SEE.Externals;
import java.util.Scanner;

class Run {
    public static void main(String args[]) {
        int num;
        Scanner s4 = new Scanner(System.in);
        System.out.println("Enter no of students");
        num = s4.nextInt();
        Externals[] e = new Externals[num];

        for (int i = 0; i < num; i++) {
            Scanner input = new Scanner(System.in);
            System.out.println("Enter USN NAME SEM for student " + (i + 1));
            String usn = input.nextLine();
            String name = input.nextLine();
            int sem = input.nextInt();

            e[i] = new Externals(usn, name);
            e[i].input();
            e[i].disp();
            e[i].inpcemarks();
            e[i].inpseemarks();
            e[i].calc_final_marks();
            e[i].display_final_marks();
        }
    }
}

```

}

**OUTPUT:**

**Enter USN NAME SEM**

33

Parth

1

**Enter Marks for CIE**

23

34

45

45

40

**Enter Marks for SEE**

45

34

45

34

45

**Student Details**

**USN – 33**

**NAME – Parth**

**Sem – 1**

## Program 7

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:

The handwritten algorithm is organized into two main sections separated by a horizontal line. The top section is for the Father class and the bottom section is for the Son class.

**Father Class:**

```
1 void wrongage(int a) throw myException
2   if (a < 0)
3     throw newException()
4 else
5   System.out.println("Valid age");
```

**Son Class:**

```
1 class son extends father
2   int sage;
3   son (int fage, int sage)
4     super(fage)
5     this.sage = sage;
6   void check age () throws myExcept
7     if (sage >= fage)
8       throw new myExcept();
9 else
10   System.out.println ("Valid son age "+sage);
```

2 void nuorange(int a) throw my exception

{

if (a < 0) {

throw new exception();

}

else

{

System.out.println("Valid age");

}

3 class son extends Father

{

int sage;

son (int page, int sage)

{

super(sage)

this.sage = sage;

}

void check age () throws my except

{

if (sage > page)

throw new my except();

}

else

{

System.out.println("Valid son age") + sage);

}

}

```
class main
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        int age1, age2;
        System.out.println ("Enter father age");
        age1 = s.nextInt();
        Father f = new Father (age1);
        try
        {
            f.munage (age1, age2);
        }
        catch (myexception e)
        {
            System.out.println ("Caught " + e);
            System.out.println ("Enter age");
            age2 = s.nextInt();
            Son s = new Son (age1, age2);
            try
            {
                s1.munage ();
            }
            catch (myexception e)
            {
                System.out.println ("Caught " + e);
            }
        }
    }
}
```

### Output

Enter father's age:

60

Valid age 60

Enter son's Age:

40

Valid, son's age is 40

### Code:

```
import java.util.Scanner;

class MyException extends Exception {
    public String toString() {
        return "Age should be greater than zero";
    }
}

class MyExcep extends Exception {
    public String toString() {
        return "Father's age should be greater than son's age";
    }
}

class Father {
    int fage;

    Father(int fage) {
        this.fage = fage;
    }

    void wrongAge(int a) throws MyException {
        if (a < 0) {
            throw new MyException();
        } else {
            System.out.println("Valid age: " + a);
        }
    }
}

class Son extends Father {
    int sage;

    Son(int fage, int sage) {
        super(fage);
        this.sage = sage;
    }

    void checkAge() throws MyExcep {
        if (sage > fage) {
            throw new MyExcep();
        } else {
            System.out.println("Valid son's age: " + sage);
        }
    }
}
```

```
}
```

```
class Main {
```

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

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

```
        int age1, age2;
```

```

        System.out.print("Enter father's age: ");
```

```
        age1 = s.nextInt();
```

```

        Father f = new Father(age1);
```

```
        try {
```

```
            f.wrongAge(age1);
```

```
        } catch (MyException e) {
```

```
            System.out.println("Caught: " + e);
```

```
        }
```

```

        System.out.print("Enter son's age: ");
```

```
        age2 = s.nextInt();
```

```

        Son s1 = new Son(age1, age2);
```

```
        try {
```

```
            s1.checkAge();
```

```
        } catch (MyExcep e) {
```

```
            System.out.println("Caught: " + e);
```

```
        }
```

```
    }
```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 7>java Main.java
Enter father's age: 60
Valid age: 60
Enter son's age: 40
Valid son's age: 40
```

### Program 8

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:

Q) LAB 8

e) WAP which creates two threads BMSCE out of every 10 seconds and CSE every two seconds.

```
class BMSCE extends Thread  
{  
    public void run()  
    {  
        for(int i=1; i<=2; i++)  
        {  
            System.out.println("BMSCE");  
            try  
            {  
                Thread.sleep(10*1000);  
            }  
            catch(InterruptedException e)  
            {}  
        }  
    }  
  
    class CSE extends Thread  
{  
        public void main()  
        {  
            for (int i=1; i<10; i++)  
            {  
                System.out.println("CSE");  
                try  
                {  
                    Thread.sleep(2*1000);  
                }  
            }  
        }  
    }  
}
```

(with interrupted exception)

1

2

3

public class Main

4

public static void main (String args[])

BMSCE b1 = new BMSCE();

b1.start();

CSE c1 = new CSE();

c1.start();

5

6

Output

BMSCE

CSE

CSE

CSE

~~BMSCE~~

~~CSE~~

~~CSE~~

~~BMSCE~~

**CODE:**

```
class BMSCE extends Thread {  
    public void run() {  
        for (int i = 1; i <= 2; i++) {  
            try {  
                System.out.println("BMSCE ");  
                Thread.sleep(10 * 1000);  
            } catch (InterruptedException e) {  
            }  
        }  
    }  
  
    class CSE extends Thread {  
        public void run() {  
            for (int i = 1; i <= 10; i++) {  
                try {  
                    System.out.println("CSE ");  
                    Thread.sleep(2 * 1000);  
                } catch (InterruptedException e) {  
                }  
            }  
        }  
    }  
  
    public class Main {  
        public static void main(String args[]) {  
            BMSCE b1 = new BMSCE();  
            b1.start();  
            CSE c1 = new CSE();  
            c1.start();  
        }  
    }  
}
```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 8>java Main  
BMSCE  
CSE  
CSE  
CSE  
CSE  
CSE  
BMSCE  
CSE
```

## Program 9

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 ArithmeticException. Display the exception in a message dialog box.

### Algorithm:

LAB 9

8 NAP to create interfaces to perform integer div.  
The user enters two Num1 & Num2. The div. of Num1 and Num2 to be display when divide button is clicked. If Num1 or Num2 not entered or Num2 = 0 throw the error.

```
import java.util.*;  
import java.awt.*;  
import java.awt.event.*;  
class SwingDemo  
{  
    JFrame jfm = new JFrame("Divide App");  
    jfm.setSize(275, 150);  
    jfm.setLayout(new FlowLayout());  
    jfm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    JLabel jlab = new JLabel("Enter the divisor  
and dividend:");  
    JTextField aifj = new JTextField(8);  
    JTextField bifj = new JTextField(8);  
    JButton button = new JButton("Calculate");  
    JLabel cu = new JLabel();  
    JLabel alab = new JLabel();  
    Label anslab = new Label();  
Label blab = new Label();  
    jfm.add(cu);  
    jfm.add(jlab);  
    jfm.add(aifj);  
    jfm.add(bifj);  
    jfm.add(button);  
    jfm.add(alab);
```

```
if(jum.add(blab));  
if(jum.add(anslab));  
Action listener s = new Action listener()  
{  
    public void action performed(ActionEvent evt)  
    {  
        System.out.println("Action event from a text field");  
    }  
}
```

```
ajq.add Action listener(s);  
bjq.add Action listener(s);  
button.add Action listener(new Action listener()  
{  
    public void action performed(ActionEvent evt)  
    {  
        try  
        {  
            int a = Integer.parseInt(ajq.getText());  
            int b = Integer.parseInt(bj.gettext());  
            int ans = a/b;  
            alab.setText("A = " + a);  
            blab.setText("B = " + b);  
            anslab.setText("Ans = " + ans);  
            em.setText("");  
        }  
        catch(NumberFormatException e)  
        {  
            alab.setText("");  
            blab.setText("");  
            anslab.setText("");  
            em.setText("Enter only integers!");  
        }  
    }  
}}
```

catch (ArithmeticException e)

{

    alab.setText("11");

    bbb.setText("11");

    cc.setText("B should not be zero");

    jf.setVisible(true);

}

public static void main(String[] args)

{

    System.out.println("USN: 33 In Name: Pauth\n");

    swingUtilities.invokeLater(new Runnable())

{

    public void run()

    new SWINGDemo();

}

}

Output

USN: 33

Name: Pauth

~~Calculator App~~ -  X

Enter divisor and dividend

25	5
----	---

Calculate A=25 B=5

Ans = 5

## **CODE:**

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        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(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        ActionListener l = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                System.out.println("Action event from a text field");
            }
        };

        ajtf.addActionListener(l);
        bjtf.addActionListener(l);

        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;

