

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## OBJECT ORIENTED JAVA PROGRAMMING

*Submitted by*

**SHASHANK GOWDA L(1BM23CS311)**

*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-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**" carried out by **SHASHANK GOWDA L(1BM23CS311)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

**Dr. Nandhini Vineeth**

Associate Professor,  
Department of CSE,  
BMSCE, Bengaluru

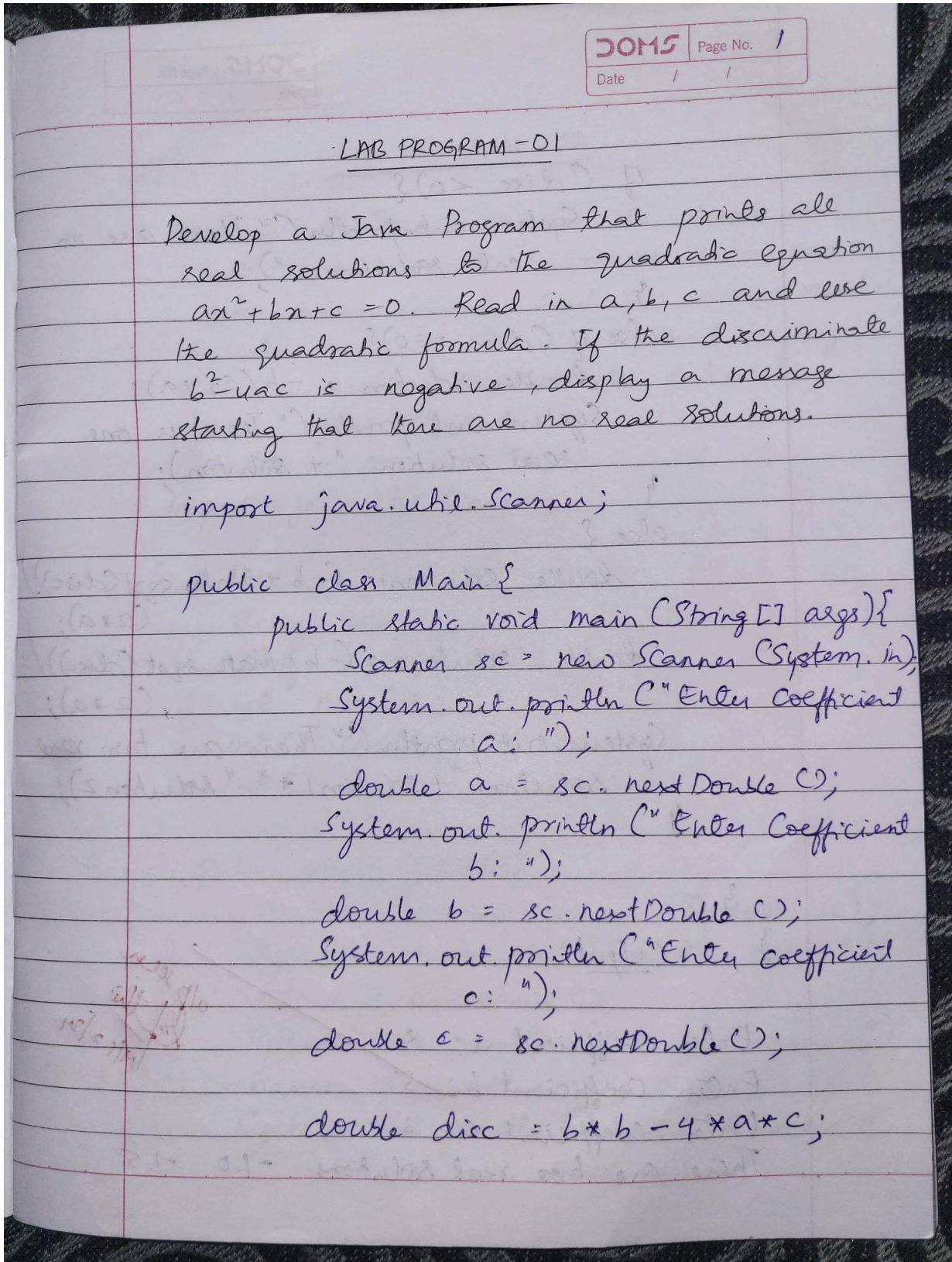
**Dr. Kavitha Sooda**

Professor and Head,  
Department of CSE  
BMSCE, Bengaluru

## **INDEX**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1		Quadratic equation	1-3
2		SGPA calculation	4-9
3		Display book details	10-16
4		Abstract class demonstration	17-22
5		Bank account management	23-33
6		Packages demonstration	34-41
7		Exception handling demonstration	42-48
8		Multithreading demonstration	49-51
9		Creating an user interface	52-53
10		Inter Process communication and deadlock	54

**Question 1: 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.**



if (disc < 0) {

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

}

else if (disc == 0) {

double solution = -b / (2 \* a);

System.out.println ("There is one  
real solution: " + solution);

}

else {

double solution1 = (-b + Math.sqrt(disc)) /  
(2 \* a);

double solution2 = (-b - Math.sqrt(disc)) /  
(2 \* a);

System.out.println ("There are two real  
solutions " + solution1 + " " + solution2);

