Chapter 02

Programming Principles

Data Types, Variables & Control Flow

Mr. SOK Pongsametrey
metreysk@gmail.com

This session focuses on Java data types, variables, input/output, and flow control — the building blocks of logical thinking in programming.

ABOUT

What are Data Types?

- Define the kind of data a variable can hold
- Java is statically typed
- Two categories:
  - Primitive Types: int, double, char, boolean, etc.
  - Reference Types: String, arrays, objects

Java Primitive Data Types

Type	Example	Description
byte	100	8-bit signed integer
short	10000	16-bit signed integer
int	42	32-bit signed integer
long	12345678900L	64-bit signed integer
float	3.14f	32-bit floating-point
double	3.14159	64-bit floating-point
boolean	true/false	Logical values
char	'A'	Single Unicode character
String	“This is a string”	

Variables and Constants

- Declaring a variable: int age = 25;
- Updating value: age = 30;
- Final keyword: final double PI = 3.14159;
- Naming rules: case-sensitive, camelCase style

User Input using Scanner

- Need to import Scanner
- Useful for interactive console apps

```
import java.util.Scanner;
Scanner scanner = new Scanner(System.in);
System.out.print("Enter name: ");
String name = scanner.nextLine();
```

```
public static void getUserInput() {
    Scanner scanner = new Scanner(System.in);

    System.out.print(s:"Enter first number: ");
    int num1 = scanner.nextInt();

    System.out.print(s:"Enter second number: ");
    int num2 = scanner.nextInt();

    System.out.print(s:"Enter third number: ");
    int num3 = scanner.nextInt();

    // Using the existing App class methods to find maximum
    int result1 = App.findMax(num1, num2, num3);
    int result2 = App.findMaxV2(num1, num2, num3);

    System.out.println("Maximum (method 1) of " + num1 + ", " + num2 + ", "
        + num3 + " is: " + result1);
    System.out.println("Maximum (method 2) of " + num1 + ", " + num2 + ", "
        + num3 + " is: " + result2);

    scanner.close();
}
```

```
E:\Lecturing\JAVA\javabasic-v2-1\target\c
Welcome to Week 2 Example!
Enter first number: 12
Enter second number: 43
Enter third number: 45
Maximum (method 1) of 12, 43, 45 is: 45
Maximum (method 2) of 12, 43, 45 is: 45
PS E:\Lecturing\JAVA\javabasic-v2-1>
```

Control Flow – Conditional Statements

- if, else if, else
- switch statement
- Comparison operators: ==, !=, >, <, >=, <=

```java
public static void demonstrateConditionals(Scanner scanner) {
    System.out.println("\n=== CONDITIONAL STATEMENTS DEMO ===");

    // if, else if, else example
    System.out.print("Enter a number (1-100): ");
    int num = scanner.nextInt();

    if (num >= 80) {
        System.out.println("High score!");
    } else if (num >= 50) {
        System.out.println("Medium score!");
    } else {
        System.out.println("Low score!");
    }

    // Comparison operators
    System.out.print("Enter two numbers: ");
    int a = scanner.nextInt();
    int b = scanner.nextInt();

    System.out.println(a + " == " + b + ": " + (a == b));
    System.out.println(a + " != " + b + ": " + (a != b));
    System.out.println(a + " > " + b + ": " + (a > b));
    System.out.println(a + " < " + b + ": " + (a < b));
    System.out.println(a + " >= " + b + ": " + (a >= b));
    System.out.println(a + " <= " + b + ": " + (a <= b));

    // Switch statement
    System.out.print("Enter day (1-3): ");
    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;
        default:
            System.out.println("Invalid day");
    }
}
```

Control Flow – Loops

- while loop
- do-while loop
- for loop
- Loop control: break, continue

```java
public static void demonstrateLoops() {
    // While loop
    int i = 1;
    while (i <= 3) { System.out.println("While: " + i); i++; }

    // Do-while loop
    int j = 1;
    do { System.out.println("Do-while: " + j); j++; } while (j <= 3);

    // For loop with break and continue
    for (int k = 1; k <= 5; k++) {
        if (k == 3) continue; // Skip 3
        if (k == 5) break;    // Stop at 5
        System.out.println("For: " + k);
    }
}
```

Mini Example – Even or Odd Checker

```
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number: ");
int num = scanner.nextInt();
if (num % 2 == 0) {
    System.out.println("Even");
} else {
    System.out.println("Odd");
}
```

Lab Activity

- Write a grading system:
  - Input: exam score
  - Output: grade A, B, C, D, or F
- Loop challenge: print numbers from 1–50, but skip multiples of 3

Exercise

- Task: Implement a Java program that:
  - Accepts an integer input
  - Checks if it’s prime, even, or divisible by 5
- Due: Before next class

Expectation:

- Technically explain how you do, even using AI
- How to run it and what is the result

Wrap-up and Q&A

- Key Takeaways:
  - Java is statically typed
  - Control flow enables logic-based decisions
  - Scanner allows interactive inputs
- Next Sesson: Functions and Modular Thinking