```

```

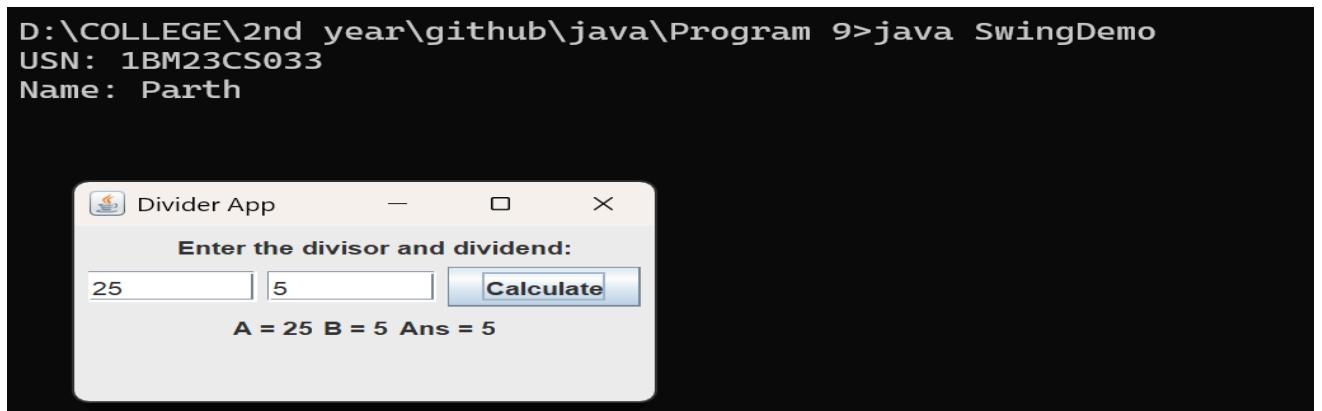
        alab.setText("A = " + a);
        blab.setText("B = " + b);
        anslab.setText("Ans = " + ans);
        err.setText("");
    } catch (NumberFormatException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
    }
}
});

jfrm.setVisible(true);
}

public static void main(String[] args) {
    System.out.println("USN: 1BM23CS033\nName: Parth\n");
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}

```

**Output:**



## **Program 10**

Program 10 Demonstrate Inter process Communication and deadlock

### **Algorithm:**

## LAB 10

e) demonstrate inter process communication and deadlock

class A

{

synchronized void foo(B b)

{

String name = Thread.currentThread().getName();

System.out.println(name + " called A::foo");

try

{

Thread.sleep(1000);

}

catch (Exception e)

{

System.out.println("A interrupted");

}

System.out.println(name + " trying to call B::last");

b.last();

}

synchronized void last()

{

System.out.println("Inside A::last");

}

Class B

synchronized void bar(A a)

{

String name = Thread.currentThread().getName();

system.out.println("name + "entered B::bar");

try {

Thread.sleep(1000);

}

catch (Exception e)

{

system.out.println("B interrupted");

}

system.out.println(name + " trying to call " + last());

a.last();

}

synchronized void last()

{

system.out.println("Inside B.last");

}

class deadlock implements Runnable

{

A a = new A();

B b = new B();

deadlock()

{

Thread currentThread.getName("Main Thread");

Thread t = new Thread(this, "Raising Thread");

t.start();

a.foo(b);

System.out.println("Back in main thread");

public void run()

{

b  
1

3  
public

{  
5  
1  
2

3  
Y

4  
out

5  
Main

Run

6  
Ra

Ra

7  
DEAD

b.bar(a);

System.out.println("Bar in other thread");

3 public static void main (String [] args)

System.out.println("Deadlock");  
new deadlock();

4

y

x

x

ell + last());

### Output

~~Deadlock~~

Main thread entered a foo

Raising thread entered b.bar

~~Raising thread trying to call A.last().~~

~~Raising thread trying to call B.last().~~

~~deadlock~~

x

x

### Code:

```
class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }

    synchronized void last() {
        System.out.println("Inside A.last");
    }
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    synchronized void last() {
        System.out.println("Inside B.last");
    }
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
    }
}
```

```
a.foo(b);
System.out.println("Back in main thread");
}

public void run() {
    b.bar(a);
    System.out.println("Back in other thread");
}

public static void main(String[] args) {
    new Deadlock();
}
```

**Output:**

```
D:\COLLEGE\2nd year\github\java\Program 10>java Deadlock.java
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
|
```