CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH

Department of Computer Science & Engineering

Subject Name:Java programming

Semester:3rd

Subject Code: CSE201

Academic year: 2024-25

PART-III Object Oriented Programming: Classes, Methods, Constructors

No.	Aim of the Practical

12.	Imagine you are developing a currency conversion tool for a travel agency. This tool should be able to convert an amount in Pounds to Rupees. For simplicity, we assume the conversion rate is fixed: 1 Pound = 100 Rupees. The tool should be able to take input both from command-line arguments and interactively from the user.
	
	PROGRAM CODE:
	
	import java.lang.*;
	
	import java.util.Scanner;
	
	class practice
	
	{
	
	public static void main(String a[])
	
	{
	
	int x=Integer.parseInt(a[0]);
	
	int c=x*100;
	
	System.out.println("Value in rupees is "+c);

```
}

}
```

OUTPUT:

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java pound 3
Value in rupees is 300
```

CONCLUSION

AIM:

Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

PROGRAM CODE:

```
import java.util.Scanner;



class Employee {

    String firstName;

    String lastName;

    double monthlySalary;



   public Employee(String firstName, String lastName, double
monthlySalary) {

       this.firstName = firstName;
```

13.

```
    this.lastName = lastName;
    if (monthlySalary > 0) {
        this.monthlySalary = monthlySalary;
    } else {
        this.monthlySalary = 0.0;
    }
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}
```

```
    public double getMonthlySalary() {
        return monthlySalary;
    }

    public void setMonthlySalary(double monthlySalary) {
        if (monthlySalary > 0) {
            this.monthlySalary = monthlySalary;
        } else {
            this.monthlySalary = 0.0;
        }
    }

    public double getYearlySalary() {
        return monthlySalary * 12;
    }

    public void giveRaise(double percentage) {
        monthlySalary *= (1 + percentage / 100);
    }
}

public class EmployeeTest {
```

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

        System.out.println("Enter first name for employee 1:");
        String firstName1 = scanner.nextLine();
        System.out.println("Enter last name for employee 1:");
        String lastName1 = scanner.nextLine();
        System.out.println("Enter monthly salary for employee 1:");
        double monthlySalary1 = scanner.nextDouble();
        scanner.nextLine(); // Consume newline left-over

        System.out.println("Enter first name for employee 2:");
        String firstName2 = scanner.nextLine();
        System.out.println("Enter last name for employee 2:");
        String lastName2 = scanner.nextLine();
        System.out.println("Enter monthly salary for employee 2:");
        double monthlySalary2 = scanner.nextDouble();
        scanner.nextLine(); // Consume newline left-over

        Employee employee1 = new Employee(firstName1, lastName1,
monthlySalary1);
        Employee employee2 = new Employee(firstName2, lastName2,
monthlySalary2);

        System.out.println("Employee 1 yearly salary: " +
```

```
employee1.getYearlySalary());

        System.out.println("Employee 2 yearly salary: " +
employee2.getYearlySalary());



        employee1.giveRaise(10);

        employee2.giveRaise(10);



        System.out.println("Employee 1 yearly salary after 10% raise: " +
employee1.getYearlySalary());

        System.out.println("Employee 2 yearly salary after 10% raise: " +
employee2.getYearlySalary());

    }

}
```

OUTPUT:

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java EmployeeTest
Enter first name for employee 1:
Saumya
Enter last name for employee 1:
Chandwani
Enter monthly salary for employee 1:
20000
Enter first name for employee 2:
Khushi
Enter last name for employee 2:
Dadhaniya
Enter monthly salary for employee 2:
12000
Employee 1 yearly salary: 240000.0
Employee 2 yearly salary: 144000.0
Employee 1 yearly salary after 10% raise: 264000.0
Employee 2 yearly salary after 10% raise: 158400.00000000003
```

CONCLUSION:

From this practical, we can conclude that the `Employee` class allows us to create and manage employee objects with proper encapsulation and salary handling. The raise demonstrates how the class can adapt to changes in salary.

14.

AIM:

Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

PROGRAM CODE:

```
import java.util.Scanner;


class Date {
    int month;
    int day;
    int year;

   Date(int month, int day, int year) {
      this.month = month;
      this.day = day;
      this.year = year;
   }

    int getMonth() {
      return month;
   }
```

```
    void setMonth(int month) {
        this.month = month;
    }

    int getDay() {
        return day;
    }

    void setDay(int day) {
        this.day = day;
    }

    int getYear() {
        return year;
    }

    void setYear(int year) {
        this.year = year;
    }

    void displayDate() {
        System.out.println(day + "/" + month + "/" + year);
```

```
    }
}


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


        System.out.println("Enter day:");
        int day = scanner.nextInt();
                System.out.println("Enter month:");
        int month = scanner.nextInt();
        System.out.println("Enter year:");
        int year = scanner.nextInt();


        Date date = new Date(month, day, year);


        System.out.println("Date: ");
        date.displayDate();
    }
}
```

OUTPUT:

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java DateTest
Enter day:
15
Enter month:
10
Enter year:
2005
Date:
15/10/2005
```

CONCLUSION:

This practical demonstrates how to create a simple class (`Date`) with instance variables, constructors, setters, getters, and a custom method (`displayDate`).

- It emphasizes the importance of encapsulation and proper data handling.
- By testing the `Date` class, we verify that it correctly initializes and displays dates.
- Overall, this practical reinforces fundamental OOP concepts in Java.

15.

AIM:

Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

PROGRAM CODE:

```
import java.util.Scanner;


class Area {

    double length;

    double breadth;



    Area(double length, double breadth) {
```

```
        this.length = length;
        this.breadth = breadth;
    }

    double returnArea() {
        return length * breadth;
    }
}

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

        System.out.println("Enter length of rectangle:");
        double length = scanner.nextDouble();
        System.out.println("Enter breadth of rectangle:");
        double breadth = scanner.nextDouble();

        Area area = new Area(length, breadth);

        System.out.println("Area of rectangle: " + area.returnArea());
    }
}
```

OUTPUT:

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java AreaTest
Enter length of rectangle:
5
Enter breadth of rectangle:
4
Area of rectangle: 20.0
```

CONCLUSION:

This practical demonstrates how to create a class (`Area`) with a constructor and a method. It shows how to accept input from the keyboard using `Scanner`. By calculating the area of a rectangle, we reinforce the concept of encapsulation and method abstraction in Java.

SUPPLEMENTARY EXPERIMENT:

1.Write a Java program to create a class called "Airplane" with a flight number, destination, and departure time attributes, and methods to check flight status and delay.

PROGRAM CODE:

```
import java.util.Scanner;


class Airplane {

    private String flightNumber;

    private String destination;

    private String departureTime;

    private boolean isDelayed;


    public Airplane(String flightNumber, String destination, String departureTime) {

        this.flightNumber = flightNumber;
```

```
        this.destination = destination;
        this.departureTime = departureTime;
        this.isDelayed = false;
    }

    public String getFlightNumber() {
        return flightNumber;
    }

    public String getDestination() {
        return destination;
    }

    public String getDepartureTime() {
        return departureTime;
    }

    public void checkFlightStatus() {
        if (isDelayed) {
            System.out.println("Flight " + flightNumber + " to " + destination +
" is delayed.");
        } else {
            System.out.println("Flight " + flightNumber + " to " + destination +
" is on time.");
```

```
        }

    }



    public void delayFlight() {

        isDelayed = true;

        System.out.println("Flight " + flightNumber + " to " + destination + "
has been delayed.");

    }



    public void resumeFlight() {

        isDelayed = false;

        System.out.println("Flight " + flightNumber + " to " + destination + "
is back on schedule.");

    }

}



public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);



        System.out.println("Enter flight number:");

        String flightNumber = scanner.nextLine();

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

        String destination = scanner.nextLine();
```

```
        System.out.println("Enter departure time:");
        String departureTime = scanner.nextLine();