}

}

} Output:

Enter coefficient a: 2

Enter coefficient b: 5

Enter coefficient c: 3

There are two real solutions -1.0 -1.5

old seen  
old this  
last 2 fm

```
import java.util.Scanner;

public class javalab1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

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

        if (discriminant < 0) {
            System.out.println("No real solutions.");
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("One real solution: " + root);
        } else {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two real solutions: " + root1 + " and " + root2);
        }
    }
}
```

```
[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab1
Enter coefficient a: 3
Enter coefficient b: 7
Enter coefficient c: 4
Two real solutions: -1.0 and -1.3333333333333333
shashank@Shashanks-MacBook-Air 00J_Lab % ]
```

**Question 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 of a student.**

DOMS Page No. 3

Date / /

### LAB PROGRAM - 02

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.

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

```
class Student {
```

```
    String usn;
```

```
    String name;
```

```
    int numSub;
```

```
    int [] credit;
```

```
    int [] marks;
```

```
    public void getd(){
```

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

```
        System.out.println ("Enter USN:");
```

~~usn = sc.nextLine();~~~~System.out.println ("Enter name:");~~~~name = sc.nextLine();~~~~System.out.println ("Enter number of subjects:");~~

```
numSub = sc.nextInt();
```

```
Credits = new int [numSub];
```

```
Marks = new int [numSub];
```

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

```
System.out.println ("Enter credit
```

```
for subject " + (i+1) + ":" );
```

```
Credits [i] = sc.nextInt();
```

```
System.out.println (" Enter marks
```

```
for subject " + (i+1) + ":" );
```

```
Marks [i] = sc.nextInt();
```

```
}
```

```
}
```

```
void display () {
```

```
System.out.println (" \n Student Details : ")
```

```
System.out.println (" USN: " + usn);
```

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

~~```
System.out.println ("Subject-wise Credit  
and Marks: ");
```~~~~```
for (int i=0; i< numSub ; i++) {
```~~~~```
System.out.println (" Subject " + C[i]
```~~~~```
+ ":" : Credits = " + Credits [i] + "
```~~~~```
" , Marks = " + Marks [i] ) );
```~~~~```
}
```~~~~```
L
```~~

```
double SGPA();
```

```
int tot_credits = 0;
```

```
int w_gradepts = 0;
```

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

```
    int gradept = getGP(marks[i]);
```

```
    w_gradepts += gradept * credits[i];
```

```
    tot_credits += credits[i];
```

```
}
```

```
return (double) w_gradepts / tot_credits;
```

```
int getGP (int marks) {
```

```
    if (marks >= 90) return 10;
```

```
    else if (marks >= 80) return 9;
```

```
    else if (marks >= 70) return 8;
```

```
    else if (marks >= 60) return 7;
```

~~```
    else if (marks >= 50) return 6;
```~~~~```
    else if (marks >= 40) return 5;
```~~~~```
    else return 0;
```~~

```
}
```

class Main {

public static void main (String[] args) {

Student s = new Student();

s. getd();

s. display();

double sgpa = s. SGPA();

System.out.println ("SGPA : %.2f\n",  
sgpa);

}

Output:

Enter USN: 1BM2SCS311

Enter name: Shashank

Enter number of subjects: 2

Enter Credits for subject 1: 10

Enter marks for subject 1: 95

Enter credits for subject 2: 5

Enter marks for subject 2: 86

Student Details:

USN: 1BM2SCS311 Name: Shashank

Subject wise credits & marks:

Subject 1: Credits = 10, Marks = 95

Subject 2: Credits = 5, Marks = 86

SGPA: 9.67

OP seen  
Date  
19/12/2024

```

import java.util.Scanner;

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

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = scanner.nextLine();
        System.out.print("Enter Name: ");
        name = scanner.nextLine();
        System.out.print("Enter number of subjects: ");
        int n = scanner.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = scanner.nextInt();
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Credits and Marks:");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits=" + credits[i] + ", Marks=" +
marks[i]);
        }
    }

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

```

```

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

private int getGradePoint(int marks) {
    if (marks >= 90) return 10;
    if (marks >= 80) return 9;
    if (marks >= 70) return 8;
    if (marks >= 60) return 7;
    if (marks >= 50) return 6;
    if (marks >= 40) return 5;
    return 0;
}
}

```

```

public class javalab2 {
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
        student.calculateSGPA();
    }
}

```

```

[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab2
Enter USN: 1BM23CS311
Enter Name: shashank
Enter number of subjects: 3
Enter credits for subject 1: 2
Enter marks for subject 1: 94
Enter credits for subject 2: 3
Enter marks for subject 2: 87
Enter credits for subject 3: 4
Enter marks for subject 3: 97
USN: 1BM23CS311
Name: shashank
Credits and Marks:
Subject 1: Credits=2, Marks=94
Subject 2: Credits=3, Marks=87
Subject 3: Credits=4, Marks=97
SGPA: 9.666666666666666
shashank@Shashanks-MacBook-Air 00J_Lab %

```

**Question 3:** 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.

DOMS Page No. 7  
Date / /

### LAB PROGRAM - 03

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.

