
41039 Programming 1

Complete Course Notes

FREE PREVIEW — Chapter 1 of 13

This preview includes

- Chapter 1: Java Fundamentals — in full

The complete set also covers

- Program Structure & I/O
- Control Flow
- Methods
- Records
- Classes & OOP
- Interfaces
- Arrays & Collections
- Generics
- Errors & Exceptions
- File Handling
- Python Data Structures
- Key Coding Concepts

Side-by-side Java & Python coverage, with syntax-highlighted code examples throughout.

1 Java Fundamentals

1.1 Program Structure Basics

Every line of code that runs in Java must be inside a **class**.

- A class name should always start with an uppercase letter.
- The name of the Java file must match the class name.
- The `main()` method is required in every Java program — it's where execution starts.

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

`println()` inserts a new line after printing; `print()` does not.

1.2 Variables & Data Types

```
int myNum = 5;  
float myFloatNum = 5.99f;  
char myLetter = 'D';  
boolean myBool = true;  
String myText = "Hello";
```

Variables follow the format `type variableName = value;`

- Float literals must end with `f`. Alternatively, use `double` for greater precision, ending the literal with `d`.
- **Primitive** types start with a lowercase letter (e.g. `int`) and always hold a value.
- **Non-primitive** types start with an uppercase letter (e.g. `String`) and can be `null`.

```
final int myNum = 15; // constant - cannot be reassigned
```

The keyword `final` declares a variable as a constant — any attempt to change it raises an error.

```
var x = 5; // compiler infers x is an int  
System.out.println(x);
```

`var` lets the compiler auto-detect a variable's type from its assigned value — but only works when the value is assigned at the same time.

```
int x = 5, y = 6;  
System.out.println("The sum is " + x + y); // "The sum is 56" (string concatenation!)  
System.out.println("The sum is " + (x + y)); // "The sum is 11" (parentheses force  
    addition)
```

1.3 Type Casting

Widening casting converts a smaller type into a larger one — happens automatically:

byte → short → char → int → long → float → double

- byte: 8 bits, signed (range −128 to 127)
- short: 16 bits, signed
- char: 16 bits, unsigned (range 0 to 65,535)
- int: 32 bits
- long: 64 bits
- float: 32 bits, signed — stores 6-7 decimal places; ranked above long despite fewer bits since it's floating-point
- double: 64 bits, signed — stores 15-16 decimal places

```
int myInt = 9;
double myDouble = myInt; // automatic widening: int to double
System.out.println(myDouble); // 9.0
```

Narrowing casting converts a larger type into a smaller one — must be done manually:

```
double myDouble = 9.78d;
int myInt = (int) myDouble; // manual narrowing: double to int
System.out.println(myInt); // 9
```

1.4 Operators

```
int x = 10, y = 3;
System.out.println(x + y); // 13
System.out.println(x - y); // 7
System.out.println(x * y); // 30
System.out.println(x / y); // 3
System.out.println(x % y); // 1
```

```
int z = 5;
++z; System.out.println(z); // 6
--z; System.out.println(z); // 5
```

- **Comparison** operators (==, !=, <, >, <=, >=) return true or false.
- **Logical** operators also return true/false: && (AND), || (OR), ! (NOT).

1.5 Strings

```
String txt1 = "Hello World";
System.out.println(txt1.length()); // 11
System.out.println(txt1.toUpperCase()); // "HELLO WORLD"
System.out.println(txt1.toLowerCase()); // "hello world"
System.out.println(txt1.charAt(0)); // H
System.out.println(txt1.charAt(4)); // o

String txt2 = "Please locate where 'locate' occurs!";
System.out.println(txt2.indexOf("locate")); // 7
System.out.println(txt1.equals(txt2)); // false

String txt3 = " Hello World ";
```

```
System.out.println(txt3.trim()); // "Hello World"
```

Java counts positions from zero.

To add a double quote inside a string, use a backslash. This also works for single quotes and backslashes themselves:

```
String txt = "We are the so-called \"Vikings\" from the north.";
```

1.6 Math Methods

```
System.out.println(Math.max(5, 10)); // 10
System.out.println(Math.min(5, 10)); // 5
System.out.println(Math.sqrt(64)); // 8.0
System.out.println(Math.abs(-4.7)); // 4.7
System.out.println(Math.pow(2, 8)); // 256.0
System.out.println(Math.round(4.6)); // 5
System.out.println(Math.ceil(4.1)); // 5.0
System.out.println(Math.floor(4.9)); // 4.0
System.out.println(Math.random() * 100); // random double in [0.0, 100.0)
System.out.println((int)(Math.random() * 101)); // random int in [0, 100]
```

PREVIEW

Enjoying these notes?

This is just Chapter 1 of 13.

The full set includes:

- Program Structure & I/O
- Control Flow
- Methods
- Records
- Classes & OOP
- Interfaces
- Generics
- Errors & Exceptions
- File Handling
- Python Data Structures
- Key Coding Concepts
- Arrays & Collections

Same format — full theory, key formula boxes, and worked examples for every topic, with Java and Python covered side by side.

Thanks for checking out the preview!