

# Week 1: Java Fundamentals Review & Setup

---

## Teaching Guide & Hands-on Project

---

### Session Overview

**Duration:** 2 - 3 Hours

**Teaching:** 75 minutes | **Hands-on Practice:** 45 minutes

---

### Learning Objectives

By the end of this session, students will be able to:

1. Set up a complete Java development environment
  2. Understand and use basic Java syntax and data types
  3. Implement control structures (if-else, loops, switch)
  4. Create and call methods with different return types
  5. Build a simple calculator application
- 

### Pre-Class Preparation (Students Action)

**Before class, students should:**

1. Download and install JDK 17 or later from [Oracle](#) or [OpenJDK](#)
  2. Download IntelliJ IDEA Community Edition from [JetBrains](#)
  3. Verify installation by opening terminal/command prompt and typing: `java -version`
- 

### Detailed Lesson Plan (120 Minutes)

Part 1: Environment Setup & Introduction (15 minutes)

#### Welcome & Icebreaker (5 minutes)

- Brief introductions
- Course expectations and structure
- Importance of asking questions in online format

#### Environment Verification (10 minutes)

**Live demonstration:**

1. Check Java Installation

```
# In terminal/command prompt  
java -version  
javac -version
```

## 2. Create First IntelliJ Project

- File → New → Project
- Select "Java" and choose JDK 17+
- Project name: "Week1JavaFundamentals"
- Click "Create"

## 3. Create First Java Class

- Right-click on `src` folder → New → Java Class
- Name it "HelloWorld"

### First Program Together:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Advanced Java!");  
        System.out.println("Today we'll review fundamentals");  
    }  
}
```

**Run the program** - Click the green play button

---

## Part 2: Java Basics Review (25 minutes)

### Variables and Data Types (10 minutes)

#### Teaching Points:

- Java is strongly typed - must declare types
- Primitive types vs Reference types
- Naming conventions (camelCase for variables)

#### Live Coding Example:

```
public class DataTypesDemo {  
    public static void main(String[] args) {  
        // Integer types  
        int age = 25;  
        long population = 7800000000L;  
  
        // Floating-point types  
        double price = 19.99;
```

```

        float rating = 4.5f;

        // Character and Boolean
        char grade = 'A';
        boolean isStudent = true;

        // String (Reference type)
        String name = "John Doe";

        // Print examples
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Price: $" + price);
        System.out.println("Is Student: " + isStudent);
    }
}

```

### Key Points to Emphasize:

- Difference between `int` and `long` (add `L` suffix)
- Difference between `float` and `double` (add `f` suffix for float)
- String uses double quotes, char uses single quotes

### Operators (5 minutes)

#### Quick Demo:

```

public class OperatorsDemo {
    public static void main(String[] args) {
        // Arithmetic operators
        int a = 10, b = 3;
        System.out.println("Addition: " + (a + b));           // 13
        System.out.println("Subtraction: " + (a - b));         // 7
        System.out.println("Multiplication: " + (a * b));       // 30
        System.out.println("Division: " + (a / b));           // 3 (integer
division!)
        System.out.println("Modulus: " + (a % b));            // 1

        // Comparison operators
        System.out.println("a > b: " + (a > b));           // true
        System.out.println("a == b: " + (a == b));          // false

        // Logical operators
        boolean x = true, y = false;
        System.out.println("x && y: " + (x && y));       // false
        System.out.println("x || y: " + (x || y));         // true
    }
}

```

**Important Note:** Highlight integer division vs. floating-point division

## Getting User Input (10 minutes)

### Introduce Scanner class:

```
import java.util.Scanner;

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

        // Get string input
        System.out.print("Enter your name: ");
        String name = scanner.nextLine();

        // Get integer input
        System.out.print("Enter your age: ");
        int age = scanner.nextInt();

        // Get double input
        System.out.print("Enter your height (in meters): ");
        double height = scanner.nextDouble();