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

```
class Book {
```

```
    private String name;  
    private String author;  
    private double price;  
    private int numPages;
```

```
    public Book (String name, String author,  
                double price, int numPages) {
```

```
        this.name = name;
```

~~```
        this.author = author;
```~~~~```
        this.Price = price;
```~~~~```
        this.numPages = numPages;
```~~

}

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

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

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

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

```
String getName (){  
    return name;  
}
```

```
String getAuthor (){  
    return author;  
}
```

```
double getPrice (){  
    return price;  
}
```

```
int getNumPages (){  
    return numPages;  
}
```

@Override

```
public String toString() {  
    return "Book Name: " + name +  
        "\nAuthor: " + author + "\nPrice: " +  
        price + "\nNumber of Pages: " +  
        numPages;
```

}

Class Main {

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

```
System.out.println ("Enter number  
of books: ");
```

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

for.

```
BOOK [] books = new Book [n];
```

for (int i=0; i<n; i++) {

System.out.println ("Enter details  
for book " + (i+1) + ": ");

System.out.println ("Enter book name");

String name = sc.nextLine();

System.out.println ("Enter author name");

String author = sc.nextLine();

```
System.out.println("Enter price: ");
double price = sc.nextInt();
System.out.println("Enter number of
pages: ");
int numPages = sc.nextInt();
```

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

```
}
```

```
System.out.println("\n Book Details : ")
for (int i=0; i<n; i++) {
```

```
System.out.println("\n Details of
book " + (i+1) + ":" );
```

```
System.out.println(books[i].toString());
```

o/p  
not seen

o/p  
9/12/24

Lab program → output:

Enter the number of books: 3

Enter details for book 1

Enter name: Pride

Enter author: Embolo Mbue

Enter price: 399

Enter number of pages: 800

Enter details for book 2

Enter name: Grief

Enter author: Ceridwen Dovey

Enter price: 499

Enter number of pages: 300

Enter details for book 3

Enter name: Ulysses

Enter author: James Joyce

Enter price: 599

Enter number of pages: 400

Book Details:

Name: Pride, Author: Embolo Mbue, Price: 399.0, Pages: 800

Name: Grief, Author: Ceridwen Dovey, Price: 499.0, Pages: 500

Name: Ulysses, Author: James Joyce, Price: 599.0, Pages: 400

```
import java.util.Scanner;

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

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

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

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }

    public String toString() {
        return "Name: " + name + ", Author: " + author + ", Price: " + price + ", Pages: " +
numPages;
    }
}
```

```

public class javalab3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1));
            scanner.nextLine();
            System.out.print("Enter name: ");
            String name = scanner.nextLine();
            System.out.print("Enter author: ");
            String author = scanner.nextLine();
            System.out.print("Enter price: ");
            double price = scanner.nextDouble();
            System.out.print("Enter number of pages: ");
            int numPages = scanner.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }

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

```

```

[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab3
Enter the number of books: 3
Enter details for book 1
Enter name: Pride
Enter author: Imbolo Mbue
Enter price: 399
Enter number of pages: 200
Enter details for book 2
Enter name: Grief
Enter author: Ceridwen Dovey
Enter price: 499
Enter number of pages: 300
Enter details for book 3
Enter name: Ulysses
Enter author: James Joyce
Enter price: 599
Enter number of pages: 400
Book Details:
Name: Pride, Author: Imbolo Mbue, Price: 399.0, Pages: 200
Name: Grief, Author: Ceridwen Dovey, Price: 499.0, Pages: 300
Name: Ulysses, Author: James Joyce, Price: 599.0, Pages: 400
shashank@Shashanks-MacBook-Air 00J_Lab %

```

**Question 4: 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 contains only the method printArea() that prints the area of the given shape.**

DOMS | Page No. 11  
Date / /

### LAB PROGRAM -04

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 of the classes contain only one method printArea() that prints the area of the given shape.

```
abstract class Shape {  
    int dimension1;  
    int dimension2;  
    Shape (int dim1, int dim2) {  
        this.dimension1 = dim1;  
        this.dimension2 = dim2;  
    }
```

abstract void printArea();

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

@override

void printArea() {

int area = dimension1 \* dimension2;

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 \* dimension1 \* dimension2;

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

}

}

class Circle extends Shape {

Circle (int radius) {

super(radius, 0);

}

@override

void printArea () {

double area = Math.PI \* dimension1  
\* dimension1;

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

public class Main {

```
public static void main (String [] args) {  
    Book book = new Book ("1015");
```

Shape rect = new Rectangle(10, 5);

Shape tri = new Triangle (10, 15);

~~Shape circle = new Circle(7);~~

rect. park Area C);

for (int i = 0; i < points.length; i++)

circle. point Area;

y

write the o/p

Output:

Area of rectangle : 50

Area of triangle : 25

Area of circle : 153.93804

slip seen  
effici  
19/12/2023

```

import java.util.Scanner;

abstract class Shape {
    int dim1, dim2;

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.dim1 = length;
        this.dim2 = breadth;
    }

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

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.dim1 = base;
        this.dim2 = height;
    }

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

class Circle extends Shape {
    Circle(int radius) {
        this.dim1 = radius;
        this.dim2 = 0;
    }

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

public class javalab4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
    }
}

```

