

Assignment:4

Title:

Write a program to demonstrate method overloading and the use of this keyword.

Objective:

1. Demonstrate method overloading, where multiple methods with the same name but different parameter lists are defined.
2. Demonstrate the usage of this keyword, which can be used to refer to the current instance of the class or to differentiate between class attributes and parameters when they have the same name.

Theory:

Method Overloading:

1. Method overloading is a feature in Java where you can define multiple methods with the same name but with different signatures (i.e., different types or numbers of parameters).
2. Java determines which method to call based on the number and type of arguments passed.

this Keyword:

The this keyword is a reference variable that refers to the current instance of the class. It can be used to:

1. Access instance variables when the parameter names are the same as the instance variable names.
2. Call another constructor of the class.
3. Refer to the current object.

Algorithm:

1. Define a class named Calculator.
2. Define overloaded methods to perform addition, with each method accepting different numbers or types of parameters.
3. Use the this keyword to differentiate between instance variables and constructor parameters where necessary.
4. In the main method, create an object of the Calculator class and call the overloaded methods.

Program:

```
// Calculator.java
class Calculator {
    // Instance variables
    int number1;
    int number2;
```

```

// Constructor to initialize the values
public Calculator(int number1, int number2) {
    this.number1 = number1; // Use of 'this' to refer to instance variables
    this.number2 = number2;
}

// Method to add two integers (method overloading)
public int add(int a, int b) {
    return a + b;
}

// Overloaded method to add three integers
public int add(int a, int b, int c) {
    return a + b + c;
}

// Overloaded method to add two doubles
public double add(double a, double b) {
    return a + b;
}

// Method to display the result of addition
public void displayAddition() {
    System.out.println("Addition of " + number1 + " and " + number2 + ": " +
add(number1, number2));
}

public static void main(String[] args) {
    // Creating an object of Calculator class
    Calculator calculator = new Calculator(10, 20);

    // Calling the method with two integer arguments
    System.out.println("Addition of 5 and 10: " + calculator.add(5, 10));

    // Calling the overloaded method with three integer arguments
    System.out.println("Addition of 5, 10, and 15: " + calculator.add(5, 10, 15));

    // Calling the overloaded method with two double arguments
    System.out.println("Addition of 5.5 and 10.5: " + calculator.add(5.5, 10.5));

    // Displaying the result of addition of two instance variables
    calculator.displayAddition();
}
}

```

Explanation:

Class Definition: The class Calculator has instance variables number1 and number2 to store integer values.

Constructor: The constructor accepts two integers (number1 and number2) and initializes the instance variables. The this keyword is used to distinguish between the instance variables (number1 and number2) and the constructor parameters with the same name.

Method Overloading:

1. The add method is overloaded in three ways:
 1. The first add method takes two integers and returns their sum.
 2. The second add method takes three integers and returns their sum.
 3. The third add method takes two doubles and returns their sum.
2. This demonstrates method overloading as the method name add is used multiple times but with different parameters.

Method to Display Addition: The method displayAddition uses the add method to display the sum of the instance variables number1 and number2.

Main Method: The main method:

1. Creates an instance of the Calculator class.
2. Demonstrates method overloading by calling add with different types and numbers of arguments.
3. Displays the sum of the instance variables using the displayAddition method.

Output:

Addition of 5 and 10: 15

Addition of 5, 10, and 15: 30

Addition of 5.5 and 10.5: 16.0

Addition of 10 and 20: 30

Conclusion:

This program demonstrates how method overloading allows you to define multiple methods with the same name but different parameter lists, and how the this keyword is used to refer to the current instance of the class. Method overloading enhances the flexibility of a program, and this helps manage instance variables and method parameters efficiently.