

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



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

*Submitted by*

**Jayashree Tarai (24BECS410)**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)  
**BENGALURU-560019**

**Sep-2024 to Jan-2025**

**B.M.S. College of Engineering,  
Bull Temple Road, Bangalore 560019  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Jayashree Tarai (24BECS410)**, 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. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

## **Index**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	30/9/24	QUADRATIC EQUATION	4-6
2	07/10/24	STUDENT SGPA CALCULATOR	7-10
3	15/10/24	BOOK COLLECTION	11-13
4	22/10/24	SHAPE AREA CALCULATOR	14-16
5	28/10/24	BANK ACCOUNT MANAGER	17-22
6	12/11/24	FINAL MARKS CALCULATOR	23-27
7	22/12/24	INHERITANCE EXCEPTION	31-34
8	2/12/24	THREAD DEMO	35-37
9	2/12/24	DIVISION APP AWT	38-40
10			

**GITHUB LINK** <https://github.com/Jayashreecse/OOJ-javalab>

### Lab 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 discriminate  $b-4ac$  is negative, display a message stating that there are no real solutions.

Scanner input = new Scanner (System.in)  
System.out.println ("Enter value of a:");  
double a = input.nextInt();  
System.out.println ("Enter value of b:");  
double b = input.nextInt();  
System.out.println ("Enter value of c:");  
double c = input.nextInt();  
  
double formula1 = b\*b - 4\*a\*c  
if formula1 <= 0 {  
 System.out.println ("There are no real numbers");  
} else {  
 double root1 = (-b + Math.sqrt(b\*b - 4\*a\*c)) / (2\*a);  
 double root2 = (-b - Math.sqrt(b\*b - 4\*a\*c)) / (2\*a);  
 System.out.println ("The roots are: ");  
 System.out.println ("The root1: " + root1);  
 System.out.println ("The root2: " + root2);  
}

import java.util.Scanner;  
class product {  
 public static void main (String [] args)  
 {  
 Scanner input = new Scanner (System.in);  
 System.out.println ("Enter num1:");  
 int number1 = input.nextInt();  
 System.out.println ("Enter num2:");  
 int number2 = input.nextInt();  
 int product = number1 \* number2;  
 System.out.println (product);  
 }  
  
Output  
num1: 5  
num2: 5  
product: 25  
  
3) Lab program 1 (Quadratic Equation)  
import java.util.Scanner;  
import java.lang.Math;  
  
class Main {  
 public static void main (String args[]) {  
 }

```
import java.util.Scanner;  
import java.lang.Math;  
  
class Solu {  
  
    public static void main(String[] args) {  
        // Declare variables  
        double a, b, c, d, x1, x2;  
  
        // Create a Scanner object to read input  
        Scanner s = new Scanner(System.in);  
  
        // Get user input  
        System.out.println("Jayashree\n24BECS410\n3B batch 4\nEnter the values of a, b, and c:");
```

```

// Read the values of a, b, and c
System.out.println("enter a ");
a = s.nextDouble();
System.out.println("enter b ");
b = s.nextDouble();
System.out.println("enter c ");
c = s.nextDouble();

// Calculate the discriminant
d = b * b - 4 * a * c;

// Check if the discriminant is zero (for one real root)
if (d == 0) {
    x1 = x2 = -b / (2 * a);
    System.out.println("The roots are: " + x1 + " and " + x2);
}

// Check if the discriminant is positive (for two real roots)
else if (d > 0) {
    x1 = (-b + Math.sqrt(d)) / (2 * a);
    x2 = (-b - Math.sqrt(d)) / (2 * a);
    System.out.println("The roots are: " + x1 + " and " + x2);
}

// If the discriminant is negative (no real roots)
else {
    System.out.println("No real solutions. The discriminant is negative that is b2-4ac=less than
0.");
}

// Close the scanner
s.close();
}
}

```

## Quadratic equation Lab program 1

```
Jayashree
24BECS410
3B batch 4
Enter the values of a, b, and c:
enter a
1
enter b
0
enter c
1
No real solutions. The discriminant is negative that is b2-4ac=less than 0.
```

```
Jayashree
24BECS410
3B batch 4
Enter the values of a, b, and c:
enter a
1
enter b
3
enter c
-4
The roots are: 1.0 and -4.0
```

