

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY, JNANASANGAMA,
BELGAUM - 590014, KARNATAKA**



LABORATORY RECORD
ON
Object Oriented Java Programming
(23CS3PCOOJ) *Submitted by*
MOHAMMAD YASEEN IMAM (1BM22CS156)

In partial fulfilment 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
December-2022 to April-2023

INDEX

Sl. No.	Date	Experiment Title	Page No.
1	18.12.2023	Laboratory Program – 1	3-6
2	01.01.2024	Laboratory Program – 2	7-11
3	01.01.2024	Laboratory Program – 3	12-15
4	08.01.2024	Laboratory Program – 4	16-19
5	08.01.2024	Laboratory Program – 5	20-26
6	22.01.2024	Laboratory Program – 6	27-32
7	22.01.2024	Laboratory Program – 7	33-36
8	05.02.2024	Laboratory Program – 8	37-39
9	19.02.2024	Laboratory Program – 9	40-43

LABORATORY PROGRAM - 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.

```
import java.util.Scanner;
import java.lang.Math;
class Quad
{
    double Disc(double a,double b,double c)
    {
        return b*b-4*a*c;
    }
    void roots(double a,double b, double c)
    {
        double D = Disc(a,b,c);

        if (D<0)
        {
            double realPart = -b/(2*a);
            double imaginaryPart = Math.sqrt(Math.abs(D))/(2*a);
            System.out.println("The Quadratic Equation has Conjugate Imaginary
roots:");
            System.out.printf("Root 1: %.5f + %.5fi%n",realPart,imaginaryPart);
            System.out.printf("Root 2: %.5f - %.5fi%n",realPart,imaginaryPart);
        }
        else if (D>0)
        {
            System.out.println("The Quadratic Equation has Two Distinct Real Roots:");
            double r1=(-b+Math.sqrt(D))/(2*a);
            double r2=(-b-Math.sqrt(D))/(2*a);
            System.out.printf("Root 1: %.5f%n",r1);
            System.out.printf("Root 2: %.5f%n", r2);
        }
        else
        {
            System.out.println("The Quadratic Equation has Equal and Real Root:");
            double r1=(-b)/(2*a);
            System.out.printf("Both Root 1 and Root 2: %.5f%n",r1);
        }
    }
}

class QuadEqn
{
    public static void main(String sx[])
    {
```

```

{
    Scanner S1 = new Scanner(System.in);
    System.out.println("Enter the Coefficients of Quadratic Equation : ");
    double a = S1.nextDouble();
    double b = S1.nextDouble();
    double c = S1.nextDouble();
    if (a==0)
    {
        System.out.println("Since the Coefficient of x^2 is Zero, it's not a Quadratic
Equation");
    }
    else
    {
        Quad quadratic=new Quad();
        quadratic.Disc(a,b,c);
        quadratic.roots(a,b,c);
    }
}
}

```

OUTPUT

```

D:\NotePad++\Java>javac QuadEqn.java

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
23
24
26
The Quadratic Equation has Conjugate Imaginary roots:
Root 1: -0.52174 + 0.92640i
Root 2: -0.52174 - 0.92640i

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
1
2
1
The Quadratic Equation has Equal and Real Root:
Both Root 1 and Root 2: -1.00000

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
6
8
0
The Quadratic Equation has Two Distinct Real Roots:
Root 1: 0.00000
Root 2: -1.33333

```

develop a java program that prints all real solution to the quadratic equation $ax^2+bx+c=0$. Read co-efficients and use the quadratic formula. If the discriminant b^2-4ac is negative & display a message stating that there are no real solutions.

```
import java.util.Scanner  
class First  
{  
    public static void main (String arg [])  
    {  
        int a,b,c,d;  
        float r1,r2;  
        System.out.print("Enter the co-efficients of  
        quadratic equation")  
        Scanner s1 = new Scanner (System.in);  
        a = s1.nextInt();  
        b = s1.nextInt();  
        c = s1.nextInt();  
        if (a==0)  
        {  
            System.out.print("co-efficients are invalid");  
        }  
        else  
        {  
            d = (b*b)-(4*a*c);  
            if (d>0)  
            {  
                System.out.print("It has real and distinct roots");  
                r1 = (-b+sqrt(d))/(2*a);  
                r2 = (-b-sqrt(d))/(2*a);  
                System.out.print("The roots are " + r1 + " and " + r2);  
            }  
        }  
    }  
}
```

```

the
the
{
    system.out.printIn("It has real and equal roots");
    r1 = (-b)/(2*a);
    system.printIn("The roots are "+r1+" and "+r2);
}
else {
    system.out.printIn("There are no real solutions");
}
}
}

O/P: Enter the co-efficients of quadratic equation

