**week 1 summary:**

**Topics and Key Learnings**

**Lecture 1-1: Introduction to Computer Science**

1. **Course Objectives:**

• Gain basic exposure to computer science and programming.

• Develop analytical skills and appreciation for clarity in code.

• Build foundational skills for learning other programming languages.

2. **Introduction to Programming:**

• Key principles: Correctness, efficiency, readability, maintainability.

• The importance of writing code that is easy to understand and extend.

3. **Why Java?**

• Java’s versatility, multi-platform capabilities, and robust development tools.

• Example of a simple Java program to print numbers from 0 to 5 using a while loop.

4. **Java Program Structure:**

• Basic elements of Java syntax:

• Classes, methods, variables, loops (while), and comments.

• Syntax, semantics, and style in programming.

• Importance of program readability and clarity.

5. **Program Development Process:**

• Steps: Editing, compiling, running, debugging.

• Debugging tools and common error types:

• Compile-time errors (syntax issues), runtime errors (program crashes), logical errors (unexpected behavior).

**Lecture 1-2: Data Types and Variables**

1. **Data and Variables:**

• Variables as named containers with a type and a value.

• Examples of variable declaration and assignment for int, String, and double.

2. **Primitive and Object Data Types:**

• **Primitive Types:** Integer, real numbers, boolean.

• **Object Types:** Strings and more complex aggregates.

3. **Arithmetic Operations and Expressions:**

• Integer and real number operations: addition, subtraction, multiplication, division, modulo.

• Precedence rules and use of parentheses for clarity.

4. **Strings:**

• String concatenation with other strings or variables of other types.

• Example of string manipulation using + operator.

5. **Command-Line Arguments:**

• Using args[] to pass input to Java programs.

• Example: Arithmetic operations on two numbers passed as arguments.

6. **Real Numbers and Precision:**

• Limitations in representing real numbers accurately due to finite bits.

• Potential for rounding errors and exotic bugs in calculations.

7. **Boolean Data Type:**

• Logical expressions with true and false.

• Example: Leap year calculation using logical operators.

8. **Type Casting:**

• Implicit and explicit type conversions in Java.

• Examples of casting integers, doubles, and strings.

9. **Random Numbers:**

• Using Math.random() to generate pseudo-random values.

**Themes Across Lectures 1-1 and 1-2:**

• **Foundation of Programming:**

• Introduction to Java syntax, program structure, and development practices.

• **Core Data Types:**

• Exploration of primitive types, object types, and type conversion.

• **Problem-Solving:**

• Using logical constructs, arithmetic, and string manipulation to solve problems.

• **Debugging and Error Handling:**

• Importance of debugging in program development.

• **Practical Applications:**

• Examples such as solving quadratic equations, checking leap years, and generating random integers.