## Lab 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.

8/10/24 LAB Program -2

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

```
import java.util.Scanner;

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

    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 no. of subjects");
        n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
    }
}
```

```
for (int i = 0; i < n; i++) {
    System.out.print("Enter credits for sub" + (i + 1) +
                    " : ");
    Credits[i] = sc.nextInt();
}
System.out.println("Enter marks for sub" + (i + 1) +
                    " : ");
marks[i] = sc.nextInt();
}

void display() {
    System.out.println("Usn :" + usn);
    System.out.println("Name :" + name);
    System.out.println("Subject wise details");
    for (int i = 0; i < n; i++) {
        System.out.println("Subject " + (i + 1) +
                           " : credits = " + credits[i] +
                           ", Marks = " + marks[i]);
    }
}

double calculate_SGPA() {
    int totalCredits = 0;
    double totalpoints = 0.0;
    for (int i = 0; i < n; i++) {
        int grade_point = calculate
            grade_point(marks[i]);
    }
}
```

```

totalCredits += credits[i];
totalPoints += gradePoint * credits[i];
3 return totalPoints / totalCredits;

int calculateGradePoint (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 if (marks >= 30) return 4;
    else if (marks >= 20) return 3;
    else if (marks >= 10) return 2;
    else return 1;
}

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

```

```

import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n; // Number of subjects

    // Method to accept student details
    void acceptDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = sc.nextLine();
        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter number of subjects: ");
        n = sc.nextInt();

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

```

// Input credits and marks for each subject
for (int i = 0; i < n; i++) {
    System.out.print("Enter credits for subject " + (i+1) + ": ");
    credits[i] = sc.nextInt();
    System.out.print("Enter marks for subject " + (i+1) + ": ");
    marks[i] = sc.nextInt();
}

// Method to display student details
void displayDetails() {
    System.out.println("\nStudent Details:");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Subject-wise details:");
    for (int i = 0; i < n; i++) {
        System.out.println("Subject " + (i+1) + ": Credits = " + credits[i] + ", Marks = " +
marks[i]);
    }
}

// Method to calculate SGPA
double calculateSGPA() {
    int totalCredits = 0;
    double totalPoints = 0.0;

    for (int i = 0; i < n; i++) {
        int gradePoint = calculateGradePoint(marks[i]);
        totalCredits += credits[i];
        totalPoints += gradePoint * credits[i];
    }

    return totalPoints / totalCredits;
}

// Method to convert marks to grade point
int calculateGradePoint(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; // Fail
    }

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

```

## **STUDENT LAB PROGRAM 2**

```

Enter USN: 24BECS410
Enter Name: JAYASHREE
Enter number of subjects: 3
Enter credits for subject 1: 9
Enter marks for subject 1: 99
Enter credits for subject 2: 10
Enter marks for subject 2: 89
Enter credits for subject 3: 9
Enter marks for subject 3: 99

Student Details:
USN: 24BECS410
Name: JAYASHREE
Subject-wise details:
Subject 1: Credits = 9, Marks = 99
Subject 2: Credits = 10, Marks = 89
Subject 3: Credits = 9, Marks = 99
SGPA: 9.642857142857142

```

## Lab 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.

LAB Program -3

Create a class Book contains four members name, author, price, numPages. 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 all details. Develop a Java program to create n books.

```
public class Book_410 {  
  
    private String name;  
    private String author;  
    private double price;  
    private int numPg;  
  
    public Book_410(String name, String author, double price, int numPg) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPg = numPg;  
    }  
  
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }  
  
    public String getAuthor() { return author; }  
    public void setAuthor(String author) { this.author = author; }  
  
    public double getPrice() { return price; }  
    public void setPrice(double price) { this.price = price; }  
  
    public int getNumPg() { return numPg; }  
    public void setNumPg(int numPg) { this.numPg = numPg; }  
  
    @Override  
    public String toString() {  
        return "Book [name = " + name + ", Author = "  
            + author + ", Price = " + price + ",  
            numPg = " + numPg + "]";  
    }  
}
```

```
public String getAuthor() { return author; }  
public void setAuthor(String author) { this.  
    author = author; }  
  
public double getPrice() { return price; }  
public void setPrice(double price) { this.  
    price = price; }  
  
public int getNumPg() { return numPg; }  
public void setNumPg(int numPg) { this.  
    numPg = numPg; }  
  
@Override  
public String toString() {  
    return "Book [name = " + name + ", Author = "  
        + author + ", Price = " + price + ",  
        numPg = " + numPg + "]";  
}  
  
public static void main (String[] args) {  
    Book_410[] books = new Book_410[3];  
    books[0] = new Book_410 ("Book 1", "Author1",  
        299.12, 100);  
    books[1] = new Book_410 ("Book 2", "Author2",  
        188.9, 150);  
    books[2] = new Book_410 ("Book 3", "Author3",  
        347.9, 200);  
}
```

Read Values from the User:

```
For (Book book : books) {  
    System.out.println("book");  
}  
}  
  
Output:  
- Book [name = book 1, Author = author 1,  
    price = 299.12, number of pages = 100]  
Book [name = book 2, Author = author 2  
    price = 188.9, number of pages = 150]  
Book [name = book 3, Author = author 3  
    price = 347.9, number of pages = 200]  
  