```

O/P: Enter the co-efficients of quadratic equation

1

4

1

It has real and distinct roots

The roots are -0.26794 and -3.7320

The roots are -0.26794 and -3.7320

Enter the co-efficients of quadratic equation

1/4

4

1

It has no real roots

Enter the co-efficients of quadratic equation

1

1

1

It has real and equal roots

LABORATORY PROGRAM - 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.

```
import java.util.Scanner;

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

    public void acceptDetails() {
        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: ");
        int n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = sc.nextInt();
            System.out.println("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

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

    public double calculateSGPA() {
```

```

double totalGrade = 0;
int totalCredit = 0;
for (int i = 0; i < credits.length; i++) {
    totalGrade += getGrade(marks[i]) * credits[i];
    totalCredit += credits[i];
}
return totalGrade / totalCredit;
}

private double getGrade(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;
    }
}

public static void main(String[] args) {
    Student student = new Student();
    student.acceptDetails();
    student.displayDetails();
    System.out.println("SGPA: " + student.calculateSGPA());
}
}

```

OUTPUT

```
D:\NotePad++\Java>javac Student.java

D:\NotePad++\Java>java Student
Enter USN:
L24
Enter Name:
Royce
Enter number of subjects:
3
Enter credits for subject 1:
4
Enter marks for subject 1:
97
Enter credits for subject 2:
3
Enter marks for subject 2:
98
Enter credits for subject 3:
2
Enter marks for subject 3:
99
USN: L24
Name: Royce
Marks:
Subject 1: 97
Subject 2: 98
Subject 3: 99
Credits:
Subject 1: 4
Subject 2: 3
Subject 3: 2
SGPA: 10.0
```

Code 2:

```
import java.util.Scanner;

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

class Student {
    Subject subject[];
    String name;
    String USN;
    double SGPA;
    Scanner sc;
}

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

void getStudentDetails() {
    sc = new Scanner(System.in);
    System.out.print("Enter name:");
    this.name = sc.next();

    System.out.print("Enter your USN:");
    this.USN = sc.next();
}

void getMarks() {
    sc = new Scanner(System.in);
    for (int i=0; i<8; i++) {
        System.out.print("Enter " + (i+1) + " subject marks:");
    }
}
```

```

subject[i].subjectMarks = sc.nextInt();
System.out.print("Enter number of credits : ");
subject[i].credits = sc.nextInt();
subject[i].grade == (subject[i].subjectMarks / 10) + 1;
if (subject[i].grade == 11) subject[i].grade = 10;
if (subject[i].grade <= 4) subject[i].grade = 0;
}
}

void computeSGPA() {
int totalCredits = 0;
for (int i = 0; i < 8; i++) {
totalCredits += subject[i].credits;
}
int totalGradeAndCredit = 0;
for (int i = 0; i < 8; i++) {
totalGradeAndCredit += (subject[i].credits * subject[i].grade);
}
this.SGPA = ((float) totalGradeAndCredit / totalCredits);
System.out.println("SGPA of the student is :" + this.SGPA);
}

class StudentMain {
public static void main(String[] args) {
Student s1 = new Student();
s1.getStudentDetails();
s1.getMarks();
s1.computeSGPA();
}
}

```

O/P

Enter USN:
L24
Enter Name:
Royce
Enter number of subjects:
3
Enter credits off subject 1: 4
Enter marks for subject 1: 97
Enter credits for subject 2: 3
Enter marks for subject 2: 98
Enter credits for subject 3: 2
Enter marks for subject 3: 99
USN: L24
Name: Royce
Marks:
Subject 1: 97
Subject 2: 98
Subject 3: 99
Credits:
Subject 1: 4
Subject 2: 3
Subject 3: 2
SGPA: 10.0

LABORATORY PROGRAM - 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.

```
import java.util.Scanner;

class Book
{
    String name;
    String author;
    double price;
    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 void setName(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return name;
    }

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

    public String getAuthor()
    {
        return author;
    }

    public void setPrice(double price)
    {
```

```

        this.price = price;
    }

    public double getPrice()
    {
        return price;
    }

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

    public int getNumPages()
    {
        return num_pages;
    }

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

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of books: ");
        int n = sc.nextInt();
        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 name: ");
            String name = sc.next();
            System.out.println("Enter author: ");
            String author = sc.next();
            System.out.println("Enter price: ");
            double price = sc.nextDouble();
            System.out.println("Enter number of pages: ");
            int num_pages = sc.nextInt();
            books[i] = new Book(name, author, price, num_pages);
        }
        for (int i = 0; i < n; i++){
            System.out.println("Details of book " + (i + 1) + ":" );
            System.out.println(books[i].toString());
        }
    }
}

