

JAVA LAB PROGRAMS

Name : Kathasagaram Aishwarya

USN : 1BM22CS123

Section : 3C

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.

Code:

```
/* Develop a Java program that prints all real solutions to the quadratic
equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the
discriminant b2-4ac is negative, display a message stating that there are no
real solutions. */
import java.util.*;
import java.math.*;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;

    void coeff()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter cooefficients a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        d = (b*b) - (4*a*c);
    }
    void evalu()
    {
        while(a==0)
        {
            System.out.println("Not a QE.");
            System.out.println("Enter non zero coefficient");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
        if(d==0)
        {
            System.out.println("Roots are real and equal.");
            r1 = (-b)/(2*a);
            System.out.println("Root1=Root2="+r1);
        }
        else
        {
            double p,q;
            p = (-b + Math.sqrt(d))/(2*a);
            q = (-b - Math.sqrt(d))/(2*a);
            System.out.println("Root1=" + p);
            System.out.println("Root2=" + q);
        }
    }
}
```

```

        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1="+r1+"+i"+r2);
            System.out.println("Root2="+r1+"-i"+r2);
        }
        else
        {
            System.out.println("Roots are real and distinct");
            r1 = (-b+(Math.sqrt(d)))/(2*a);
            r2 = (-b-(Math.sqrt(d)))/(2*a);
            System.out.println("root1= "+r1+"root2= "+r2);
        }
    }

}
class QuadraticEq
{
    public static void main(String sx[])
    {
        Quadratic q = new Quadratic();
        q.coeff();
        q.evalu();
    }
}

```

Output:

```

● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % javac QuadraticEq.java
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % java QuadraticEq
Enter cooefficients a,b,c
1 2 1
Roots are real and equal.
Root1=Root2=-1.0
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % javac QuadraticEq.java
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % java QuadraticEq
Enter cooefficients a,b,c
1 -7 12
Roots are real and distinct
root1= 4.0root2= 3.0
○ (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop %

```

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

Code:

```

/* Develop a Java program to create a class Student with members usn, name, an array
credits and array marks. Include methods to accept and display details and a

```

Object Oriented Java Programming - 23CS3PCOOJ

```
method to calculate SGPA of a student. */
import java.util.Scanner;
class Student {
    String usn,name;
    int[] credits,marks;
    // Method to accept details of a student
    void acceptDetails() {
        Scanner hello = new Scanner(System.in);
        System.out.print("Enter USN: ");
        this.usn = hello.next();
        System.out.print("Enter Name: ");
        this.name = hello.next();
        credits = new int[4];
        marks = new int[4];
        System.out.println("Enter details of credits and marks in order for 4 subjects:");
        for (int i = 0; i < 4; i++) {
            System.out.print("Enter credits for Subject " + (i + 1) + ": ");
            credits[i] = hello.nextInt();
            System.out.print("Enter marks for Subject " + (i + 1) + ": ");
            marks[i] = hello.nextInt();
        }
    }
    // Method to display details of a student
    void display() {
        System.out.println("Student Details.");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        for (int i = 0; i < 4; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] +
", Marks: " + marks[i]);
        }
    }
    // Method to calculate SGPA of a student
    double calc() {
        int totalCredits = 0;
        double totalGradePoints = 0;
        for (int i = 0; i < 4; i++) {
            totalCredits += credits[i];
            totalGradePoints += gradePoints(marks[i]) * credits[i];
        }
        return totalGradePoints / totalCredits;
    }
    // Method to calculate grade points based on marks
    int gradePoints(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 class SGPA {
```

```
public static void main(String[] args) {  
    // Example usage of the Student class  
    Student student = new Student();  
    student.acceptDetails();  
    System.out.println("\nStudent Details:");  
    student.display();  
    System.out.println("\nSGPA: " + student.calc());  
}  
}
```

Output:

```
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % javac SGPA.java  
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % java SGPA  
Enter USN: 123  
Enter Name: Aish  
Enter details of credits and marks in order for 4 subjects:  
Enter credits for Subject 1: 4  
Enter marks for Subject 1: 99  
Enter credits for Subject 2: 3  
Enter marks for Subject 2: 91  
Enter credits for Subject 3: 1  
Enter marks for Subject 3: 83  
Enter credits for Subject 4: 2  
Enter marks for Subject 4: 78  
  
Student Details:  
Student Details.  
USN: 123  
Name: Aish  
Subject 1 - Credits: 4, Marks: 99  
Subject 2 - Credits: 3, Marks: 91  
Subject 3 - Credits: 1, Marks: 83  
Subject 4 - Credits: 2, Marks: 78  
  
SGPA: 9.5
```

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.