```
System.out.print("Enter length and breadth of rectangle: ");
int length = scanner.nextInt();
int breadth = scanner.nextInt();
Rectangle rectangle = new Rectangle(length, breadth);

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

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

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

```
[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab4
Enter length and breadth of rectangle: 10
20
Enter base and height of triangle: 10
20
Enter radius of circle: 25
Area of Rectangle: 200
Area of Triangle: 100.0
Area of Circle: 1963.4954084936207
shashank@Shashanks-MacBook-Air 00J_Lab % ]
```

**Question 5: Develop a Java program to create a class Bank that maintains two kinds of accounts 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 to achieve the following tasks: a) Accept deposit from the customer and update the balance. b) Display the balance. c) Compute and deposit interest. d) Permit withdrawal and update the balance. Check for the minimum balance, impose a penalty if necessary, and update the balance.**

### LAB PROGRAM - 05

Develop a Java program to create a class Bank that maintains two kinds of accounts for its customers, one called savings account and the other current account. The savings account provider compounds interest and withdraw facilities but no cheque book facility. The current account provider 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 & update the balance
- b) Display the balance

- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

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

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

```
class Account {
```

```
    String customerName;
    int accountNumber;
    String accountType;
    double balance;
```

```
public Account (String customerName,
                int accountNumber, String accountType,
                double balance) {
```

```
    this.customerName = customerName;
```

```
    this.accountNumber = accountNumber;
```

```
    this.accountType = accountType;
```

```
    this.balance = balance;
```

```
}
```

```

public void deposit (double amount) {
    if (amount > 0) {
        balance += amount;
        System.out.println ("Deposited: " +
                            + amount);
        displayBalance ();
    } else {
        System.out.println ("Invalid
                            deposit amount.");
    }
}

```

```

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

```

~~class SavAcct extends Account {~~

~~private final double interestRate = 0.04;~~

```

public SavAcct (String customerName,
                int accountNumber, double balance) {
    super (customerName, accountNumber,
          "Savings", balance);
}

```

```
public void Comp (int years) {
    double interest = balance *  

        Math.pow((1 + interestRate), years)
    - balance;
}
```

```
deposit (interest);
System.out.println ("Interest deposited  

    for " + years + " years: " + interest);
```

```
}
```

```
public void withdraw (double amt) {
    if (amt > 8) amt <= balance;
```

```
balance -= amt;
```

```
System.out.println ("Withdraw: " +  

    amt);
```

```
}
```

```
else {
```

```
System.out.println ("Insufficient  

    balance or invalid amount.");
```

```
}
```

```
}
```

class Current extends Account {

private final double minBal = 500.0;

private final double serCh = 50.0;

public Current (String c, int a, double b)  
{

super (c, a, "Current", b);

public void check () {

if (balance < minimumBalance) {

System.out.println ("Balance

below minimum. Imposing

service charge of " + serCh );

balance -= serviceCharge;

displayBalance ();

}

public void withdraw (double amount) {

if (amount > 0 && amount < balance) {

balance -= amount;

System.out.println ("withdrawn : "  
+ amount);

check ();

displayBalance();

}

else {

System.out.println ("Insufficient  
balance or invalid amount.");

}

}

}

public class Main {

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

SavAcct savAcct = new SavAcct  
( "Alice", 1001, 2000.0 );

System.out.println ("Savings  
account created for Alice");

savAcct.deposit (500);

savAcct.comp (2);

savAcct.withdraw (300);

~~System.out.println ();~~

CurrAcct currAcct = new CurrAcct  
( "Bob", 1002, 1500.0 );

System.out.println ("Current account  
created for Bob");

2200

1200

DOMS | Page No. 21  
Date / /

current deposit (Good);  
current withdraw (1000);  
current withdraw (1300);

3

with the o/p

Output:

Savings account created for Alice

Deposited : 500

Interest deposited for 2 years : 204

Withdrawn : 300

Current account created for Bob

Deposited : 700

Withdrawn : 1000

Inufficient balance or invalid amount.

off see  
~~off the  
balance~~

```

import java.util.Scanner;

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

    Account(String customerName, String accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited. Updated balance: " + balance);
    }

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

class SavAcct extends Account {
    SavAcct(String customerName, String accountNumber) {
        super(customerName, accountNumber, "Savings");
    }

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

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

```

```

}

class CurAcct extends Account {
    private static final double MIN_BALANCE = 500;
    private static final double PENALTY = 50;

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

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn. Updated balance: " + balance);
            if (balance < MIN_BALANCE) {
                balance -= PENALTY;
                System.out.println("Minimum balance not maintained. Penalty imposed.
Updated balance: " + balance);
            }
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

public class javalab5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SavAcct savings = new SavAcct("Alice", "SAV123");
        CurAcct current = new CurAcct("Bob", "CUR456");

        while (true) {
            System.out.println("\nChoose Account Type: 1. Savings 2. Current 3. Exit");
            int accountChoice = sc.nextInt();
            if (accountChoice == 3) break;

            Account selectedAccount = (accountChoice == 1) ? savings : current;
            System.out.println("Choose Operation: 1. Deposit 2. Withdraw 3. Display
Balance");
            int operation = sc.nextInt();

            switch (operation) {
                case 1:
                    System.out.print("Enter amount to deposit: ");