```

OUTPUT

```
D:\NotePad++\Java>java Book
Enter the number of books:
2
Enter details for book 1:
Enter name:
Rog
Enter author:
Zephyrus
Enter price:
560
Enter number of pages:
1536
Enter details for book 2:
Enter name:
Power
Enter author:
Murphy
Enter price:
399
Enter number of pages:
256
Details of book 1:
Name: Rog
Author: Zephyrus
Price: 560.0
Number of Pages: 1536
Details of book 2:
Name: Power
Author: Murphy
Price: 399.0
Number of Pages: 256
```

```

Code 3:
books of four members
name, author, price, num-pages
import java.util.Scanner;
class Books {
    Scanner sc = new Scanner(System.in);
    String name; String author;
    int price; int numPages;
    Books() {
        Books(String name, String author, int price,
              int numPages) {
            this.name = name;
            this.author = author;
            this.price = price;
            this.numPages = numPages;
        }
    }
    class BookDetail {
        public static void main(String args[]) {
            Scanner sc = new Scanner(System.in);
            Books b1 = new Books("Mita", "Neil", 56, 154);
            System.out.println("Constructor values : " + b1);
            System.out.print("Enter the no. of books : ");
            int n = sc.nextInt();
        }
    }
}

```

```

books sc[] = new Books[n];
for(int i=0; i<n; i++) {
    sc[i] = new Books();
    System.out.print("Enter the details of " + (i+1) + " book");
    sc[i].getData();
}
for(int i=0; i<n; i++) {
    System.out.println("\n Details of the book " + (i+1));
    System.out.println(sc[i]);
}

```

Enter the number of books :
 2
 Enter details for book 1:
 Enter name : Reg
 Enter author : Zephyrus
 Enter price : 560
 Enter number of pages : 1536
 Enter details for book 2:
 Enter name : Rouser
 Enter author : Murphy
 Enter price : 399
 Enter number of pages : 256
 Details of book 1:
 Name : Reg
 Author : Zephyrus
 Price: 560
 Number of pages : 1536
 Details of book 2:
 Name : Rouser
 Author : Murphy
 Price : 399.0
 Number of pages : 256

LABORATORY PROGRAM - 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 contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;

abstract class Shape
{
    abstract void printArea();
    int length,breadth;
}

class Rectangle extends Shape
{
    Rectangle(int l,int b)
    {
        length=l;
        breadth=b;
    }

    void printArea()
    {
        int area=length*breadth;
        System.out.println("Area of Rectangle is "+area);
    }
}

class Triangle extends Shape
{
    Triangle(int l,int b)
    {
        length=l;
        breadth=b;
    }

    void printArea()
    {
        double area=0.5*length*breadth;
        System.out.println("Area of Triangle is "+area);
    }
}

class Circle extends Shape
```

```

{
    Circle(int r)
    {
        length=r;
    }

    void printArea()
    {
        double area=3.14*length*length;
        System.out.println("Area of Circle is "+area);
    }
}

class Display
{
    public static void main(String sx[])
    {
        Scanner s1=new Scanner(System.in);
        Rectangle r1=new Rectangle(0,0);
        System.out.println("Enter the Length and Breadth of Rectangle, to get it's Area: ");
        int l=s1.nextInt();
        int b=s1.nextInt();
        r1=new Rectangle(l,b);
        r1.printArea();
        Triangle t1=new Triangle(0,0);
        System.out.println("Enter the base and height of triangle, to get it's Area: ");
        int bs=s1.nextInt();
        int h=s1.nextInt();
        t1=new Triangle(bs,h);
        t1.printArea();
        Circle c1=new Circle(0);
        System.out.println("Enter the Radius of Circle, to get it's Area: ");
        int r=s1.nextInt();
        c1=new Circle(r);
        c1.printArea();

    }
}

```

OUTPUT

```
D:\NotePad++\Java>javac Shape.java

D:\NotePad++\Java>java Display
Enter the Length and Breadth of Rectangle, to get it's Area:
2
3
Area of Rectangle is 6
Enter the base and height of triangle, to get it's Area:
2
4
Area of Triangle is 4.0
Enter the Radius of Circle, to get it's Area:
2
Area of Circle is 12.56
```

```

Code 4:

import java.util.Scanner;
abstract class shape
{
    double a; double b;
    abstract void printArea(double a, double b);
}
class Rectangle extends shape
{
    void printArea(double a, double b)
    {
        double r;
        r = a * b;
        System.out.println("Area " + r);
    }
}

class triangle extends shape
{
    void printArea(double a, double b)
    {
        double r;
        r = (a * b) * 3 / 4;
        System.out.println("Area " + r);
    }
}

class ShapeDemo
{
    public static void main(String args[])
    {
        int sc;
        System.out.print("Enter the character");
        Scanner sc = new Scanner(System.in);
        sc = s.nextInt();
        circle c = new circle();
    }
}

```