Code:

```
/* 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
```

Object Oriented Java Programming - 23CS3PCOOJ

```
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 numPages;  
    // Constructor to set values for the members  
    public Book(String name, String author, double price, int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
    // Method to set details of the book  
    public void setDetails() {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter book name: ");  
        this.name = scanner.nextLine();  
        System.out.print("Enter author name: ");  
        this.author = scanner.nextLine();  
        System.out.print("Enter price: ");  
        this.price = scanner.nextDouble();  
        System.out.print("Enter number of pages: ");  
        this.numPages = scanner.nextInt();  
    }  
    // Method to get details of the book  
    public void getDetails() {  
        System.out.println("Book Name: " + name);  
        System.out.println("Author: " + author);  
        System.out.println("Price: $" + price);  
        System.out.println("Number of Pages: " + numPages);  
    }  
    // toString method to display complete details of the book  
    public String toString() {  
        return "Book Details:\n" +  
            "Name: " + name + "\n" +  
            "Author: " + author + "\n" +  
            "Price: $" + price + "\n" +  
            "Number of Pages: " + numPages;  
    }  
}  
public class Books {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of books: ");  
        int n = scanner.nextInt();  
        // Creating an array to store n book objects  
        Book[] books = new Book[n];  
        // Creating n book objects and setting their details  
        for (int i = 0; i < n; i++) {  
            System.out.println("\nEnter details for Book " + (i + 1) + ":");  
        }  
    }  
}
```

```
        books[i] = new Book("", "", 0.0, 0);
        books[i].setDetails();
    }
    // Displaying details of all the books
    System.out.println("\nDetails of all books:");
    for (int i = 0; i < n; i++) {
        System.out.println("\nBook " + (i + 1) + ":");

        books[i].getDetails();
    }
    // Using the toString method to display complete details of the books
    System.out.println("\nComplete details of all books:");
    for (int i = 0; i < n; i++) {
        System.out.println("\nBook " + (i + 1) + ":\n" +
books[i].toString());
    }
}
```

Output:

```
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % javac Books.java
● (base) aishwaryakathasagaram@Aishus-MacBook-Air desktop % java Books
Enter the number of books: 1

Enter details for Book 1:
Enter book name: JAVA
Enter author name: abc
Enter price: 20
Enter number of pages: 50

Details of all books:

Book 1:
Book Name: JAVA
Author: abc
Price: $20.0
Number of Pages: 50

Complete details of all books:

Book 1:
Book Details:
Name: JAVA
Author: abc
Price: $20.0
Number of Pages: 50
```

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.

Code:

```
import java.util.Scanner;

abstract class Shape{
    double a,b;
    abstract void printArea();
}

class Rectangle extends Shape{
    void printArea() {
        System.out.println("Area of rectangle is "+(a*b));
    }
}

class Triangle extends Shape{
    void printArea() {
        System.out.println("Area of triangle is "+(0.5*a*b));
    }
}

class Circle extends Shape{
    void printArea() {
        System.out.println("Area of circle is "+(3.14*a*a));
    }
}

class Shapearea{
    public static void main(String args[]){
        int n;
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();

        while(true){
            Scanner s1 = new Scanner(System.in);
            System.out.println("\nMAIN MENU\nSelect shape\n1. Rectangle\n2. Triangle\n3. circle\nEnter choice:");
            n = s1.nextInt();

            switch(n) {

                case 1:
                {
                    System.out.print("Enter length: ");
                    r.a = s1.nextDouble();
                    System.out.print("Enter breadth: ");
                }
            }
        }
    }
}
```

```
        r.b = s1.nextDouble();
        r.printArea();
        break;
    }
    case 2:
    {
        System.out.print("Enter length: ");
        t.a = s1.nextDouble();
        System.out.print("Enter breadth: ");
        t.b = s1.nextDouble();
        t.printArea();
        break;
    }
    case 3:
    {
        System.out.print("\nEnter radius: ");
        c.a = s1.nextDouble();
        c.printArea();
        break;
    }
}

default: System.out.println("Invalid input");
}
}
}
}
```

Output:

```
● (base) aishwaryakathasagaram@Aishus-MacBook-Air downloads % javac Sarea.java
○ (base) aishwaryakathasagaram@Aishus-MacBook-Air downloads % java Sarea
Enter following choices to find area of shapes of your choice 1.Rectangle 2.Triangle 3.Circle
1
Enter the length and breadth of the rectangle:
4 8
Area of Rectangle is 32.0
Enter following choices to find area of shapes of your choice 1.Rectangle 2.Triangle 3.Circle
2
Enter the height and breadth of the triangle:
6 8
Area of Triangle is 24.0
Enter following choices to find area of shapes of your choice 1.Rectangle 2.Triangle 3.Circle
3
Enter the radius of the circle:
1
Area of Circle is 3.141592653589793
```

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.