Reading Values from the User  
Output  
Enter the number of books to be created: 2  
Enter the name of book 1: Mahatma Gandhi  
Enter the name of author: author1  
Enter the price of book: 200  
Enter the no. of pages of book: 150  
  
Enter the number of books to be created: 2  
Enter the name of book 2: Pandit Nehru  
Enter the name of author: author2  
Enter the price of book: 250  
Enter the no. of pages of book: 200
```

```

import java.util.ArrayList;
import java.util.Scanner;

public class Book410 {

    private String name;
    private String author;
    private double price;
    private int num_pg;

    public Book410(String name, String author, double price, int num_pg) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pg = num_pg;
    }

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

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

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

    public int getNumPg() { return num_pg; }
    public void setNumPg(int num_pg) { this.num_pg = num_pg; }

    @Override
    public String toString() {
        return "Book [Name=" + name + ", Author=" + author + ", Price=" + price + ", Number of Pages=" + num_pg + "]";
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Book410> books = new ArrayList<>();

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

        for (int i = 1; i <= n; i++) {
            System.out.print("Enter the name of the book " + i + ": ");
        }
    }
}

```

```

String name = scanner.nextLine();

System.out.print("Enter the name of the author " + i + ": ");
String author = scanner.nextLine();

System.out.print("Enter the price of the book " + i + ": ");
double price = scanner.nextDouble();

System.out.print("Enter the number of pages of the book " + i + ": ");
int num_pg = scanner.nextInt();
scanner.nextLine();
books.add(new Book410(name, author, price, num_pg));
}

System.out.println("Details of the books created:");
for (Book410 book : books) {
    System.out.println(book);
}

scanner.close();
}
}

```

```

C:\Users\bmsce\Documents\24becs410>javac Book410.java

C:\Users\bmsce\Documents\24becs410>java Book410
Enter the number of books to be created: 2
Enter the name of the book 1: gandhiji
Enter the name of the author 1: author1
Enter the price of the book 1: 190
Enter the number of pages of the book 1: 190
Enter the name of the book 2: panchatantra
Enter the name of the author 2: author2
Enter the price of the book 2: 150
Enter the number of pages of the book 2: 200
Details of the books created:
Book [Name=gandhiji, Author=author1, Price=190.0, Number of Pages=190]
Book [Name=panchatantra, Author=author2, Price=150.0, Number of Pages=200]

```

## Lab 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.

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.

```
abstract class Shape {  
    int dim1;  
    int dim2;  
    abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle (int length ,int breadth) {  
        dim1 = length ;  
        dim2 = breadth ;}  
    void printArea {  
        System.out.println ("the Rectangle =  
            " + dim1 * dim2 );}  
}
```

Class Triangle extends Shape {

```
Triangle C int base ,int height){  
    dim1 = base ;  
    dim2 = height ;  
    printArea (base,height);  
    System.out.println ("the area is "+  
        0.5* dim1* dim2 );}  
  
}  
  
Class Circle extends Shape { int radius  
Circle C int radius) {  
    dim1 = radius; radius = r;  
    void printArea () {  
        System.out.println ("the area of circle  
            is "+ 3.14 * Radius * Radius );}  
    }  
}  
  
Class MainMath {  
    public static void main (String args [] ) {  
        Rectangle R1 = new Rectangle (10,20);  
        R1.printArea ();  
        Triangle t1 = new triangle (10,20);  
        t1.printArea ();  
        Circle C1 = new circle (7);  
        C1.printArea ();  
    }  
}
```

### Code

```
abstract class Shape{  
int dim1;  
int dim2;  
  
abstract void printarea();  
  
}  
  
class Rectangle extends Shape {  
  
    Rectangle (int length ,int breadth){  
}
```

```

dim1=length;
dim2=breadth;
}

void printarea(){
System.out.println("the area of rectangle is "+ dim1*dim2);
}
}

class Triangle extends Shape {

Triangle (int base ,int height){
dim1=base;
dim2=height;
}

void printarea(){
System.out.println("the area of Triangle is "+ 0.5* dim1+dim2);
}
}

class Circle extends Shape {
int radius;
Circle (int r){
radius=r;
}

void printarea(){
System.out.println("the area of circle  is "+ 3.14*radius*radius);
}
}

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

Rectangle r1 =new Rectangle(10,20);
r1.printarea();

Triangle t1 =new Triangle(10,25);
t1.printarea();