```

```

        double depositAmount = sc.nextDouble();
        selectedAccount.deposit(depositAmount);
        break;
    case 2:
        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = sc.nextDouble();
        if (selectedAccount instanceof SavAcct) {
            ((SavAcct) selectedAccount).withdraw(withdrawAmount);
        } else {
            ((CurAcct) selectedAccount).withdraw(withdrawAmount);
        }
        break;
    case 3:
        selectedAccount.displayBalance();
        break;
    default:
        System.out.println("Invalid choice.");
    }
}

sc.close();
}

```

```

|shashank@Shashanks-MacBook-Air 00J_Lab % java javalab5

Choose Account Type: 1. Savings 2. Current 3. Exit
1
Choose Operation: 1. Deposit 2. Withdraw 3. Display Balance
1
Enter amount to deposit: 2000
Amount deposited. Updated balance: 2000.0

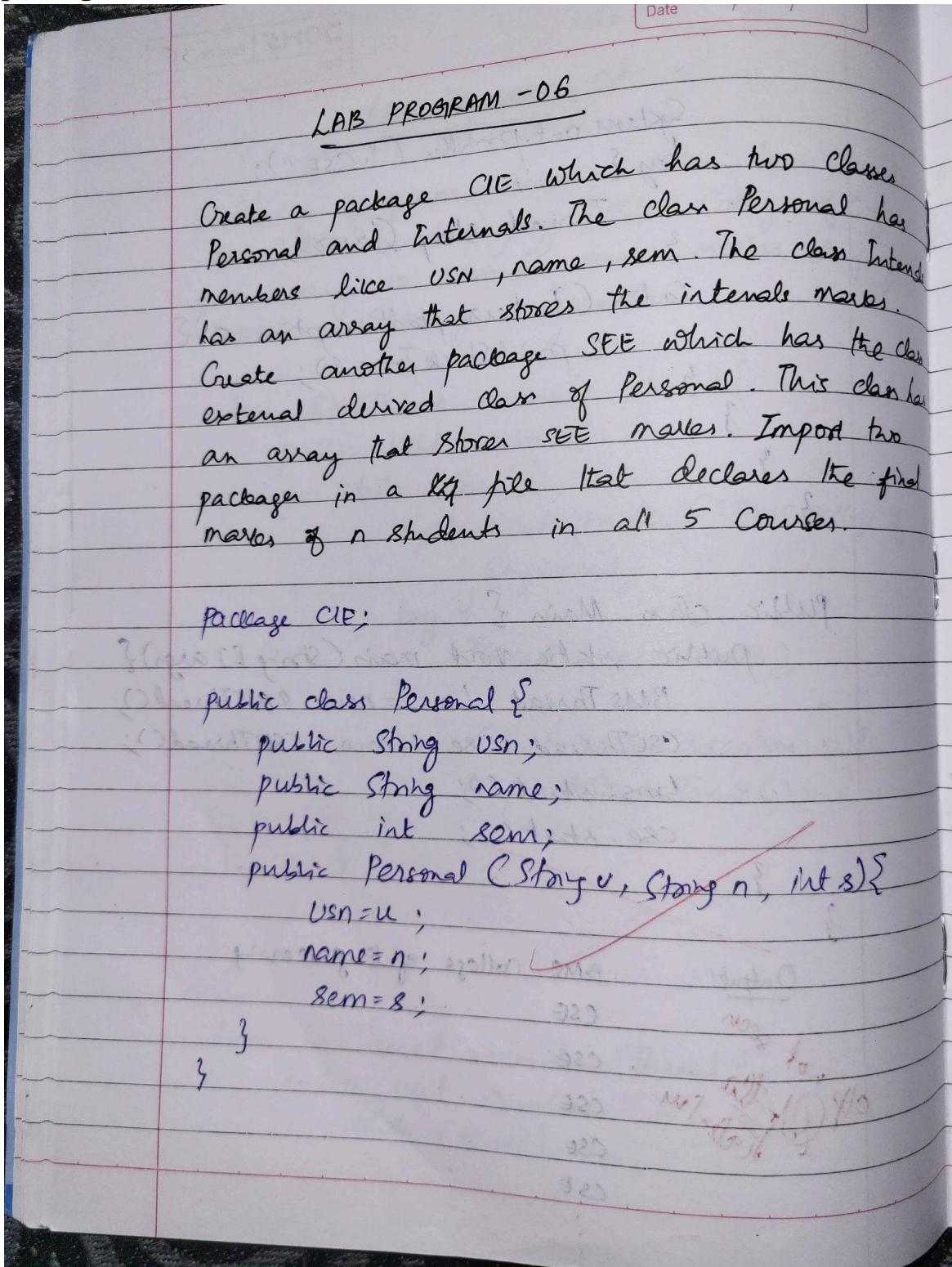
Choose Account Type: 1. Savings 2. Current 3. Exit
1
Choose Operation: 1. Deposit 2. Withdraw 3. Display Balance
2
Enter amount to withdraw: 500
Amount withdrawn. Updated balance: 1500.0

Choose Account Type: 1. Savings 2. Current 3. Exit
1
Choose Operation: 1. Deposit 2. Withdraw 3. Display Balance
3
Account Balance: 1500.0

Choose Account Type: 1. Savings 2. Current 3. Exit
3
shashank@Shashanks-MacBook-Air 00J_Lab %

```

**Question 6:** Create a package CIE which has two classes - Personal 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 Personal. 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.