Code:

```
class Bank {  
    public static void main(String[] args) {  
        SavingsAccount savingsAccount = new SavingsAccount("John Doe", "SA1001");  
        CurrentAccount currentAccount = new CurrentAccount("Jane Smith", "CA2002");  
        // Perform operations on savings account  
        savingsAccount.deposit(5000);  
        savingsAccount.displayBalance();  
        savingsAccount.computeInterest();  
        savingsAccount.displayBalance();  
        savingsAccount.withdraw(2000);  
        savingsAccount.displayBalance();  
        // Perform operations on current account  
        currentAccount.deposit(8000);  
        currentAccount.displayBalance();  
  
        currentAccount.withdraw(5000);  
        currentAccount.displayBalance();  
    }  
}  
  
class Account {  
    protected String customerName;  
    protected String accountNumber;  
    protected double balance;  
    public Account(String customerName, String accountNumber) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.balance = 0;  
    }  
    public void deposit(double amount) {  
        balance += amount;  
    }  
    public void withdraw(double amount) {  
        if (balance < amount) {  
            System.out.println("Insufficient balance");  
        } else {  
            balance -= amount;  
        }  
    }  
    public void displayBalance() {  
        System.out.println("Customer Name: " + customerName + "  
        Account Number: " + accountNumber + "  
        Balance: " + balance);  
    }  
}
```

Object Oriented Java Programming - 23CS3PCOOJ

```
System.out.println("Deposit of $" + amount + " successful");
}
public void displayBalance() {
System.out.println("Account Number: " + accountNumber + "\nBalance: " +
balance);
}
}
class SavingsAccount extends Account {
public SavingsAccount(String customerName, String accountNumber) {
super(customerName, accountNumber);
}
public void computeInterest() {
double interestRate = 0.05; // Assuming a 5% interest rate
double interest = balance * interestRate;
balance += interest;
System.out.println("Interest of $" + interest + " computed and added to the
balance.");
}
public void withdraw(double amount) {
if (balance >= amount) {
balance -= amount;
System.out.println("Withdrawal of " + amount + " successful");
} else {
System.out.println("Insufficient funds for withdrawal.");
}
}
}
class CurrentAccount extends Account {
private double minimumBalance = 1000; // Assuming a minimum balance
requirement
public CurrentAccount(String customerName, String accountNumber) {
super(customerName, accountNumber);
}
public void withdraw(double amount) {
if (balance - amount >= minimumBalance) {
balance -= amount;
System.out.println("Withdrawal of " + amount + " successful.");
} else {
System.out.println("Insufficient funds. Service charge applied.");
imposeServiceCharge();
}
}
private void imposeServiceCharge() {
double serviceCharge = 20; // Assuming a service charge of $20
balance -= serviceCharge;
System.out.println("Service charge of $" + serviceCharge + " imposed.");
}
}
```

Output:

```
[Running] cd "/Users/aishwaryakathasagaram/Downloads/" && javac Bank.java && java Bank
Deposit of $5000.0 successful
Account Number: SA1001
Balance: 5000.0
Interest of $250.0 computed and added to the balance.
Account Number: SA1001
Balance: 5250.0
Withdrawal of 2000.0 successful
Account Number: SA1001
Balance: 3250.0
Deposit of $8000.0 successful
Account Number: CA2002
Balance: 8000.0
Withdrawal of 5000.0 successful.
Account Number: CA2002
Balance: 3000.0
```

6. Create a package CIE which has two classes- Student and Internals. The class Student 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.

Code:

Pack/CIE/Internal.Java

```
package CIE;

import java.util.Scanner;

public class Internal extends CIE.Student{
    public int m[] = new int[5];
    CIE.Student student = new CIE.Student();
    public void accept(){
        student.accept();

        Scanner s1 = new Scanner(System.in);
```

Object Oriented Java Programming - 23CS3PCOOJ

```
System.out.println("Enter Internal Marks:");
for(int i=0;i<5;i++) {
    m[i] = s1.nextInt();
}
}

public void display(){

    student.display();
    for(int i=0;i<5;i++){
        System.out.println("Marks of sub" + (i+1) + " = " + m[i]);
    }
}
}
```

Pack/CIE/Student.Java

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

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

    public void accept(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name:");
        this.name = s.nextLine();
        System.out.println("Enter usn:");
        this.usn = s.nextLine();
        System.out.println("Enter sem");
        this.sem = s.nextInt();
    }

    public void display(){
        System.out.println("Name: " + this.name + "\nUSN: " + this.usn +
"\nSem: " + this.sem);
    }
}
```

Pack/SEE/External.Java

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

import CIE.Internal;
import CIE.Student;

public class External extends CIE.Student{
    public int x[] = new int[5];
```

```

public void accept(){
    Scanner s2 = new Scanner(System.in);
    System.out.println("Enter External Marks:");
    for(int i=0;i<5;i++) {
        x[i] = s2.nextInt();
    }
}

public void display(){
    super.display();

    for(int i=0;i<5;i++){
        System.out.println("Marks of sub" + (i+1) + " = " + x[i]);
    }
}
}

