

Introduction to Java Programming

Generated by LectureMate

February 22, 2025

1 Lecture Overview

This lecture covers essential aspects of Java programming, including lab and tutorial structure, the importance of foundational coding skills, and an introduction to basic programming concepts such as variables, data types, control flow, and loops.

2 Lab and Tutorial Structure

2.1 Labs

Students have been allocated specific labs in their timetable, but they can attend any lab session. Labs are not assessed but are crucial for developing Java skills.

2.2 Tutorials

Tutorials are compulsory and will start in week three. Students will be assigned specific tutorials and must attend them.

3 Communication and Support

Students are encouraged to use Piazza for questions and feedback. The teaching team is new, and responses may take time, but they aim to foster student dialogue.

4 Course Administration

There have been errors in the course timetable regarding assessment dates. Students are encouraged to report any discrepancies to the teaching team.

5 Introduction to Java

5.1 Importance of Learning Java

Understanding Java is crucial for developing programming skills, especially with the rise of AI tools. While AI can assist in coding, foundational knowledge is necessary for effective use of these tools.

5.2 Basic Java Syntax

Java requires explicit declaration of variable types. For example:

```
int a = 3;  
int b = 17;  
int c = a + b;
```

This code snippet demonstrates variable declaration and initialization.

6 Data Types

Java has several built-in data types:

- **int**: Integer values
- **double**: Floating-point numbers
- **char**: Single characters (enclosed in single quotes)
- **String**: Sequences of characters (enclosed in double quotes)
- **boolean**: True or false values

7 Operators

7.1 Arithmetic Operations

Java supports standard arithmetic operations:

- Addition: +
- Subtraction: -
- Multiplication: *
- Division: /
- Modulus: %

Integer division truncates the decimal part, while double division retains it.

7.2 Casting

Casting allows conversion between data types. For example, to convert an integer to a double:

```
double result = (double) a / b;
```

8 Control Flow

8.1 If Statements

Conditional statements allow execution of code based on boolean expressions:

```
if (x < y) {  
    System.out.println("y is bigger than x");  
} else {  
    System.out.println("x is bigger than y");  
}
```

8.2 Loops

Loops enable repeated execution of code blocks:

8.2.1 While Loops

A while loop continues until a specified condition is false:

```
while (i < n) {  
    // code to execute  
    i++;  
}
```

8.2.2 For Loops

A for loop is a compact way to iterate over a range:

```
for (int i = 0; i < n; i++) {  
    // code to execute  
}
```

9 Conclusion

The lecture emphasized the importance of foundational programming skills in Java, the structure of labs and tutorials, and introduced basic programming concepts. Students are encouraged to practice coding and attend lab sessions for support.