```

triangle tri = new triangle();
while (1)
{
    System.out.print("\n 1. Rectangle\n 2. Circle\n 3. Triangle");
    switch (ch)
    {
        case 1:
            System.out.print("Enter length, breadth");
            double l = sc.nextDouble();
            double b = sc.nextDouble();
            rec.printArea(l, b);
            break;
        case 2:
            System.out.print("Enter radius");
            double r = sc.nextDouble();
            c.printArea(r, 3.14);
            break;
        case 3:
            System.out.print("Enter base and altitude");
            double ba = sc.nextDouble();
            double al = sc.nextDouble();
            tri.printArea(ba, al);
            break;
        default:
            System.out.print("Invalid value");
    }
}
// end of while
// end of program

```

Output

Enter the character:

- 1) Rectangle
- 2) Circle
- 3) Triangle

1
Enter the length and breadth
20
30
Area = 600

2
Enter the radius
5
Area = 78.5

3
Enter the base and altitude
20
20
Area = 400

LABORATORY PROGRAM - 5

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
- 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;
    long accountNumber;
    String accountType;
    double balance;

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

    public void deposit(double amount)
    {
        balance += amount;
        System.out.println("Deposit successful. Updated balance: " + balance);
    }

    public void displayBalance()
    {
        System.out.println("Account Number: " + accountNumber);
```

```

        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: " + balance);
    }
}

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

    public void computeAndDepositInterest(double rate)
    {
        double interest = balance * rate / 100;
        balance += interest;
        System.out.println("Interest computed and deposited. Updated balance: " + balance);
    }

    public void withdraw(double amount)
    {
        if (amount <= balance)
        {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        }
        else
        {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class CurrAcct extends Account
{
    double minimumBalance;
    double serviceCharge;

    public CurrAcct(String customerName, long accountNumber, double balance, double minimumBalance, double serviceCharge)
    {
        super(customerName, accountNumber, "Current", balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    private void checkMinimumBalance()

```

```

        {
        if (balance < minimumBalance)
            {
                balance -= serviceCharge;
                System.out.println("Minimum balance not maintained. Service charge imposed. Updated balance: " + balance);
            }
        }

    public void withdraw(double amount)
    {
        if (amount <= balance)
            {
                balance -= amount;
                System.out.println("Withdrawal successful. Updated balance: " + balance);
                checkMinimumBalance();
            }
        else
            {
                System.out.println("Insufficient funds. Withdrawal failed.");
            }
    }
}

public class Bank
{
    public static void main(String[] args)
    {
        Scanner s1 = new Scanner(System.in);

        System.out.print("Enter customer name for Savings Account: ");
        String SCN = s1.nextLine();
        System.out.print("Enter account number for Savings Account: ");
        long SAN = s1.nextLong();
        System.out.print("Enter initial balance for Savings Account: ");
        double SIB = s1.nextDouble();
        SavAcct SA = new SavAcct(SCN, SAN, SIB);

        System.out.print("Enter customer name for Current Account: ");
        String CCN = s1.next();
        System.out.print("Enter account number for Current Account: ");
        long CAN = s1.nextLong();
        System.out.print("Enter initial balance for Current Account: ");
        double CIB = s1.nextDouble();
        System.out.print("Enter minimum balance for Current Account: ");
        double MB = s1.nextDouble();
        System.out.print("Enter service charge for Current Account: ");
        double SC = s1.nextDouble();
    }
}

```

```
CurrAcct CA = new CurrAcct(CCN, CAN, CIB, MB, SC);

System.out.print("Enter deposit amount for Savings Account: ");
double SDA = s1.nextDouble();
SA.deposit(SDA);

System.out.print("Enter interest rate for Savings Account: ");
double SIR = s1.nextDouble();
SA.computeAndDepositInterest(SIR);

System.out.print("Enter withdrawal amount for Savings Account: ");
double SWA = s1.nextDouble();
SA.withdraw(SWA);

System.out.print("Enter deposit amount for Current Account: ");
double CDA = s1.nextDouble();
CA.deposit(CDA);

System.out.print("Enter withdrawal amount for Current Account: ");
double CWA = s1.nextDouble();
CA.withdraw(CWA);

System.out.println("\nFinal Balances:");
System.out.println("Savings Account:");
SA.displayBalance();

System.out.println("\nCurrent Account:");
CA.displayBalance();

}
```

OUTPUT

```
D:\NotePad++\Java>javac Bank.java

D:\NotePad++\Java>java Bank
Enter customer name for Savings Account: Ram
Enter account number for Savings Account: 2324
Enter initial balance for Savings Account: 5000
Enter customer name for Current Account: Ram
Enter account number for Current Account: 2324
Enter initial balance for Current Account: 6000
Enter minimum balance for Current Account: 1000
Enter service charge for Current Account: 100
Enter deposit amount for Savings Account: 2000
Deposit successful. Updated balance: 7000.0
Enter interest rate for Savings Account: 2
Interest computed and deposited. Updated balance: 7140.0
Enter withdrawal amount for Savings Account: 500
Withdrawal successful. Updated balance: 6640.0
Enter deposit amount for Current Account: 1000
Deposit successful. Updated balance: 7000.0
Enter withdrawal amount for Current Account: 750
Withdrawal successful. Updated balance: 6250.0

Final Balances:
Savings Account:
Account Number: 2324
Customer Name: Ram
Account Type: Savings
Balance: 6640.0

Current Account:
Account Number: 2324
Customer Name: Ram
Account Type: Current
Balance: 6250.0
```