```

Pack/Final.Java

```

import java.util.Scanner;

import CIE.Student;
import CIE.Internal;
import SEE.External;

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

        System.out.println("Enter n:");
        int y = n.nextInt();

        CIE.Internal[] c1 = new CIE.Internal[y];
        SEE.External[] c2 = new SEE.External[y];

        for(int i=0;i<y;i++){
            c1[i] = new CIE.Internal();
            c2[i] = new SEE.External();

            c1[i].accept();
            c2[i].accept();

            // c1[i].accept();c2[i].accept();
            c1[i].display();c2[i].display();

            for(int j=0;j<5;j++){
                double calc = c1[i].m[j]+((c2[i].x[j])/2);
                System.out.println("Final marks of sub["+ (i+1) +"]= "+calc);
            }
        }
    }
}

```

Output:

```

(base) aishwaryakathasagaram@Aishus-MacBook-Air Aish.java % javac CIE/Student.java
(base) aishwaryakathasagaram@Aishus-MacBook-Air Aish.java % javac CIE/Internal.java
(base) aishwaryakathasagaram@Aishus-MacBook-Air Aish.java % javac SEE/External.java
(base) aishwaryakathasagaram@Aishus-MacBook-Air Aish.java % javac Final.java
(base) aishwaryakathasagaram@Aishus-MacBook-Air Aish.java % java Final
Enter n:
5
Enter Name:
Aish
Enter usn:
123
Enter sem
3
Enter Internal Marks:
50 49 47 39 35
Enter External Marks:
100 100 100 100 100
Name: Aish
USN: 123
Sem: 3
Marks of sub1 = 50
Marks of sub2 = 49
Marks of sub3 = 47
Marks of sub4 = 39
Marks of sub5 = 35
Name: null
USN: null
Sem: 0
Marks of sub1 = 100
Marks of sub2 = 100
Marks of sub3 = 100
Marks of sub4 = 100
Marks of sub5 = 100
Final marks of sub[1]= 100.0
Final marks of sub[1]= 99.0
Final marks of sub[1]= 97.0
Final marks of sub[1]= 89.0
Final marks of sub[1]= 85.0

```

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.

Code:

```

/* 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

```

Object Oriented Java Programming - 23CS3PCOOJ

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

class Father {
    private int fatherAge;

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

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age should be less than
Father's age");
        }
        this.sonAge = sonAge;
        System.out.println("Father's Age: " + fatherAge);
        System.out.println("Son's Age: " + sonAge);
    }
}

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

        try {
```

```
        System.out.print("Enter Father's Age: ");
        int fatherAge = scanner.nextInt();
        Father father = new Father(fatherAge);

        System.out.print("Enter Son's Age: ");
        int sonAge = scanner.nextInt();
        Son son = new Son(fatherAge, sonAge);

    } catch (WrongAgeException e) {
        System.out.println("Exception: " + e.getMessage());
    }
}

}
```

Output:

```
● (base) aishwaryakathasaram@Aishus-MacBook-Air desktop % javac ExceptionInheritanceDemo.java
● (base) aishwaryakathasaram@Aishus-MacBook-Air desktop % java ExceptionInheritanceDemo
Enter Father's Age: 40
Enter Son's Age: 41
Exception: Son's age should be less than Father's age
```

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.

Code:

```
class Thread1 extends Thread{
    public void run() {
        while(true) {
            try{
                for(int i=0;i<5;i++) {
                    System.out.println("BMSCE");
                    Thread.sleep(1000);
                }
            }catch(InterruptedException e){System.out.println("BMSCE IE");}
        }
    }
}

class Thread2 extends Thread{
    public void run() {
        while(true) {
```

```
try{
    for(int i=0;i<5;i++){
        System.out.println("CSE");
        Thread.sleep(1500);
    }

}catch(InterruptedException e){System.out.println("CSE IE");}
}

}

public class multi{
public static void main(String[] args) {
    Thread1 t1 = new Thread1();
    Thread2 t2 = new Thread2();
    t1.start();t2.start();
}
}
```

Output:

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.

Code:

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

class SwingDemo {
    SwingDemo() {
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and dividend:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        // add in order :)
        jfrm.add(err); // to display error bois
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
```