Circle c1 =new Circle(6);
c1.printarea();
}
}

```

```
}
```

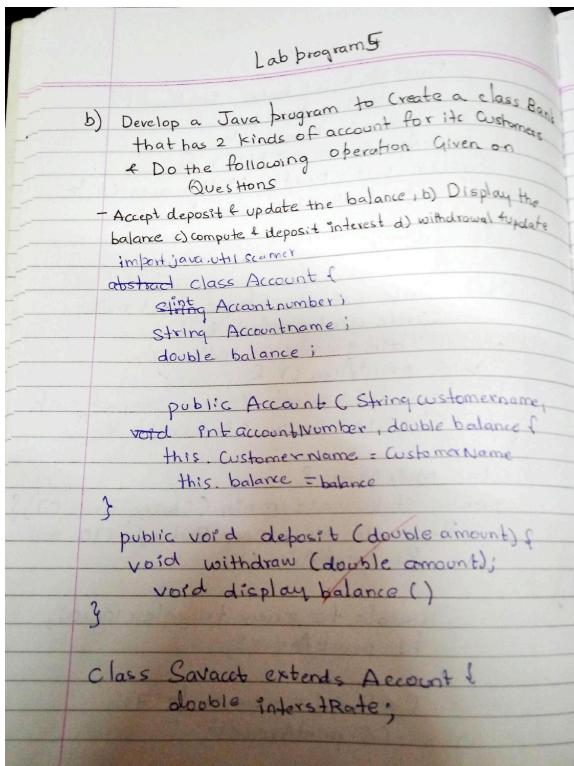
```
C:\Users\bmscse\Documents\jayashreecse>javac Mainmath.java
```

```
C:\Users\bmscse\Documents\jayashreecse>java Mainmath
the area of rectangle is 200
the area of Triangle is 5.025
the area of circle   is 113.0399999999999
```

## Lab 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.**



```
Savacct (String customerName, int
accountnumber,
Super (Accountnumber, Accountname,
balance),
this.interestrate = interestrate);

void deposit (double amount) {
    Super.deposit (amount);
    System.out.println ("Amount deposited to
Saving account + amount");

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

void displayBalance () {
    System.out.println ("Saving account balance",
balance);
}
```

```

void calculateInterest() {
    double interest = balance * interestRate / 100
    balance += interest
    System.out.println("Interest added to savings account: " + interest)
}

class CurrentAcc extends Account {
    double minimumBalance;
}

CurrentAcc (String customerName, int accountNumber, String accountType, double minimumBalance) {
    Super (customerName, customerAccountNumber, accountType);
    this.minimumBalance = minimumBalance;
}

void deposit(double amount) {
    super.deposit(amount);
    System.out.println("Amount deposited to current account: " + amount);
}

void withdrawal(double amount) {
    if (balance - amount >= withdrawalBalance) {
        balance -= amount;
    }
    System.out.println("Amount to be withdrawn");
}

```

```

From current account is "+ amount);
else {
    System.out.println("Insufficient balance in current account. Service charge applied");
    balance -= 50;
}

void displayBalance() {
    System.out.println("The amount is "+ amount);
}

class Bank {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter customer name: ");
        Scanner
        String customerName = scanner.nextLine();
        System.out.print ("Enter account number: ");
        int accNumber = scanner.nextInt();
        System.out.print ("Enter account type: ");
        Savings or Current);
        String acctype = scanner.next();
    }

    if (acctype.equalsIgnoreCase ("Savings")) {
        System.out.println ("Enter interest Rate");
    }
}

```

```

else if (acctype.equalsIgnoreCase ("Current")) {
    System.out.print ("Enter minimum balance: ");
    double minBal = scanner.nextDouble();
}
else {
    System.out.println ("Invalid opt");
    scanner.close();
}

Output
Enter customer name for saving account:
ABC
Enter acc number for saving account
723476
Enter Interest Rate
2
Deposited Amount(1000)
Deposited Amount 1000

```

```

import java.util.Scanner;

class Account {
    private String customerName;
    private String accountNumber;
    protected double balance;

    public Account(String customerName, String accountNumber) {

```

```

        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

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

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

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw amount: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal!");
        }
    }

    protected double getBalance() {
        return balance;
    }
}

class SavAcct extends Account {
    private double interestRate;

    public SavAcct(String customerName, String accountNumber, double interestRate) {
        super(customerName, accountNumber);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double currentBalance = getBalance();
        double interest = currentBalance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest deposited: " + interest);
    }
}

```

```

        }
    }

class CurAcct extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurAcct(String customerName, String accountNumber, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount) {
        if (getBalance() - amount < minimumBalance) {
            System.out.println("Service charge imposed: " + serviceCharge);
            deposit(-serviceCharge);
            System.out.println("Insufficient balance.");
        } else {
            super.withdraw(amount);
        }
    }
}