Dr 19/2/2024

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 a 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 Current and Savings 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.
- d) Permit withdrawal 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) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0;
    }
}
```

```
public void deposit (double amount) {
    balance += amount;
    System.out.println("Deposit of " + amount + " successful.");
}
```

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

```
class SavingsAccount extends Account {
```

~~double interestRate;~~

```
public SavingsAccount (String customerName, int accountNumber) {
    super (customerName, accountNumber, "Savings");
    this.interestRate = 0.05;
}
```

```
public void depositInterest () {
    double interest = balance * interestRate;
    deposit (interest);
    System.out.println("Interest deposited : " + interest);
}
```

```
public void withdraw (double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal of " + amount + " successful.");
    } else {
        System.out.println("Insufficient balance for withdrawal");
    }
}
```

```
class CurrentAccount extends Account {
```

```
double minimumBalance;
double serviceCharge;
public CurrentAccount (String customerName, int accountNumber) {
    super (customerName, accountNumber, "Current");
    this.minimumBalance = 1000;
    this.serviceCharge = 50;
}
```

```
public void withdraw (double amount) {
    if (balance - amount >= minimumBalance) {
        balance -= amount;
        System.out.println("Withdrawal of " + amount + " successful.");
    }
}
```

```
} else {
```

```
    System.out.println("Insufficient balance for withdrawal.  
Service charge of " + serviceCharge + "  
will be applied.");
```

```
    balance -= serviceCharge;
```

```
}
```

```
}
```

```
}
```

```
public class Bank {
```

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

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

```
        Savings Account.sa = new Savings Account ("JohnDoe", 123456);
```

```
        Current Account.ca = new CurrentAccount ("Yaseen", 654321);
```

```
        sa.deposit (5000);
```

```
        sa.displayBalance();
```

```
        sa.depositInterest();
```

```
        sa.displayBalance();
```

```
        ca.deposit (300);
```

```
        ca.displayBalance();
```

```
        ca.withdraw (2000);
```

```
        ca.displayBalance();
```

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

```
}
```

```
}
```

LABORATORY PROGRAM - 6

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals which is a derived class of Student and 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.

```
package CIE;

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

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

package CIE;

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

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

package SEE;
import CIE.Student;

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

```

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

import java.util.Scanner;
import CIE.*;
import SEE.*;

public class CalculateFinalMarks
{
    public static void main(String[] args)
    {
        Scanner s1 = new Scanner(System.in);

        System.out.println("Enter the number of students:");
        int n = s1.nextInt();

        Internals[] CS = new Internals[n];
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter details for CIE student " + (i + 1));
            System.out.print("USN: ");
            String usn = s1.next();
            System.out.print("Name: ");
            String name = s1.next();
            System.out.print("Semester: ");
            int sem = s1.nextInt();
            System.out.println("Enter internal marks for 5 courses:");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++)
            {
                System.out.print("Course " + (j + 1) + ": ");
                internalMarks[j] = s1.nextInt();
            }
            CS[i] = new Internals(usn, name, sem, internalMarks);
        }

        External[] SS = new External[n];
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter details for SEE student " + (i + 1));
            System.out.print("USN: ");

```

```

String usn = s1.next();
System.out.print("Name: ");
String name = s1.next();
System.out.print("Semester: ");
int sem = s1.nextInt();
System.out.println("Enter SEE marks for 5 courses:");
int[] seeMarks = new int[5];
for (int j = 0; j < 5; j++)
{
    System.out.print("Course " + (j + 1) + ": ");
    seeMarks[j] = s1.nextInt();
}

SS[i] = new External(usn, name, sem, seeMarks);
}

int[][] finalMarks = new int[n][5];
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < 5; j++)
    {
        finalMarks[i][j] = CS[i].internalMarks[j] + SS[i].seeMarks[j];
    }
}

System.out.println("\nFinal Marks:");
for (int i = 0; i < n; i++)
{
    System.out.print("USN: " + CS[i].usn + ", Name: " + CS[i].name + ", Semester: " + CS[i].sem +
    ", Final Marks: ");
    for (int j = 0; j < 5; j++)
    {
        System.out.print(finalMarks[i][j] + " ");
    }
    System.out.println();
}
}
}