Date / /

package CIE;

public class Internal {  
 public int[] internalmarkers = new int[5];  
 public Internal(int[] markers) {  
 if (markers.length == 5) {  
 internalmarkers = markers;  
 }  
 }  
 else {  
 throw new IllegalArgumentException("Exactly 5 markers");  
 }  
}

package SEE; ~~import CIE.Internal~~  
 import CIE.Personal;

~~public class External extends Personal {  
 public int[] externalmarkers = new int[5];  
 public External(String u, String n, int s,  
 int[] markers) {  
 super(u, n, s);  
 if (markers.length == 5) {  
 externalmarkers = markers;  
 }  
 }  
}~~

else {

(Know how illegal Argument Exception C<sup>u</sup>  
Exactly 5 marks are required);

import CIE.\*;  
import SEE.\*;

public class Main {  
 public static void main(String[] args) {  
 int n = 2;

Personal[] students = new Personal[n];

Internal[] Internmarks = new Internal[n];

External[] Externalmarks = new External[n];

Students[0] = new Personal("IBM23CS11",

"Shashank", 5);

Internmarks[0] = new Internal(new Int[]

{25, 30, 32, 35, 37});

Externalmarks[0] = new External(new Int[]

{"Shashank", 5, new Int[] {70, 90, 85, 80, 90}});

for (int i=0; i < n; i++) {

System.out.println ("USN: " + students[i].  
usn);

System.out.println ("Name: " + students[i].name);

System.out.println ("Sem: " + students[i].sem);

System.out.println ("Total marks: ");

for (int j=0; j < 5; j++) {

int finalmark = internalmarks[i].

internalmarks[j] +

externalmarks[i].

externmarks[j];

}

System.out.println ();

Q?   
 off not seen  
 off this question

### Lab program 6 output:

Enter the number of students: 1

Enter details for student 1

USN: IBM23CS311

Name: Shashank

Semester: 3

Enter internal marks for 5 courses:

29

37

40

34

29

Enter SEE marks for 5 Courses:

77

86

79

82

100

Final marks of Students :

Student 1 - USN: IBM23CS311, Name: Shashank,

Semester: 3

Course-wise Final marks: Course 1: 62

Course 2: 80

Course 3: 89

Course 4: 75

Course 5: 79

```
package CIE;

public class Internals {
    public int[] internalMarks = new int[5];

    public void setMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }
}
```

```
package CIE;

public class Personal {
    public String usn;
    public String name;
    public int sem;

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

```
package SEE;

import CIE.Personal;

public class External extends Personal {
    public int[] externalMarks = new int[5];

