Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24

EXPERIMENT NO. 6

AIM / OBJECTIVE:

To implement array of objects and passing/ returning objects

DESCRIPTION OF EXPERIMENT:

- Syntax for creating multiple objects and array of objects.
- Pass by value, pass by reference
- This pointer

- To implement array of objects and passing/ returning objects
- a. WAP to arrange the names of students in descending order of their total marks, input data consists of students details such as names, ID.no, marks of maths, physics, chemistry. (Use array of objects)

Code-

```
import java.util.Scanner;


class Student {

    String name;

    int id;

    int mathMarks;

    int physicsMarks;
```

```
    int chemistryMarks;

    int totalMarks;


    Student(String name, int id, int mathMarks, int physicsMarks, int
chemistryMarks) {

        this.name = name;

        this.id = id;

        this.mathMarks = mathMarks;

        this.physicsMarks = physicsMarks;

        this.chemistryMarks = chemistryMarks;

        this.totalMarks = mathMarks + physicsMarks + chemistryMarks;

    }


    public String toString() {

        return "Name: " + name;

    }

}


public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
```

```
int numStudents = scanner.nextInt();

scanner.nextLine(); // Consume newline


Student[] students = new Student[numStudents];


for (int i = 0; i < numStudents; i++) {
    System.out.println("Enter details for student " + (i + 1) + ":");
    System.out.print("Name: ");
    String name = scanner.nextLine();
    System.out.print("ID: ");
    int id = scanner.nextInt();
    System.out.print("Math Marks: ");
    int mathMarks = scanner.nextInt();
    System.out.print("Physics Marks: ");
    int physicsMarks = scanner.nextInt();
    System.out.print("Chemistry Marks: ");
    int chemistryMarks = scanner.nextInt();
    scanner.nextLine(); // Consume newline


    students[i] = new Student(name, id, mathMarks, physicsMarks,
chemistryMarks);
}
```

```
        scanner.close();


        // Bubble sort
        for (int i = 0; i < numStudents - 1; i++) {
            for (int j = 0; j < numStudents - i - 1; j++) {
                if (students[j].totalMarks < students[j + 1].totalMarks) {
                    // Swap students
                    Student temp = students[j];
                    students[j] = students[j + 1];
                    students[j + 1] = temp;
                }
            }
        }


        System.out.println("Students sorted by total marks in descending order:");
        for (Student student : students) {
            System.out.println(student);
        }
    }
}
```

Output-

```
C:\Users\Arshad\Desktop\study\java>java Main
Enter the number of students: 3
Enter details for student 1:
Name: Arshad
ID: 64
Math Marks: 99
Physics Marks: 98
Chemistry Marks: 99
Enter details for student 2:
Name: joe
ID: 98
Math Marks: 33
Physics Marks: 65
Chemistry Marks: 65
Enter details for student 3:
Name: Tom
ID: 78
Math Marks: 94
Physics Marks: 64
Chemistry Marks: 58
Students sorted by total marks in descending order:
Name: Arshad
Name: Tom
Name: joe
```

- b. WAP to perform mathematical operations on 2 complex numbers by passing and returning object as argument. Show the use of this pointer.

Code-

```
class Complex {

    double real;

    double imaginary;


    Complex(double real, double imaginary) {

        this.real = real;

        this.imaginary = imaginary;
```

```
    }

    // Method to add two complex numbers
    void add(Complex other) {
        this.real += other.real;
        this.imaginary += other.imaginary;
    }

    // Method to subtract two complex numbers
    void subtract(Complex other) {
        this.real -= other.real;
        this.imaginary -= other.imaginary;
    }

    // Method to display the complex number
    void display() {
        System.out.println("Complex Number: " + this.real + " + " + this.imaginary +
"i");
    }
}

public class Main {
```

```
    public static void main(String[] args) {
        Complex num1 = new Complex(2, 3);
        Complex num2 = new Complex(4, 5);

        // Perform addition
        System.out.println("Before addition:");
        num1.display();
        num2.display();
        num1.add(num2);
        System.out.println("After addition:");
        num1.display();

        // Perform subtraction
        System.out.println("\nBefore subtraction:");
        num1.display();
        num2.display();
        num1.subtract(num2);
        System.out.println("After subtraction:");
        num1.display();
    }
}
```

Output-

```
C:\Users\Arshad\Desktop\study\java>java Main
Before addition:
Complex Number: 2.0 + 3.0i
Complex Number: 4.0 + 5.0i
After addition:
Complex Number: 6.0 + 8.0i

Before subtraction:
Complex Number: 6.0 + 8.0i
Complex Number: 4.0 + 5.0i
After subtraction:
Complex Number: 2.0 + 3.0i
```

CONCLUSION:

Complex Number Class: We created a Complex class to represent complex numbers. This class included fields to store the real and imaginary parts of a complex number, as well as methods to perform mathematical operations such as addition and subtraction.

Mathematical Operations: We implemented methods within the Complex class to perform addition and subtraction of complex numbers. These methods modified the real and imaginary parts of a complex number directly.

Input and Output: We demonstrated taking input for complex numbers by creating instances of the Complex class with user-provided values. Additionally, we displayed the results of mathematical operations using the display method.

Use of this Pointer: Initially, we utilized the this pointer explicitly to refer to instance variables within the class. However, in response to your request, we modified the code to eliminate the explicit use of this.

Website References: (Give references to the cited material)

Object Oriented Programming using Java Laboratory (DJS23FLES201)
Academic Year 2023-24