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

        System.out.println("Enter customer name for Savings Account:");
        String savingsCustomerName = scanner.nextLine();
        System.out.println("Enter account number for Savings Account:");
        String savingsAccountNumber = scanner.nextLine();
        System.out.println("Enter interest rate for Savings Account:");
        double interestRate = scanner.nextDouble();

        SavAcct savingsAccount = new SavAcct(savingsCustomerName, savingsAccountNumber,
        interestRate);
        savingsAccount.deposit(1000);
        savingsAccount.computeAndDepositInterest();
    }
}

```

```

savingsAccount.displayBalance();

System.out.println("Enter amount to withdraw from Savings Account:");
double withdrawAmount = scanner.nextDouble();
savingsAccount.withdraw(withdrawAmount);
savingsAccount.displayBalance();

scanner.nextLine();
System.out.println("Enter customer name for Current Account:");
String currentCustomerName = scanner.nextLine();
System.out.println("Enter account number for Current Account:");
String currentAccountNumber = scanner.nextLine();
System.out.println("Enter minimum balance for Current Account:");
double minimumBalance = scanner.nextDouble();
System.out.println("Enter service charge for Current Account:");
double serviceCharge = scanner.nextDouble();

CurAcct currentAccount = new CurAcct(currentCustomerName, currentAccountNumber,
minimumBalance, serviceCharge);
currentAccount.deposit(2000);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from Current Account:");
double currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from Current Account (may incur service
charge):");
currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

scanner.close();
}
}

```

```
E:\javaprograms>java Bank
Enter customer name for Savings Account:
abhi
Enter account number for Savings Account:
1234
Enter interest rate for Savings Account:
5
Deposited amount: 1000.0
Deposited amount: 50.0
Interest deposited: 50.0
Balance amount: 1050.0
Enter amount to withdraw from Savings Account:
100
Withdraw amount: 100.0
Balance amount: 950.0
Enter customer name for Current Account:
raaj
Enter account number for Current Account:
5678
Enter minimum balance for Current Account:
1900
Enter service charge for Current Account:
90
Deposited amount: 2000.0
Balance amount: 2000.0
Enter amount to withdraw from Current Account:
9000
Service charge imposed: 90.0
Deposited amount: -90.0
Insufficient balance.
Balance amount: 1910.0
Enter amount to withdraw from Current Account (may incur service charge):
800
Service charge imposed: 90.0
Deposited amount: -90.0
Insufficient balance.
Balance amount: 1820.0
```

## Lab program 6

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

12-11-24 Lab program 6

Create a package CIE which has two classes Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n student in all 5 Courses

```

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;
    }
}

```

2nd file // CIE / Internals.java

```

package CIE;

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

    public Internals (int[] marks) {
        for (int i = 0; i < 5; i++) {
            InternalsMarks[i] = marks[i];
        }
    }
}

// SEE / External.java

Package SEE;
import CIE.Student;

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

    public External (String usn, String name, int sem,
    int[] marks) {
        Super(usn, name, sem);
        for (int i = 0; i < 5; i++) {
            SeeMarks[i] = marks[i];
        }
    }
}

```

Main.java

```

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

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

        System.out.print ("Enter the number of students");
        int n = sc.nextInt();
        System.out.print ("Enter USN: ");
        String usn = sc.nextLine();
        System.out.print ("Enter Name: ");
        String name = sc.nextLine();
        System.out.print ("Enter Sem: ");
        int sem = sc.nextInt();

        int[] SeeMarks = new int [5];
        System.out.println ("Enter 5 See marks:");
        for (int j = 0; j < 5; j++) {
            SeeMarks[j] = sc.nextInt();
        }
        sc.nextLine();
    }
}

```

Internals internal = new Internals (marks);
External external = new External (usn, name, sem,
SeeMarks);

```

System.out.println ("Final marks for student");
name + " USN: " + usn + ">");
for (int j = 0; j < 5; j++) {
    int finalMarks = internal.InternalsMarks[j] +
    (SeeMarks[j] / 2);
    System.out.println ("Course " + (j + 1) + " : "
    + finalMarks);
}
System.out.println ();
sc.close();
}

To Execute
- Create a folder called Java programs
- In that create two folders which is the name of
package is CIE & SEE
- now write all code save in their particular
folder like Student & Internals in CIE
- And put the main file inside the Java
programs
Run code -> Javac cie/student.java
                package name > class name

```

Output  
 Enter the no. of students: 1  
 Enter USN:123  
 Enter Name: Sandhya  
 Enter Semester: 3  
 Enter 5 internal marks:  
 20 30 40 30 30  
 Enter 5 SEE marks  
 80 70 80 60 90  
 Final marks for student: Sandhya (USN:123)  
 Course 1: 60  
 Course 2: 65  
 Course 3: 60  
 Course 4: 60  
 Course 5: 75  
 2) → Student & Family package