Object Oriented Java Programming - 23CS3PCOOJ

```
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

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

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

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;

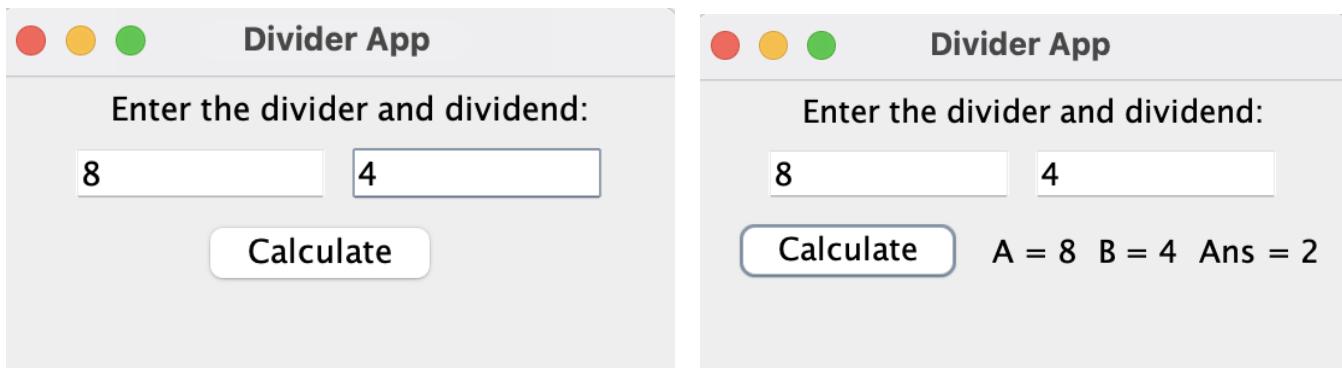
            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
        } catch (NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
 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

// display frame
jfrm.setVisible(true);
}

public static void main(String args[]) {
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {  
    new SwingDemo();  
}  
});  
}  
}
```

Output:



----- Thank You -----

Observation:

Java

- Q. 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.math.*;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void get()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
        if (a == 0)
            System.out.println("Not a quadratic equation");
        else
        {
            d = b*b - 4*a*c;
            if (d < 0)
                System.out.println("No real solutions");
            else
            {
                r1 = (-b + Math.sqrt(d)) / (2*a);
                r2 = (-b - Math.sqrt(d)) / (2*a);
                System.out.println("The roots are " + r1 + " and " + r2);
            }
        }
    }
    void compute()
    {
        while (a == 0)
        {
            System.out.print("Enter a non zero value for a");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }
    }
}

```

$$d = b^2 - 4ac;$$

if ($d == 0$)

{

$$r_1 = (-b) / (2a);$$

System.out.println ("Roots are real and equal");

System.out.println ("Root1 = Root2 = " + r1);

}

else if ($d > 0$)

{

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double})(2a);$$

$$r_2 = ((-b) - (\text{Math.sqrt}(d))) / (\text{double})(2a);$$

System.out.println ("Roots are real and distinct");

System.out.println ("Root1 = " + r1 + " Root2 = " + r2);

}

else if ($d < 0$)

{

System.out.println ("Roots are imaginary");

$$r_1 = (-b) / (2a);$$

$$r_2 = \text{Math.sqrt}(-d) / (2a);$$

System.out.println ("Root1" + r1 + " + i" + r2);

System.out.println ("Root1" + r1 + " - i" + r2);

}

}

3

class QuadraticMain {

{

public static void main (String args [])

{

Quadratic q = new Quadratic();

q.getd();

Q. compute ();

}

3

OUTPUT:

* Enter coefficients -a, b, c.

1 2 1

roots are real and equal

$$\text{root1} = \text{root2} = -1.0,$$

Enter coefficients -a b c

1 -7 12

roots are real & distinct

$$\text{root1} = 4.0 \text{ root2} = 3.0,$$

2. Develop a Java Program to create class Student with members VSN, Name , an array credits and an Array marks. Include method to accept and displays details and the method to calculate CGPA of a student.

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

```
class Student {
```

```
String VSN, Name;
```

```
int credits[], marks[];
```

```
void acceptDetails() {
```

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

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

```
VSN = hello.nextLine();
```

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

```
Name = hello.nextLine();
```

```
credits = new int[4];
```

```
marks = new int[4];
```

```
System.out.println("Enter the details of credits and  
marks in order");
```

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

```
System.out.println("Enter credits for Subject " + (i+1) +  
":");
```

```
credits[i] = s.nextInt();
```

```
System.out.println("Enter marks for subject " + (i+1) +  
":");
```

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

```
}
```

```
void display() {
```

```

System.out.println (" Student details : " );
System.out.println (" USN: " + USN );
System.out.println (" Name: " + Name );
for (int i = 0; i < 5; i++) {
    System.out.println (" Subject: " + (i + 1) + " - Credits: "
        + credits[i] + ", Marks: " + marks[i] );
}

```

```

double SGPA () {
    int total Credits = 0;
    int total Grade Points = 0;

    for (int i = 0; i < 5; i++) {
        total Credits += credits[i];
        total Grade Points += gradePoints(marks[i]) * credits[i];
    }
    double SGPA = total Grade Points / total Credits;
    return SGPA;
}

```