```

OUTPUT

```
D:\NotePad++\Java\Packages>javac CalculateFinalMarks.java

D:\NotePad++\Java\Packages>java CalculateFinalMarks
Enter the number of students:
1
Enter details for CIE student 1
USN: 1
Name: Ram
Semester: 3
Enter internal marks for 5 courses:
Course 1: 47
Course 2: 48
Course 3: 49
Course 4: 50
Course 5: 49
Enter details for SEE student 1
USN: 1
Name: Ram
Semester: 3
Enter SEE marks for 5 courses:
Course 1: 48
Course 2: 49
Course 3: 47
Course 4: 50
Course 5: 50

Final Marks:
USN: 1, Name: Ram, Semester: 3, Final Marks: 95 97 96 100 99
```

22/1/24

- 6) Create a package CIE which has two classes
- students and Internals. The class student has
members like usn, name, sem. The class
Internals has an array that stores the
internal marks scored in 5 courses of
current sem of student. Create another
package SEE which has class Internal
which stores marks scored in SEE in
5 courses.

Import the two packages in a file
that declare the final marks of
n students in all 5 ~~courses~~ courses.

→ import java.util.Scanner;
package CIE;
public class Student
{
 Scanner s1 = new Scanner(System.in);
 int usn;
 String name;
 int sem;
 public void Accept()
{
 System.out.print("Enter USN");
 usn = s1.nextInt();
 System.out.print("Enter name");
 name = s1.nextLine();
 System.out.print("Enter sem");
 sem = s1.nextInt();
 }
}

public void display()
{
 System.out.print("USN " + usn + " Name " + name +
 " SEM " + sem);
}
class Internals extends Student
{
 int marks = new int[5];
 public void accept()
 {
 System.out.print("Enter marks for each
course");
 for (int i = 0; i < 5; i++)
 {
 System.out.print(marks[i]);
 }
 }
}

```

package SEE;
public class external extends CIE.student
{
    system.out.print("enter the SEE marks for
final exam");
    int marks2[] = new int[5];
    public void accept()
    {
        system.out.print("enter marks");
        for(int i=0; i<5; i++)
        {
            marks2[i] = s1.nextInt();
        }
    }
    super.accept();
}
public void display()
{
    system.out.print("The SEE marks are");
    for(int i=0; i<5; i++)
    {
        system.out.print(marks2[i]);
    }
}

```

```

import CIE.student;
import CIE.internal;
import SEE.external;
public class StudDemo
{
    public static void main(string args[])
    {
        system.out.print("enter no. of students");
        scanner s1 = new scanner(system.in);
        internal i1[] = new internal[n];
        external e1[] = new external[n];
        system.out.print("enter " + n + " student internal
details");
        for(int i=0; i<n; i++)
        {
            i1[i] = new internals();
            i1[i].accept();
        }
        system.out.print("enter " + n + " student external
details");
        for(int i=0; i<n; i++)
        {
            e1[i] = new externals();
            e1[i].accept();
        }
    }
}

```

LABORATORY 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;

class WrongAge extends Exception
{
    public WrongAge()
    {
        super("Invalid age! Age cannot be negative nor zero.");
    }

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

class Father
{
    private int age;

    public Father(int age) throws WrongAge
    {
        if (age <= 0)
        {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge()
    {
        return age;
    }
}

class Son extends Father
{
    private int sonAge;
```

```

public Son(int fatherAge, int sonAge) throws WrongAge
{
    super(fatherAge);

    if (sonAge >= fatherAge)
    {
        throw new WrongAge("Son's age should be less than Father's age.");
    }

    this.sonAge = sonAge;
}

public int getSonAge()
{
    return sonAge;
}

public class InheritanceException
{
    public static void main(String[] args)
    {
        try
        {
            Scanner s1 = new Scanner(System.in);

            System.out.print("Enter Father's age: ");
            int fatherAge = s1.nextInt();
            Father f = new Father(fatherAge);

            System.out.print("Enter Son's age: ");
            int sonAge = s1.nextInt();
            Son s = new Son(f.getAge(), sonAge);

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

        } catch (WrongAge e)
        {
            System.out.println("Exception: " + e);
        } catch (Exception e)
        {
            System.out.println("Exception: Invalid input. Please enter valid integer values.");
        }
    }
}

```

OUTPUT

```
D:\NotePad++\Java>javac InheritanceException.java

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 46
Enter Son's age: 56
Exception: WrongAge: Son's age should be less than Father's age.