```

  package student;
  public class Student {
    public String name;
    public int age;
    public String branch;

    public Student (String name, int age,
      String branch);
  }
  
```

```

// CIE/Student.java
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;
  }
}

// CIE/Internals.java
package CIE;

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

  public Internals(int[] marks) {
  }
}
  
```

```

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

// SEE/External.java
package SEE;
import CIE.Student;

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

    public External(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        for (int i = 0; i < 5; i++) {
            seeMarks[i] = marks[i];
        }
    }
}

// Main.java
import CIE.*;
import SEE.External;
import java.util.Scanner;

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

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

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

            System.out.print("Enter USN: ");
            String usn = sc.nextLine();
        }
    }
}

```

```

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

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

int[] internalMarks = new int[5];
System.out.println("Enter 5 internal marks:");
for (int j = 0; j < 5; j++) {
internalMarks[j] = sc.nextInt();
}
sc.nextLine();
int[] seeMarks = new int[5];
System.out.println("Enter 5 SEE marks:");
for (int j = 0; j < 5; j++) {
seeMarks[j] = sc.nextInt();
}
sc.nextLine();

Internals internal = new Internals(internalMarks);
External external = new External(usn, name, sem, seeMarks);

System.out.println("\nFinal Marks for Student: " + name + " (USN: " + usn + ")");
for (int j = 0; j < 5; j++) {
int finalMark = internal.internalMarks[j] + (seeMarks[j] / 2);
System.out.println("Course " + (j + 1) + ": " + finalMark);
}
System.out.println();
sc.close();
}
}

```

## Output:

```
E:\>cd E:\javaprograms\  
E:\javaprograms>javac CIE/Student.java CIE/Internals.java SEE/External.java Main.java  
E:\javaprograms>java Main  
Enter the number of students: 2  
Enter USN: 123  
Enter Name: Sandhya  
Enter Semester: 3  
Enter 5 internal marks:  
20 30 40 30 30  
Enter 5 SEE marks:  
80 70 80 60 90  
  
Final Marks for Student: Sandhya (USN: 123)  
Course 1: 60  
Course 2: 65  
Course 3: 80  
Course 4: 60  
Course 5: 75
```

## Lab Program on interface

Program 5 (polygon)

```

interface Polygon{
    default void getPerimeter(){
        System.out.println("calculating the perimeter");
    }

    void getArea();
}

class Rectangle implements Polygon{
    int l=5;
    int b=10;

    public void getArea(){
        System.out.println("calculating the area: "+l*b);
    }
}

class Circle implements Polygon{
    public void getArea(){
        int r=5;
        System.out.println("calculating the area: "+3.14*r*r);
    }
}

```

```

public class Test 1 {
    public static void main (String args[]){
        Polygon p1 = new Rectangle ();
        p1.getArea();
        p1.getPerimeter();

        Polygon c1 = new Circle ();
        c1.getArea();
        c1.getPerimeter();
    }
}

Output
Calculating the area of rectangle: 50
Calculating the Perimeter→ 30
Calculating the area of circle: 31.40002
Calculating the perimeter → 78.5398163397

```

You can add /call the getperimeter() method in both classes

```

→ default void getPerimeter(){
    System.out.println("calculating perimeter");
}

In Rectangle class (Add this)

public void getPerimeter(){
    int l=5;
    int b=10;
    System.out.println("calculating the perimeter of rectangle: "+(2*(l+b)));
}

In Circle class (Add this)

public void getPerimeter(){
    int r=5;
    System.out.println("calculating the perimeter of the circle: "+(2*3.14*r));
}

```

```

interface Polygon {

    default void getperimeter() {
        System.out.println("Calculating the perimeter");
    }
    void getarea();
}

class Rectangle implements Polygon{
    int l=5;
    int b=10;

    public void getarea(){
        System.out.println("calculuating the area of rectangle : "+l*b);
    }

    public void getperimeter() {
        int l = 5;
        int b = 10;
        System.out.println("Calculating the perimeter of the rectangle: " + (2 * (l + b)));
    }
}

class Circle implements Polygon{
    int r=5;
    public void getarea(){
        System.out.println("calculuating the area of circle: "+Math.PI * r*r);

    }

    public void getperimeter() {
        int r = 5; // radius
        System.out.println("Calculating the perimeter of the circle: " + (2 * Math.PI * r));
    }
}

public class Test1{
    public static void main (String args[]){

        Polygon p1=new Rectangle();
        p1.getarea();
    }
}

```

```
p1.getperimeter();  
  
Polygon c1=new Circle();  
c1.getarea();  
c1.getperimeter();  
  