```

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

```

class Main {

```

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

```

Student Stores = new Student();

stores.accept Details();

stores.display();

System.out.println ("SGPA : " + stores.SGPA());

2

3

Student class

Enter uin : 1bm22cs123

8
1/12/24

Enter name : Ashwarya

Enter the details of credits & marks in order

Enter credits for subject 1 : 4

Enter marks for subject 1 : 95

2 : 3

2 : 99

3 : 2

3 : 73

4 : 1

4 : 85

Student details

uin : 1bm22cs123

Name : Ashwarya

Subject 1 : credits : 4, marks 95

2	3	99
---	---	----

3	2	73
---	---	----

4	1	85
---	---	----

SGPA = 9.5

$$\frac{4 \times 10 + 3 \times 10 + 1 \times 9 + 2 \times 8}{10} = 9.5$$

classmate

Date _____

Page _____

- 3 Create a class Book which 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 toString() method that would display the complete details of the Book. Develop a java program to create n book objects.

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

```
class Book {
```

```
String Name, Author;  
double price;  
int numPages;
```

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

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

```
this.author = author;
```

```
this.price = price;
```

```
this.numPages = numPages;
```

```
}
```

```
public void setDetails() {
```

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

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

```
this.name = hello.nextLine();
```

```
System.out.print("Enter author name: ");
```

```
this.author = hello.nextLine();
```

```
System.out.print("Enter price: ");
```

```
this.price = hello.nextDouble();
```

```
System.out.print("Enter no. of pages: ");
```

```
this.numPages = hello.nextInt();
```

```
3
```

```
public void getDetails(){  
    SOP("Book Name:" + name);  
    SOP("Author" + author);  
    SOP("Prices: $" + price);  
    SOP("NoP: " + numPages);  
}
```

```
public class Main {
```

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

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

```
        System.out.print("Enter no. of books: ");
```

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

```
        // Object Book array.
```

BOOK

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

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

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

```
            books[i] = new Book("", "", 0.0, 0);
```

```
            books[i].setDetails();
```

```
}
```

```
        SOP("Details of all books: ");
```

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

```
            SOP("Book" + (i + 1) + ": ");
```

```
            books[i].getDetails();
```

```
}
```

~~Complete~~

```
SOP ("Complete details of all books : ");
for (int i=0; i<n; i++) {
    System.out.println ("Book " + (i+1) + ". " + books[i]);
    costing());
}
```

Enter no. of Books:

Enter details for Book 1

Enter book name: Java

Enter Author: abc

Enter price: 20

Enter pages: 80

Details of all books

Book 1:

Book Name: JAVA

Author: abc

Price: \$20.0

Number of Pages: 80

9/01/24

LAB 4:

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 Triangle, Rectangle & Circle such that each one of the classes extends the class Shape. Each one of the given shape classes contain only the method printArea() that prints 'the area of the given shape.'

~~import java.util.Scanner;~~~~abstract class Shape {~~~~void printArea();~~~~System.out.println("Area of~~~~import java.util.Scanner;~~~~abstract class Shape {~~~~double a,b;~~~~abstract void printArea();~~~~}~~~~class Rectangle extends Shape {~~~~void printArea() {~~~~System.out.println("Area of rectangle is " + (a*b));~~~~}~~~~class Triangle extends Shape {~~~~void printArea() {~~~~System.out.println("Area of Triangle is " + (0.5*a*b));~~~~}~~~~}~~

class Circle extends Shape {

void printArea() {

System.out.println("Area of circle is " + (3.14 * r * r));

}

class Main {

public static void Main (String args[]){

int n;

Rectangle r = new Rectangle();

Triangle t = new Triangle();

Circle c = new Circle();

while(true){

Scanner s1 = new Scanner (System.in);

System.out.println("In MAIN MENU\nSelect shape\n1. Rectangle\n2. Triangle\n3. Circle.\nEnter choice: ");

1. Rectangle\n2. Triangle\n3. Circle.\nEnter choice: ");

FP.

n = s1.nextInt();

switch(n){

case 1:

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

r.a = s1.nextDouble();

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

r.b = s1.nextDouble();

r.printArea();

break;

}

Case 2:

{

System.out.print("Enter length:");

t.a = sc.nextDouble();

System.out.print("Enter breadth:");

t.b = sc.nextDouble();

t.printArea();

break;

}

Case 3:

{

System.out.print("Enter radius:");

c.a = sc.nextDouble();

c.printArea();

break;

}

default: System.out.println("Invalid input");

{

{

{

Output: MAIN MENU

Select Shape

1. Rectangle

2. Triangle

3. Circle

Enter choice:

2

Enter length: 2

Enter breadth: 6

Area of triangle is 6.0

MAIN MENU

1. Rectangle
2. Triangle

3 Circle

Enter choice: 1

Enter length: 2

Enter breadth: 3

Area of rectangle is 6.0.

MAIN MENU

1. Rectangle
2. Triangle
3. Circle

Enter choice: 3

Enter radius: 2

Area of circle is 12.56

8/11/24

6. Create a package CIE which has two classes - Student & Internals. The class Student has members like usn, name, sem. The class Internals has an array that stores one 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 subjects in all five courses.

file!

Aish / CIE / Student.java

Package CIE;

import java.util.Scanner;

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

public void accept()

Scanner s = new Scanner (System.in);

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

this.name = s.nextLine();

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

this.usn = s.nextLine();

s.nextLine ("Enter sem : ");

this.sem = s.nextInt();

{}