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

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

```

import CIE.Personal;
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

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

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();
        External[] students = new External[n];
        Internals[] internals = new Internals[n];

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

            students[i] = new External(usn, name, sem);
            internals[i] = new Internals();

            System.out.println("Enter internal marks for 5 courses:");
            int[] intMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                intMarks[j] = scanner.nextInt();
            }
            internals[i].setMarks(intMarks);

            System.out.println("Enter SEE marks for 5 courses:");
            int[] extMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                extMarks[j] = scanner.nextInt();
            }
            students[i].setMarks(extMarks);
        }

        System.out.println("Final Marks of Students:");
        for (int i = 0; i < n; i++) {
    
```

```

        System.out.println("Student " + (i + 1) + " - USN: " + students[i].usn + ", Name: "
" + students[i].name + ", Semester: " + students[i].sem);
        System.out.println("Course-wise Final Marks:");
        for (int j = 0; j < 5; j++) {
            int finalMark = internals[i].internalMarks[j] + (students[i].externalMarks[j] /
2);
            System.out.println("Course " + (j + 1) + ": " + finalMark);
        }
    }
}
}

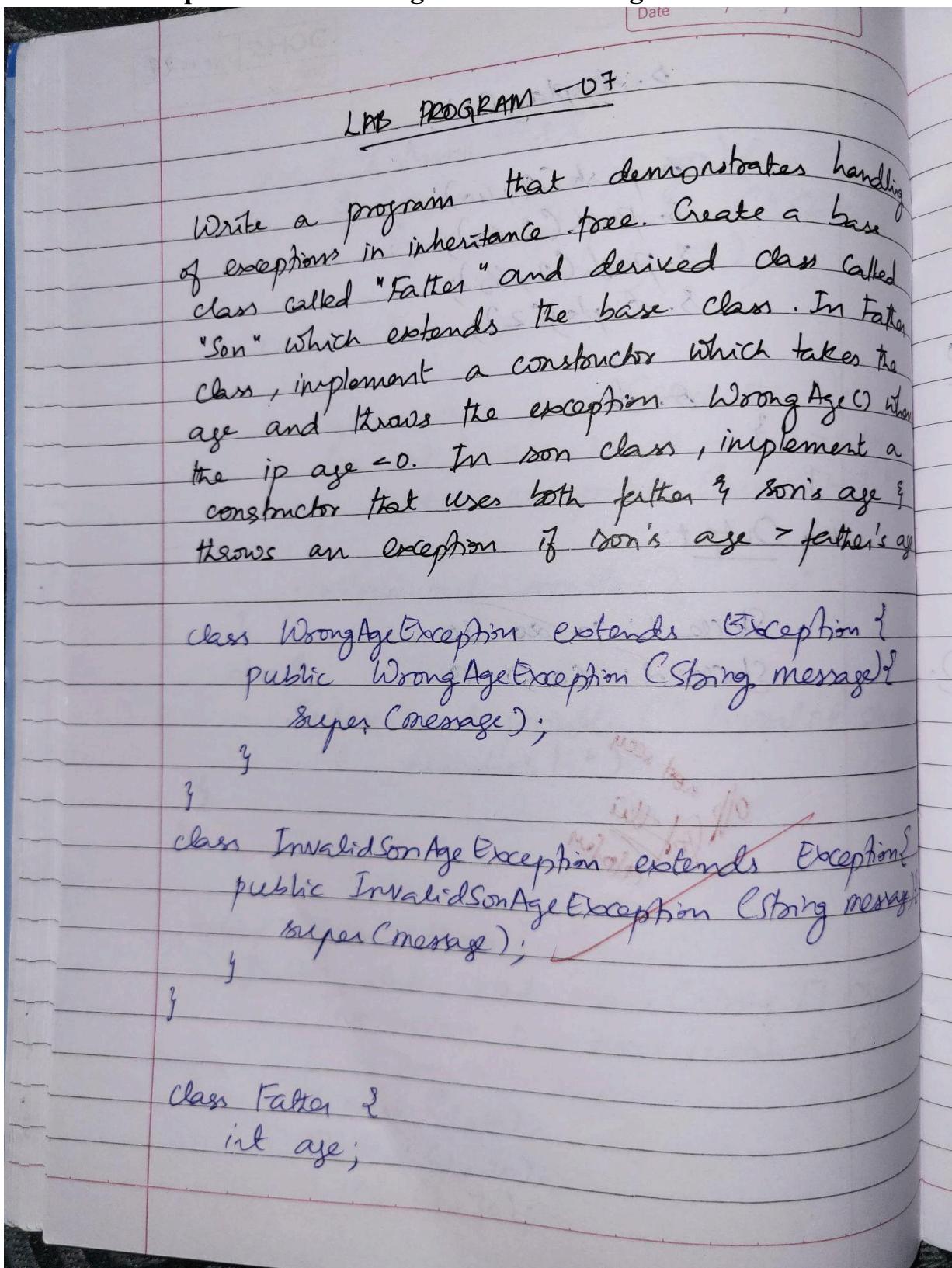
```

```

[shashank@Shashanks-MacBook-Air 00J_Lab % java Main
Enter the number of students: 1
Enter details for student 1
USN: 1BM23CS311
Name: shashank
Semester: 3
Enter internal marks for 5 courses:
24
37
40
34
29
Enter SEE marks for 5 courses:
77
86
99
82
100
Final Marks of Students:
Student 1 - USN: 1BM23CS311, Name: shashank, Semester: 3
Course-wise Final Marks:
Course 1: 62
Course 2: 80
Course 3: 89
Course 4: 75
Course 5: 79
shashank@Shashanks-MacBook-Air 00J_Lab %

```

**Question 7:** Write a program that demonstrates handling of exceptions in the inheritance tree. Create a base class called "Father" and a derived class called "Son" which extends the base class. In the Father class, implement a constructor that takes the age and throws the exception WrongAge() when the input age < 0. In the Son class, implement a constructor that uses both father and son's age and throws an exception if the son's age is  $\geq$  father's age.



public Father (int age) throws

WrongAgeException {

if (age < 0) {

throws new WrongAgeException ("

Father's age cannot be negative");

this.age = age;

class Son extends Father {

int sonAge;

public Son (int fage, int sage) throws  
WrongAgeException, InvalidSonAgeException {

super (fage);

if (sage < 0) {

throws new WrongAgeException ("Son's  
age cannot be negative.");

else if (sage >= fatherage) {

throws new InvalidSonAgeException

("Son's age cannot be greater than  
or equal to father's age.");

this. sonAge = age;

public class Example {

public static void main(String args){

try {

System.out.println("Creating");

Father f = new Father(40);

Son s = new Son(20, 20);

System.out.println("Father's age: " + f.age);

System.out.println("Son's age: " + s.age);

catch (WrongAgeException e) {

System.out.println(e);

} try {

Father f2 = new Father(-5);

catch (WrongAgeException e) {

System.out.println(e);

try {

Son .52 = new Son (30, 35);

}

Catch { InvalidSonAgeException e ) {

System.out.println (e);

}

}

op?

op not seen

if the  
exception

Lab program 7 output:

Father's age: 45

Son's age: 20

Exception: Father's age cannot be negative

Exception: Son's age cannot be greater than or equal to Father's age.

```

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

class Father {
    int age;

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

class Son extends Father {
    int sonAge;

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

public class javalab7 {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```