}  
}
```

```
C:\Users\Admin\Documents\jayashree>java Test1  
calcuating the area of rectangle : 50  
calcuating the perimerter  
calcuating the area of circle: 78.53981633974483  
calcuating the perimerter
```

## Lab Program 8

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

Output

Program on Exception

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

```

import java.util.Scanner;

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

class Father {
    int fatherage;
    public Father(int age) throws Wrongage {
        if (age < 0) {
            throw new Wrongage("father's age cannot be negative!");
        }
        this.fatherage = age;
        System.out.println("father's age: " + this.fatherage);
    }
}

class Son extends Father {
    int sonage;
    public Son(int fatherage, int sonage) throws Wrongage {
        Super(fatherage);
        if (sonage < 0) {
            throw new Wrongage("son's age cannot be negative!");
        }
        if (sonage >= fatherage) {
            throw new Wrongage("son's age cannot be greater than or equal to Father's age!");
        }
    }
}

```

Class Father {  
 int fatherage;  
 public Father(int age) throws Wrongage {  
 if (age < 0) {  
 throw new Wrongage ("father's age cannot be negative!");  
 }  
 this.fatherage = age;  
 System.out.println ("father's age: " + this.fatherage);  
 }  
}

Class Son extends Father {  
 int sonage;  
 public Son(int fatherage, int sonage) throws Wrongage {  
 Super(fatherage);  
 if (sonage < 0) {  
 throw new Wrongage ("son's age cannot be negative!");  
 }  
 if (sonage >= fatherage) {  
 throw new Wrongage ("son's age cannot be greater than or equal to Father's age!");  
 }
 }
}

```

this.sonage = sonage;
System.out.println ("son's age: " + this.sonage);
}

public class Main {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        try {
            System.out.print ("Enter father's age: ");
            int fatherage = scanner.nextInt();
            System.out.print ("Enter son's age: ");
            int sonage = scanner.nextInt();
            Son son = new Son (fatherage, sonage);
        } catch (Wrongage e) {
            System.out.println ("Exception: " + e.getMessage ());
        } finally {
            scanner.close ();
        }
    }
}

```

Output

case 1  
- Enter father's age: 45  
Enter son's age: 30  
Father's age: 45  
Son's age: 30

case 2  
Enter Father's age: 45  
Enter Son's age: 46  
- Father's age: 45  
Exception: Son's age cannot be greater than or equal to Father's Age

case 3  
Enter Father's age: -1  
Enter Son's age: 20  
Exception: Father's age cannot be negative

case 4  
Enter Father's age: 40  
Enter Son's age: -30

```

import java.util.Scanner;

// Define the WrongAge exception
class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

// Base class Father
class Father {
    int fatherAge;

    // Constructor for Father class
    public Father(int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge("Father's age cannot be negative!");
        }
        this.fatherAge = age;
        System.out.println("Father's age: " + this.fatherAge);
    }
}

// Derived class Son
class Son extends Father {
    int sonAge;

    // Constructor for Son class
    public Son(int fatherAge, int sonAge) throws WrongAge {
        super(fatherAge); // Call Father's constructor
        if (sonAge < 0) {
            throw new WrongAge("Son's age cannot be negative!");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age!");
        }
        this.sonAge = sonAge;
    }
}

```

```

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

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

        // Get input from user
        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();

            // Create Son object which also checks for Father's age and Son's age
            Son son = new Son(fatherAge, sonAge);
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}

```

```
E:\javaprograms>java Main2
Enter Father's age: 30
Enter Son's age: 35
Father's age: 30
Exception: Son's age cannot be greater than or equal to Father's age!

E:\javaprograms>java Main2
Enter Father's age: 40
Enter Son's age: 25
Father's age: 40
Son's age: 25