```
public void display() {
```

```
    System.out.println("Name: " + this.name + " ID No: " + this.id + " Class: " +
```

```
    this.sem);
```

```
}
```

8

CIE Internal.java

```
package CIE;
```

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

```
public class Internal extends CIE.Student {
```

```
    public int m[] = new int[5];
```

```
    CIE.Student student = new CIE.Student();
```

```
    public void accept()
```

```
        student.accept();
```

~~Scanners s1 = new Scanner(System.in);~~

~~System.out.println("Enter internal marks : ");~~

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

~~m[i] = s1.nextInt();~~

```
    public void display()
```

```
        student.display();
```

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

~~System.out.println("Marks of sub" + (i+1) + " = " + m[i]);~~

```
}
```

```
}
```

```
}
```

file 3

SEE External.java

package SEE;

```
import java.util.Scanner;  
import EEE.Internal;  
import CIE.Student
```

```
public class External extends CIE.Student {
```

```
    public int x[] = new int[5];
```

```
    public void accept() {
```

```
        Scanner s2 = new Scanner(System.in);
```

```
        System.out.println("Enter External marks:");
```

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

```
            x[i] = s2.nextInt();
```

```
}
```

```
    public void display() {
```

```
        super.display();
```

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

```
            System.out.print("Marks " + i + " = " + x[i]);
```

```
}
```

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

```
import CIE.Student;
```

```
import CIE.Internal;
```

```
import SEE.External;
```

```
public class Final {
```

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

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

```
        S.O.P("Enter n:");
```

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

```
        CIE.Internal[] C1 = new CIE.Internal[y];
```

```
        SEE.External[] C2 = new SEE.External[y];
```

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

```
{
```

```
            C1[i] = new CIE.Internal();
```

```
            C2[i] = new SEE.External();
```

```
            C1[i].accept(y); C2[i].accept(y);
```

```
            C1[i].display(); C2[i].display();
```

```
        for (int j = 0; j < y; j++) {
```

```
            double calu = C1[i].m(j) + (C2[i].x(j))/2;
```

```
            System.out.println("Final mark of sub" + (i+1) +
```

```
" is " + calu);
```

```
}
```

```
}
```

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

```
import java.util.*;
```

```
class WrongAgeException extends Exception {
```

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

```
        super(message);
```

```
}
```

```
class Father {
```

```
    private int fatherAge;
```

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

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