```
try {
    Father father = new Father(-5);
} catch (WrongAgeException e) {
    System.out.println("Exception: " + e.getMessage());
}

try {
    Son son = new Son(30, 35);
} catch (WrongAgeException e) {
    System.out.println("Exception: " + e.getMessage());
}
}
```

```
[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab7
Father's age: 45
Son's age: 20
Exception: Father's age cannot be negative.
Exception: Son's age cannot be greater than or equal to Father's age.
shashank@Shashanks-MacBook-Air 00J_Lab % ]
```

**Question 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.

### LAB PROGRAM - 08

Write a program which creates two threads,  
one thread displaying "BMS College of Engineering"  
once every 10 seconds and another displaying  
"CSE" once every 2 seconds.

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

```
class CSEThread extends Thread {  
    public void run() {  
        while (true) {  
            //  
        }  
    }  
}
```

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

```
    Thread.sleep (2000);
```

```
}  
catch (InterruptedException e) {  
    e.printStackTrace();
```

```
}  
}

```
}
```


```

```
}
```

```
public class Main {
```

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

```
        BMS.Thread bms = new BMSThread();
```

```
        CSEThread cse = new CSEThread();
```

```
        bms.start();
```

```
        cse.start();
```

```
}  
}
```

Output:

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

CSE

:

all not seen  
at the  
same time

```
public class javalab8 {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(() -> {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                try { Thread.sleep(10000); } catch (InterruptedException e) {}  
            }  
        });  
        Thread t2 = new Thread(() -> {  
            while (true) {  
                System.out.println("CSE");  
                try { Thread.sleep(2000); } catch (InterruptedException e) {}  
            }  
        });  
        t1.start();  
        t2.start();  
    }  
}
```

```
[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab8  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE
```

**Question 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 are not integers, the program throws a NumberFormatException. If Num2 is zero, the program throws an ArithmeticException. Display the exception in a message dialog box.**

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

public class javalab9 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Division Calculator");
        JTextField num1Field = new JTextField(10);
        JTextField num2Field = new JTextField(10);
        JTextField resultField = new JTextField(10);
        resultField.setEditable(false);
        JButton divideButton = new JButton("Divide");

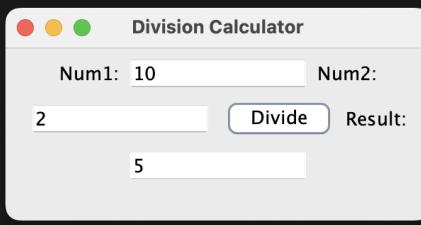
        divideButton.addActionListener((ActionEvent e) -> {
            try {
                int num1 = Integer.parseInt(num1Field.getText());
                int num2 = Integer.parseInt(num2Field.getText());
                int result = num1 / num2;
                resultField.setText(String.valueOf(result));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(frame, "Invalid number format", "Error",
JOptionPane.ERROR_MESSAGE);
            } catch (ArithmaticException ex) {
                JOptionPane.showMessageDialog(frame, "Cannot divide by zero", "Error",
JOptionPane.ERROR_MESSAGE);
            }
        });

        frame.setLayout(new FlowLayout());
        frame.add(new JLabel("Num1:"));
        frame.add(num1Field);
        frame.add(new JLabel("Num2:"));
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(new JLabel("Result:"));
        frame.add(resultField);
        frame.setSize(300, 150);
    }
}
```

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
}
}
}

shashank@Shashanks-MacBook-Air OOJ_Lab % java javalab9
2024-12-23 17:27:43.610 java[29730:11687285] +[IMKClient subclass]: chose IMKClient_Modern
2024-12-23 17:27:43.610 java[29730:11687285] +[IMKInputSession subclass]: chose IMKInputSession_Modern
[]


```



The image shows a Java application window titled "Division Calculator". The window has a light gray background and a white content area. It contains four text input fields and one button. The first field is labeled "Num1:" and contains the value "10". The second field is labeled "Num2:" and contains the value "5". Below these is a button labeled "Divide". To the right of the "Divide" button is another field labeled "Result:". The entire window is centered on a dark gray desktop background.

## Question 10: Demonstrate Inter-process Communication and deadlock.

```
class SharedResource {  
    synchronized void method1(SharedResource other) {  
        System.out.println(Thread.currentThread().getName() + " is executing method1");  
        try { Thread.sleep(100); } catch (InterruptedException e) {}  
        other.method2(this);  
    }  
  
    synchronized void method2(SharedResource other) {  
        System.out.println(Thread.currentThread().getName() + " is executing method2");  
    }  
}  
  
public class javalab10 {  
    public static void main(String[] args) {  
        SharedResource resource1 = new SharedResource();  
        SharedResource resource2 = new SharedResource();  
  
        Thread t1 = new Thread(() -> resource1.method1(resource2), "Thread-1");  
        Thread t2 = new Thread(() -> resource2.method1(resource1), "Thread-2");  
  
        t1.start();  
        t2.start();  
    }  
}  
[shashank@Shashanks-MacBook-Air 00J_Lab % java javalab10  
Thread-1 is executing method1  
Thread-2 is executing method1  
^C%  
shashank@Shashanks-MacBook-Air 00J_Lab % ]
```