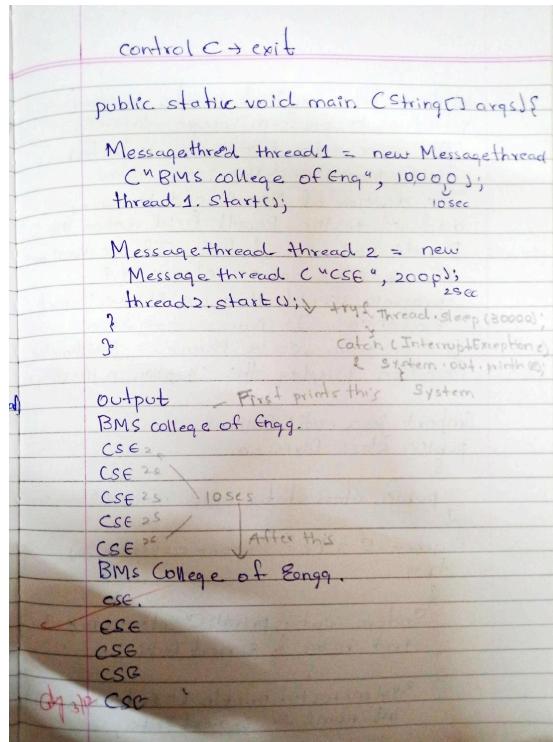
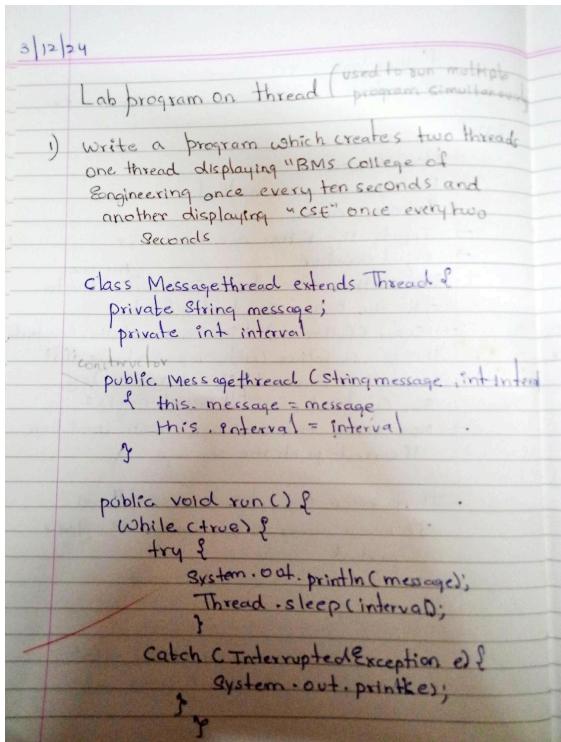
E:\javaprograms>java Main2
Enter Father's age: 0
Enter Son's age: 45
Father's age: 0
Exception: Son's age cannot be greater than or equal to Father's age!

E:\javaprograms>java Main2
Enter Father's age: -90
Enter Son's age: 67
Exception: Father's age cannot be negative!

E:\javaprograms>java Main2
Enter Father's age: 30
Enter Son's age: -25
Father's age: 30
Exception: Son's age cannot be negative!
```

## Lab Program 9

**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 Messagethread extends Thread {  
    private String message;  
    private int interval;  
  
    public Messagethread(String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    @Override  
    public void run() {  
        while (true) {  
            try {
```

```

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

public static void main(String[] args) {
    Messagethread thread1 = new Messagethread("BMS College of Engineeringg",
10000); // 10000 milliseconds = 10 seconds
    thread1.start();

    Messagethread thread2 = new Messagethread("CSE", 2000); // 2000 milliseconds =
2 seconds
    thread2.start();

    try {
        Thread.sleep(30000); // Let the threads run for 30 seconds
    } catch (InterruptedException e) {
        System.out.println(e);
    }

    // Exit the program after 30 seconds (this will terminate all threads)
    System.out.println("stopped");
    System.exit(0);

}
}

```

```
E:\javaprograms>javac Messagethread.java  
E:\javaprograms>java Messagethread  
BMS College of Engineeringg  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineeringg  
CSE  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineeringg  
CSE  
CSE  
CSE  
CSE  
stopped
```

## Lab Program 9

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

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

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

        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2, 10, 10));

        JLabel labelNum1 = new JLabel("Num1:");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Num2:");
        JTextField textNum2 = new JTextField();
        JLabel labelResult = new JLabel("Result:");
        JTextField textResult = new JTextField();
        textResult.setEditable(false);

        JButton divideButton = new JButton("Divide");

        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```

```

try {
    int num1 = Integer.parseInt(textNum1.getText());
    int num2 = Integer.parseInt(textNum2.getText());

    int result = num1 / num2;

    textResult.setText(String.valueOf(result));
} catch (NumberFormatException ex) {
    JOptionPane.showMessageDialog(frame, "Please enter valid integers.", "Number
Format Exception", JOptionPane.ERROR_MESSAGE);
} catch (ArithmaticException ex) {
    JOptionPane.showMessageDialog(frame, "Division by zero is not allowed.", "Arithmatic Exception", JOptionPane.ERROR_MESSAGE);
}
};

panel.add(labelNum1);
panel.add(textNum1);
panel.add(labelNum2);
panel.add(textNum2);
panel.add(labelResult);
panel.add(textResult);
panel.add(divideButton);

frame.add(panel);

frame.setVisible(true);
}
}

```

```
E:\javaprograms>javac DivisionApp.java
```

```
E:\javaprograms>java DivisionApp
```