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 56
Enter Son's age: 24
Father's age: 56
Son's age: 24
```

WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derive class called "son" which extends base class.

In Father class, implement a constructor which takes age and throws the exception, WrongAge() when the input age < 0

In son class, implement a constructor that carries both father and throws exception if sons age is \geq fathers age

```
→ import java.util.Scanner;
class MyException extends Exception
{
    int detail;
    public MyException(int age, String exc)
    {
        this.detail = age;
        System.out.println(exc + " given age is : "
                           + age + " please enter again");
    }
}
```

```
public String getMessage()
{
    return "Exception:" + detail;
}
```

```
}
```

```
class Father {
    int age;
    public Father(int age) throws
        MyException {
```

exception class
↓
on.
ion

```
if (age < 0) { throw new;
MyException("Age cannot be lesser than 0");
this.age = age;
}
```

```
}
```

```
class son extends Father {
    int age;
    public son(int Father.Age, int son.Age)
        throws MyException {
        super(Father.Age);
        this.age = son.age;
        if (this.age > Father.age)
            throw new
MyException("Age of son can not be
more than fathers");
}
```

```
}
```

```
public class Father-son {
    public static void main(String[] args)
    {
        try {
            Father f1 = new Father(1);
            son s1 = new son(30, 31);
        }
        catch (MyException e)
        {
            System.out.println("Exception caught " + e.getMessage());
        }
    }
}
```

```
Enter Father's age : 46
Enter Son's age : 56
Exception: Wrong age: Son's age should be less
than Father's age
```

```
Enter Father's age : 56
```

```
Enter Son's age : 24
```

```
Father's age : 56
```

```
Son's age : 24
```

LABORATORY 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.

```
class BMSThread implements Runnable
{
    public void run()
    {
        while (true)
        {
            try
            {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
            catch (InterruptedException ie)
            {
                System.out.println("BMSThread is Interrupted");
            }
        }
    }
}

class CSEThread implements Runnable
{
    public void run()
    {
        while (true)
        {
            try
            {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
            catch (InterruptedException ie)
            {
                System.out.println("CSEThread is Interrupted");
            }
        }
    }
}

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

```
{  
    Thread bms = new Thread(new BMSThread());  
    Thread cse = new Thread(new CSEThread());  
    bms.start();  
    cse.start();  
}  
}
```

OUTPUT

```
D:\NotePad++\Java>javac Display.java
```

```
D:\NotePad++\Java>java Display  
BMS College of Engineering
```

```
CSE
```

```
BMS College of Engineering
```

```
CSE
```

```
BMS College of Engineering
```

```
CSE
```

```
CSE
```

```
|
```

05/07/22 Programs on Threads

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

→ class NewThread implements Runnable

```

    {
        Thread t;
        NewThread()
        {
            t = new Thread(this, "NewThread");
            System.out.println("CT:" + t);
            t.start();
        }

        public void run()
        {
            try
            {
                for(int n=5; n>0; n--)
                {
                    System.out.println("CSE:" + n);
                    Thread.sleep(2000);
                }
            }
            catch(InterruptedException ie)
            {
                System.out.println("Child Thread interrupted");
            }
            System.out.println("Child Thread quitting");
        }
    }
}

```

class Thread2

```

    {
        public static void main(String ss[])
        {
            new NewThread();
            System.out.println("Back in main");
            try
            {
                for(int n=5; n>0; n--)
                {
                    System.out.println("BMSCE:" + n);
                    Thread.sleep(10000);
                }
            }
            catch(InterruptedException ie)
            {
                System.out.println("Main Thread interrupted");
            }
            System.out.println("Main Thread quitting");
        }
    }
}

```

Output

```

Back in main
CSE : 5
BMSCE : 5
CSE : 4
CSE : 3
CSE : 2
CSE : 1
BMSCE : 4
Child Thread quitting
BMSCE : 3
BMSCE : 2
BMSCE : 1
Main Thread quitting

```

LABORATORY 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 Arithmetic Exception Display the exception in a message dialog box.

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

public class DivisionMain1 extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;

    public DivisionMain1()
    {
        setLayout(new FlowLayout());

        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("Result:",Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
        });
    }
}
```

```

        }
    });

}

public void actionPerformed(ActionEvent ae)
{
    int n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

            /*if(n2==0)
                throw new ArithmeticException();*/
            out=n1+" "+n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }

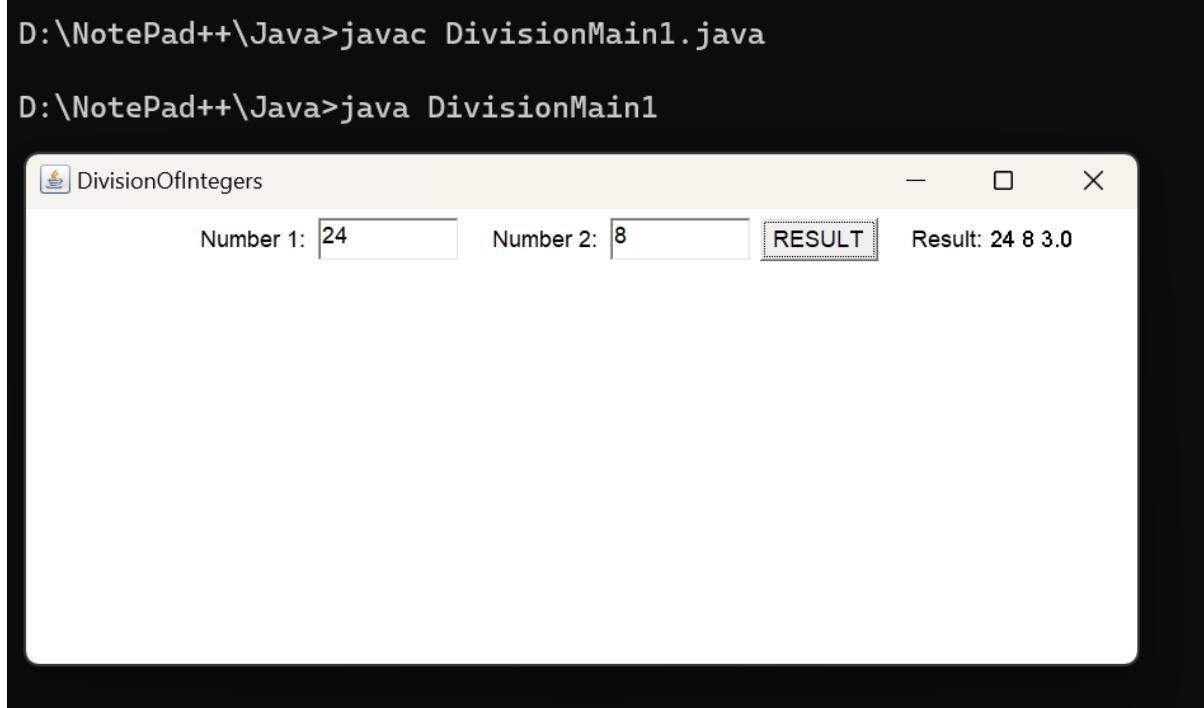
    }
    catch(NumberFormatException e1)
    {
        flag=1;
        out="Number Format Exception! "+e1;
        repaint();
    }
    catch(ArithmeticException e2)
    {
        flag=1;
        out="Divide by 0 Exception! "+e2;
        repaint();
    }
}

public void paint(Graphics g)
{
    if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-8);
    else
        g.drawString(out,100,200);
    flag=0;
}

```

```
public static void main(String[] args)
{
    DivisionMain1 dm=new DivisionMain1();
    dm.setSize(new Dimension(800,400));
    dm.setTitle("DivisionOfIntegers");
    dm.setVisible(true);
}
}
```

OUTPUT



Sah-9

Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num 1 and Num 2. The division of Num 1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num 1 or Num 2 were not an integer, the program would throw a NumberFormatException. If Num 2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

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

public class DivisionMain extends Frame implements ActionListener
{
    JTextField num1, num2;
    JButton dResult;
    JLabel outResult;
    String out = "";
    double resultNum;
    int flag = 0;

    public DivisionMain()
    {
        setLayout(new FlowLayout());

```

```

d.Result = new Button("RESULT");
Label number_1 = new Label("Number 1:", label.RIGHT);
Label number_2 = new Label("Number 2:", label.RIGHT);
num1 = new JTextField(5);
num2 = new JTextField(5);
outResult = new Label("Result:", label.RIGHT);
add(number_1);
add(num1);
add(number_2);
add(num2);
add(dResult);
add(outResult);
num1.addActionListener(this);
num2.addActionListener(this);
outResult.addActionListener(this);
addWindowListener(new WindowAdapter()
{
}

```

```
public void windowClosing(WindowEvent we)
{
    System.exit(0);
}
```

3) public void actionPerformed(ActionEvent ae)

```
{     int naga;  
try {  
    if (oxygenSource() == dResult)  
        f
```

```

n1 = Integer.parseInt(num1.getText());
n2 = Integer.parseInt(num2.getText());
/* if (n2==0)
    throw new ArithmeticException(); */
out = n1 + " + " + n2 + " = ";
resultNum = n1/n2;
out += String.valueOf(resultNum);
repaint();
}
}
catch (NumberFormatException e1)
{
    flag = 1;
    out = "Number Format Exception ! " + e1;
    repaint();
}
catch (ArithmaticException e2)
{
    flag = 1;
    out = "Divide by 0 Exception ! " + e2;
    repaint();
}

```

```

public void paint (Graphics g)
{
    if (flag == 0)
        g.drawString (outResult.getx () + outResult.getwidth () / 2,
                      outResult.gety () + outResult.getHeight () / 2);
    else
        g.drawString ("out>100,200",
                     flag = 0;
}

public static void main (String [] args)
{
    DimensionMain1 dm = new DimensionMain1 ();
    dm.setSize (new Dimension (800,400));
    dm.setTitle ("DimensionOfIntegers");
    dm.setVisible (true);
}

```

0/4

Number 1: Number 2:

RESULT Result: 24 8 3.0