Java Mini Calculator Project – Teaching Plan

This two■phase project reinforces Java fundamentals like data types, constants, Booleans, conditionals, and loops. Then it transitions into modular programming with methods. It’s designed as a progressive teaching plan for beginners.

Version 1: Basic Calculator (No Methods Yet)

Learning Objectives:

- Use variables and constants (final).
- Apply if/else statements and Boolean logic.
- Practice for, while, and do-while loops.
- Control program flow with a Boolean flag.
- Use Scanner for user input.

Example Code (Version 1):

```
import java.util.Scanner;

public class MiniCalculatorV1 {
    public static void main(String[] args) {
        final double PI = 3.14159;
        boolean running = true;
        Scanner sc = new Scanner(System.in);

        while (running) {
            System.out.println("\n=== Mini Calculator ===");
            System.out.println("1. Add");
            System.out.println("2. Subtract");
            System.out.println("3. Area of Circle");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();

            if (choice == 1) {
                System.out.println("Enter two numbers:");
                double a = sc.nextDouble(), b = sc.nextDouble();
                System.out.println("Result: " + (a + b));
            } else if (choice == 2) {
                System.out.println("Enter two numbers:");
                double a = sc.nextDouble(), b = sc.nextDouble();
                System.out.println("Result: " + (a - b));
            } else if (choice == 3) {
                System.out.print("Enter radius: ");
                double r = sc.nextDouble();
                System.out.println("Area = " + (PI * r * r));
            } else if (choice == 4) {
                running = false;
                System.out.println("Goodbye!");
            } else {
                System.out.println("Invalid choice. Try again.");
            }
        }
    }
}
```

Version 2: Calculator with Methods

Learning Objectives:

- Introduce methods with parameters and return values.
- Refactor Version 1 logic into modular, reusable methods.
- Understand method calls and scope.
- Reinforce constants and code reuse.

Example Code (Version 2):

```
import java.util.Scanner;

public class MiniCalculatorV2 {
    static final double PI = 3.14159;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        boolean running = true;

        while (running) {
            int choice = showMenu(sc);
            switch (choice) {
                case 1 -> System.out.println("Result: " + add(sc));
                case 2 -> System.out.println("Result: " + subtract(sc));
                case 3 -> System.out.println("Area = " + areaOfCircle(sc));
                case 4 -> running = false;
                default -> System.out.println("Invalid choice!");
            }
        }
        System.out.println("Program ended.");
    }

    public static int showMenu(Scanner sc) {
        System.out.println("\n=== Mini Calculator ===");
        System.out.println("1. Add");
        System.out.println("2. Subtract");
        System.out.println("3. Area of Circle");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        return sc.nextInt();
    }

    public static double add(Scanner sc) {
        System.out.println("Enter two numbers:");
        return sc.nextDouble() + sc.nextDouble();
    }

    public static double subtract(Scanner sc) {
        System.out.println("Enter two numbers:");
        return sc.nextDouble() - sc.nextDouble();
    }

    public static double areaOfCircle(Scanner sc) {
        System.out.print("Enter radius: ");
        double r = sc.nextDouble();
        return PI * r * r;
    }
}
```

Teaching Flow Summary:

- Lesson 1 – Variables, Constants, and Simple Arithmetic.
- Lesson 2 – Conditionals (if, else, Boolean logic).
- Lesson 3 – Loops (for, while, do-while).
- Lesson 4 – Combine all to create Version 1 calculator.
- Lesson 5 – Introduce methods and refactor into Version 2.
- Lesson 6 – Extend project: add Multiply, Divide, or Power methods.