        // Display results
        System.out.println("\n--- Your Information ---");
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Height: " + height + "m");

        scanner.close();
    }
}
```

### Common Pitfalls to Mention:

- Remember to import Scanner
- Use `nextLine()` for strings, `nextInt()` for integers, `nextDouble()` for decimals
- Buffer issue with `nextInt()` followed by `nextLine()` (mention briefly, will cover later)

---

## Part 3: Control Structures (20 minutes)

### If-Else Statements (8 minutes)

#### Example 1: Simple If-Else

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

        System.out.print("Enter your exam score (0-100): ");
```

```

        int score = scanner.nextInt();

        if (score >= 90) {
            System.out.println("Grade: A - Excellent!");
        } else if (score >= 80) {
            System.out.println("Grade: B - Good job!");
        } else if (score >= 70) {
            System.out.println("Grade: C - Satisfactory");
        } else if (score >= 60) {
            System.out.println("Grade: D - Needs improvement");
        } else {
            System.out.println("Grade: F - Failed");
        }

        scanner.close();
    }
}

```

## Switch Statement (5 minutes)

**Example:**

```

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

        System.out.print("Enter day number (1-7): ");
        int day = scanner.nextInt();

        switch (day) {
            case 1:
                System.out.println("Monday");
                break;
            case 2:
                System.out.println("Tuesday");
                break;
            case 3:
                System.out.println("Wednesday");
                break;
            case 4:
                System.out.println("Thursday");
                break;
            case 5:
                System.out.println("Friday");
                break;
            case 6:
                System.out.println("Saturday");
                break;
            case 7:
                System.out.println("Sunday");
                break;
        }
    }
}

```

```

        default:
            System.out.println("Invalid day number!");
    }

    scanner.close();
}
}

```

**Important:** Emphasize the `break` statement!

## Loops (7 minutes)

### For Loop Example:

```

public class LoopsDemo {
    public static void main(String[] args) {
        // For loop - count to 10
        System.out.println("Counting with for loop:");
        for (int i = 1; i <= 10; i++) {
            System.out.print(i + " ");
        }
        System.out.println();

        // While loop - sum of numbers
        System.out.println("\nSum of first 5 numbers:");
        int sum = 0;
        int n = 1;
        while (n <= 5) {
            sum += n;
            n++;
        }
        System.out.println("Sum: " + sum);

        // Do-while loop - menu example
        Scanner scanner = new Scanner(System.in);
        int choice;
        do {
            System.out.println("\n--- Menu ---");
            System.out.println("1. Option 1");
            System.out.println("2. Option 2");
            System.out.println("0. Exit");
            System.out.print("Enter choice: ");
            choice = scanner.nextInt();

            if (choice != 0) {
                System.out.println("You selected: " + choice);
            }
        } while (choice != 0);

        System.out.println("Goodbye!");
        scanner.close();
    }
}

```

```
}
```

---

## Part 4: Methods (15 minutes)

### Teaching Points:

- Methods help organize code and promote reusability
- Structure: returnType methodName(parameters) {}
- Methods must be called to execute

### Example with Multiple Methods:

```
public class MethodsDemo {  
  
    // Method with no parameters, no return  
    public static void greet() {  
        System.out.println("Hello! Welcome to Java programming.");  
    }  
  
    // Method with parameters, no return  
    public static void greetPerson(String name) {  
        System.out.println("Hello, " + name + "!");  
    }  
  
    // Method with parameters and return value  
    public static int add(int a, int b) {  
        return a + b;  
    }  
  
    // Method with return value  
    public static double calculateArea(double radius) {  
        return Math.PI * radius * radius;  
    }  
  
    // Method that calls another method  
    public static void displayCalculation(int x, int y) {  
        int result = add(x, y);  
        System.out.println(x + " + " + y + " = " + result);  
    }  
  
    public static void main(String[] args) {  
        // Calling methods  
        greet();  
        greetPerson("Alice");  
  
        int sum = add(5, 3);  
        System.out.println("Sum: " + sum);  
  
        double area = calculateArea(5.0);  
    }  
}
```

```
        System.out.println("Circle area: " + area);

        displayCalculation(10, 20);
    }
}
```

## Key Concepts:

- `void` means no return value
  - `return` statement exits method and sends value back
  - Method must be called to execute
  - Parameters are inputs to methods
- 

## 💻 HANDS-ON PROJECT: Simple Calculator (45 minutes)

### Project Requirements

Create a console-based calculator that:

1. Displays a menu with operations (Add, Subtract, Multiply, Divide, Exit)
  2. Takes two numbers as input from user
  3. Performs the selected operation
  4. Displays the result
  5. Continues until user chooses to exit
  6. Handles division by zero error
- 