```
            throw new WrongAgeException("Age cannot be negative");
```

```
}
```

```
        this.fatherAge = age;
```

class son extends Father {

private int SonAge ;

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

if (sonAge >= fatherAge) {

throw new WrongAgeException ("Son's age should be
less than Father's age");

}

this.SonAge = sonAge;

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

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

}

public class ExceptionInheritanceDemo {

psvm (String [] args) {

Scanner scanner = new Scanner (System.in);

try {

System.out.println ("Enter Father's Age: ");

int fatherAge = scanner.nextInt ();

Father father = new Father (fatherAge);

System.out.println ("Enter Son's Age: ");

Son son = new Son (father, sonAge);

} catch (WrongAgeException e) {

S.O.P ("Exception: " + e.getMessage());

}

} : Enter Father's Age: 40

}

8 9 10 11 12 Enter Son's Age: 41

Exception: Son's age should be less than
Father's age.

8. Write a program which creates two threads, one thread displaying "BMSCE" once every 10 seconds and another displaying "CSE" once every 2 seconds.

Class Thread3 extends Thread

Thread t;

Thread3()

}

t = new Thread(this, "Thread 3. BMSCE");

System.out.println(*t++*);

t.start();

}

public void run()

{

try

for(int i=0; i<10; i++)

{

System.out.println("BMSCE");

Thread.sleep(10000);

}

} catch (InterruptedException ie) { s. o. P("BMSCE 1E"); }

}

Class Thread4 extends Thread

public void run()

try

for(int j=0; j<10; j++)

{

SOPC("CSE");

Thread.sleep(2000);

} catch (InterruptedException ie) { s. o. P("CSE 1E"); }

public class Threads Three?

public static void main(String args[])

{

 Thread3 t1 = new Thread3();

 Thread4 t2 = new Thread4();

 t2.start(); t1.start();

}

3

OUTPUT:

CT: Thread [#22, Thread3@MSE, \$, main]

BMSCE

CSE

BMSEE

BMSCE

KD

BMSCE

CSE

BMSCE

BPM

:

5. Bank

class Bank {

 psvm (String[] args) {

 SavingsAccount savingsAccount = new SavingsAccount

(customerName: "John Doe", accountNumber: "SA1001");

 CurrentAccount currentAccount = new CurrentAccount

(customerName: "Jane Smith", accountNumber: "CA2002");

 savingsAccount.deposit(amount: 5000);

 savingsAccount.displayBalance();

 savingsAccount.computeInterest();

 savingsAccount.withdraw(amount: 2000);

 savingsAccount.displayBalance();

 currentAccount.withdraw(amount: 5000);

 currentAccount.displayBalance();

}

}

class Account {

 protected String customerName, accountNumber;

 protected double balance;

 public Account (String customerName, String accountNumber) {

 this.customerName = customerName;

 this.accountNumber = accountNumber;

 this.balance = 0;

}

public void deposit (double amount) {

balance = amount;

S.O.P ("Deposit of \$" + amount + "Successful");

{

public void displayBalance() {

{ S.O.P ("Account Number:" + accountNumber + "m.Balance" + balance);

}

{

class SavingsAccount extends Account {

public SavingsAccount (String customerName, String accountNumber) {

super (customerName, accountNumber);

}

public void computeInterest () {

double interestRate = 0.05;

double interest = balance * interestRate;

balance += interest;

S.O.P ("Interest:" + interest);

}

public void withdraw (double amount) {

if (balance >= amount) {

balance -= amount;

S.O.P ("withdrawal \$" + amount + "Successful");

}

else

{

S.O.P ("Insufficient funds for withdrawal");

}

{

class CurrentAccount extends Account {

 private double minimumBalance = 1000;

 public CurrentAccount(String customerName, String accountNumber)

{

 super(customerName, accountNumber);

}

 public void withdraw(double amount) {

 if (balance - amount >= minimumBalance) {

 balance -= amount;

 System.out.println("Withdrawal! " + amount);

}

else

{

 System.out.println("Insufficient funds");

 inputServiceCharge();

}

 private void inputServiceCharge() {

 double serviceCharge = 10;

 balance -= serviceCharge;

 System.out.println("Service charge: \$ " + serviceCharge);

}

}

OUTPUT:

Deposit of \$5000.0 successful

Account Number: SA1001

Balance: 5000.0

Interest of \$250.0 .

Account Number: SA1001

Balance: 5250.0

Withdrawal of 2000.0 successful

Account Number: SA1001

Balance: 3250.0

Deposit of \$8000.0 successfully

Account Number: CA20002

Balance: 8000.0

Withdrawal of 5000.0 successful

Account number: CA20002

Balance: 3000.0

By
P/T/2021/01

9.

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

class swingDemo {
    swingDemo() {
        JFrame jfrm = new JFrame ("Division App");
        jfrm.setSize (275,180);
        jfrm.setLayout (new FlowLayout ());
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel ("Enter the dividend and divisor");
        JTextField aJtf = new JTextField (8);
        JTextField bJtf = new JTextField (8);

        JButton button = new JButton ("Calculate");

        JLabel err = new JLabel ();
        JLabel aLab = new JLabel ();
        JLabel bLab = new JLabel ();
        JLabel ansLab = new JLabel ();

        jfrm.add (err);
        jfrm.add (jlab);
        jfrm.add (aJtf);
        jfrm.add (button);
        jfrm.add (aLab);
        jfrm.add (bLab);
        jfrm.add (ansLab);
```

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

```
ajtf.addActionListener(l);
```

```
bjtf.addActionListener(l);
```

```
button.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent evt) {
```

```
        try {
```

```
            int a = Integer.parseInt(ajtf.getText());
```

```
            int b = Integer.parseInt(bjtf.getText());
```

```
            int ans = a/b;
```

```
aLab.setText("In A = " + a);
```

```
bLab.setText("In B = " + b);
```

```
ansLab.setText("In Ans" + ans);
```

```
} catch (NumberFormatException e) {
```

```
aLab.setText(" ");
```

```
bLab.setText(" ");
```

```
ansLab.setText(" ");
```

```
err.setText(" ");
```

```
} catch (ArithmaticException e) {
```

```
aLab.setText(" "));
```

```
bLab.setText(" "));
```

```
ansLab.setText(" "));
```

```
err.setText("NON zero");
```

```
}
```

```
});
```

```
ifrm.setVisible(true);
```

`psvm (String args[]){`

`SwingUtilities.invokeLater(new Runnable(){`

`public void run(){`

`new SwingDemo();`

`}`

`});`

`}`

`}
000 Other App`

`enter direction left side`

`8 4`

`Calculate`

`000 my direct friend`

`8 14`

`total = 8 + 14
Ans = 2`