        Airplane airplane = new Airplane(flightNumber, destination,
departureTime);

        while (true) {
            System.out.println("Do you want to check flight status? (yes/no)");
            String choice = scanner.nextLine();

            if (choice.equalsIgnoreCase("yes")) {
                airplane.checkFlightStatus();
            } else if (choice.equalsIgnoreCase("no")) {
                System.out.println("Do you want to delay the flight? (yes/no)");
                String delayChoice = scanner.nextLine();

                if (delayChoice.equalsIgnoreCase("yes")) {
                    airplane.delayFlight();
                } else if (delayChoice.equalsIgnoreCase("no")) {
                    System.out.println("Do you want to resume the flight?
(yes/no)");
                    String resumeChoice = scanner.nextLine();

                    if (resumeChoice.equalsIgnoreCase("yes")) {
```

```
                    airplane.resumeFlight();
                } else if (resumeChoice.equalsIgnoreCase("no")) {
                    System.out.println("Exiting...");
                    break;
                } else {
                    System.out.println("Invalid choice. Please try again.");
                }
            } else {
                System.out.println("Invalid choice. Please try again.");
            }
        } else {
            System.out.println("Invalid choice. Please try again.");
        }
    }
}
```

OUTPUT:

16.

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java AirplaneTest
Enter flight number:
23
Enter destination:
Vadodara
Enter departure time:
12
Do you want to check flight status? (yes/no)
yes
Flight 23 to Vadodara is on time.
Do you want to check flight status? (yes/no)
no
Do you want to delay the flight? (yes/no)
yes
Flight 23 to Vadodara has been delayed.
Do you want to check flight status? (yes/no)
no
Do you want to delay the flight? (yes/no)
no
Do you want to resume the flight? (yes/no)
no
Exiting...
```

CONCLUSION:

This practical demonstrates how to create a Java class (`Airplane`) with attributes (flight number, destination, departure time) and methods (to check flight status and simulate delays).

- It reinforces the concept of encapsulation and method abstraction in object-oriented programming.
- By testing the class, we verify that it behaves as expected.

AIM:

Print the sum, difference and product of two complex numbers by creating a class named ‘Complex’ with separate methods for each operation whose real and imaginary parts are entered by user.

PROGRAM CODE:

```
import java.util.Scanner;



public class Complex {

    private double real;
```

```
    private double imaginary;


    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }


    public Complex add(Complex other) {
        double realPart = this.real + other.real;
        double imaginaryPart = this.imaginary + other.imaginary;
        return new Complex(realPart, imaginaryPart);
    }


    public Complex subtract(Complex other) {
        double realPart = this.real - other.real;
        double imaginaryPart = this.imaginary - other.imaginary;
        return new Complex(realPart, imaginaryPart);
    }


    public Complex multiply(Complex other) {
        double realPart = this.real * other.real - this.imaginary *
other.imaginary;
        double imaginaryPart = this.real * other.imaginary + this.imaginary *
other.real;
```

```
    return new Complex(realPart, imaginaryPart);
}

public static Complex readComplexNumber() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the real part: ");
    double real = scanner.nextDouble();
    System.out.print("Enter the imaginary part: ");
    double imaginary = scanner.nextDouble();
    return new Complex(real, imaginary);
}

public void print() {
    System.out.println(real + " + " + imaginary + "i");
}

public static void main(String[] args) {
    Complex num1 = readComplexNumber();
    Complex num2 = readComplexNumber();

    Complex sum = num1.add(num2);
    sum.print();
```

```
        Complex diff = num1.subtract(num2);
        diff.print();

        Complex prod = num1.multiply(num2);
        prod.print();
    }
}
```

OUTPUT:

```
C:\Users\saumy\OneDrive\Documents\JAVA\Practical 3>java Complex
Enter the real part: 14
Enter the imaginary part: 53
Enter the real part: 24
Enter the imaginary part: 67
38.0 + 120.0i
-10.0 + -14.0i
-3215.0 + 2210.0i
```

CONCLUSION:

This practical demonstrates how to create a class (`Complex`) to handle complex numbers.

- It reinforces the concept of encapsulation and method abstraction.
- By testing the class, we verify that it correctly performs addition, subtraction, and multiplication of complex numbers.