### Step-by-Step Guided Implementation

**Give students 5 minutes to think about the problem, then code together:**

```
import java.util.Scanner;

public class SimpleCalculator {

    // Method to display menu
    public static void displayMenu() {
        System.out.println("\n===== CALCULATOR =====");
        System.out.println("1. Addition (+)");
        System.out.println("2. Subtraction (-)");
        System.out.println("3. Multiplication (*)");
        System.out.println("4. Division (/)");
        System.out.println("5. Exit");
        System.out.println("=====");
        System.out.print("Enter your choice (1-5): ");
    }

    // Method for addition
    public static double add(double num1, double num2) {
```

```

        return num1 + num2;
    }

    // Method for subtraction
    public static double subtract(double num1, double num2) {
        return num1 - num2;
    }

    // Method for multiplication
    public static double multiply(double num1, double num2) {
        return num1 * num2;
    }

    // Method for division
    public static double divide(double num1, double num2) {
        if (num2 == 0) {
            System.out.println("Error: Cannot divide by zero!");
            return 0;
        }
        return num1 / num2;
    }

    // Method to get numbers from user
    public static double[] getNumbers(Scanner scanner) {
        double[] numbers = new double[2];

        System.out.print("Enter first number: ");
        numbers[0] = scanner.nextDouble();

        System.out.print("Enter second number: ");
        numbers[1] = scanner.nextDouble();

        return numbers;
    }

    // Method to perform calculation based on choice
    public static void performCalculation(int choice, double num1, double num2) {
        double result = 0;
        String operation = "";

        switch (choice) {
            case 1:
                result = add(num1, num2);
                operation = "+";
                break;
            case 2:
                result = subtract(num1, num2);
                operation = "-";
                break;
            case 3:
                result = multiply(num1, num2);
                operation = "*";
                break;
            case 4:

```

```

        result = divide(num1, num2);
        operation = "/";
        break;
    default:
        System.out.println("Invalid choice!");
        return;
    }

    System.out.println("\nResult: " + num1 + " " + operation + " " + num2 + " "
= " + result);
}

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

    System.out.println("Welcome to Simple Calculator!");

    do {
        displayMenu();
        choice = scanner.nextInt();

        if (choice >= 1 && choice <= 4) {
            double[] numbers = getNumbers(scanner);
            performCalculation(choice, numbers[0], numbers[1]);
        } else if (choice == 5) {
            System.out.println("\nThank you for using the calculator!");
            System.out.println("Goodbye!");
        } else {
            System.out.println("\nInvalid choice! Please enter 1-5.");
        }
    } while (choice != 5);

    scanner.close();
}
}

```

---

## Sample Output

```

Welcome to Simple Calculator!

===== CALCULATOR =====
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
=====
Enter your choice (1-5): 1

```

```

Enter first number: 10
Enter second number: 5

Result: 10.0 + 5.0 = 15.0

===== CALCULATOR =====
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
=====
Enter your choice (1-5): 4
Enter first number: 20
Enter second number: 0
Error: Cannot divide by zero!

Result: 20.0 / 0.0 = 0.0

===== CALCULATOR =====
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
=====
Enter your choice (1-5): 5

```

Thank you for using the calculator!  
Goodbye!

## Homework Assignment

### Task 1: Enhance the Calculator (Required)

Add the following features to the calculator:

1. **Modulus operation** (remainder after division)
2. **Power operation** (number raised to a power) - use `Math.pow()`
3. **Square root operation** - use `Math.sqrt()`

### Task 2: Temperature Converter (Optional Challenge)

Create a program that converts temperatures between Celsius and Fahrenheit:

- Menu: 1) Celsius to Fahrenheit, 2) Fahrenheit to Celsius, 3) Exit
- Formulas:
  - $F = (C \times 9/5) + 32$
  - $C = (F - 32) \times 5/9$
- Use methods for each conversion

- Include input validation (temperatures cannot go below absolute zero: -273.15°C or -459.67°F)
- 

## Assessment Checklist

Students should be able to:

- Set up Java development environment
  - Create and run a Java program
  - Use Scanner for user input
  - Declare variables with appropriate data types
  - Use if-else and switch statements
  - Write for, while, and do-while loops
  - Create methods with parameters and return values
  - Handle basic error cases (like division by zero)
- 

## Additional Resources for Students

- [Oracle Java Tutorials](#)
  - [W3Schools Java Tutorial](#)
  - Practice platform: [CodingBat Java](#)
- 

## Preview of Week 2

Next week we'll learn:

- Classes and Objects
- Constructors
- Encapsulation
- Building a Student Management System

**Prepare:** Review today's concepts and complete homework assignments!