

# Java Basics: What is Java?

---

**CPSC 219: Introduction to Computer Science for Multidisciplinary  
Studies II  
Fall 2023**

Jonathan Hudson, Ph.D.  
Instructor  
Department of Computer Science  
University of Calgary

Wednesday, 06 September 2023

Copyright © 2023



UNIVERSITY OF  
CALGARY

# Java

---

- **Java 20 is the official programming language for this course.**
- **We will teach syntax for Java 6/8/11 (12+ just adds more on top)**
  - Java used to be version 1.6, 1.7, 1.11
  - 1.20 -> Java 20 is the latest OpenJDK version
  - These are minor changes like going from Python 3.9 to Python 3.10 (old syntax is maintained)
- You are expected to use IDE IntelliJ
  - Others like Eclipse/Netbeans/VSCode, etc. are possible, but you may receive no help from staff
- We encourage some degree of knowledge about direct interaction with the computer systems (command line)

# Java Programming Language

---

# Python vs Java

---

## Python

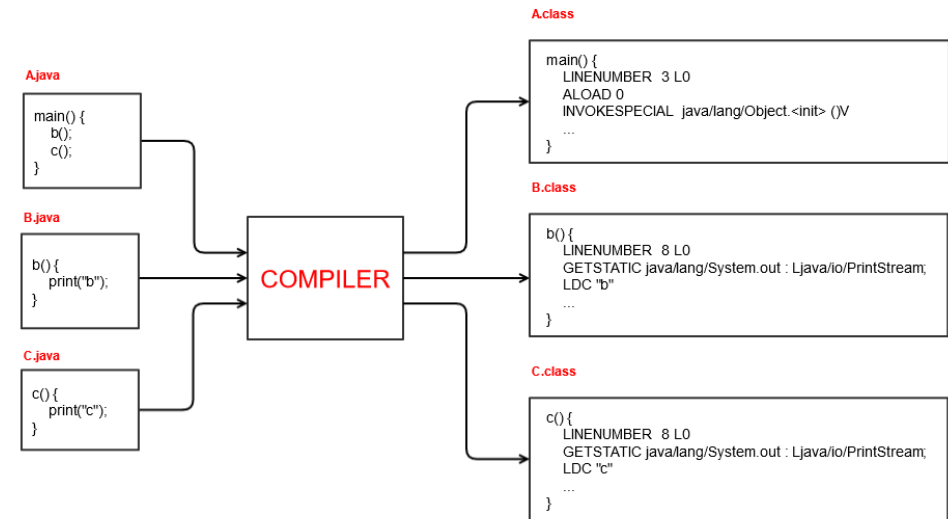
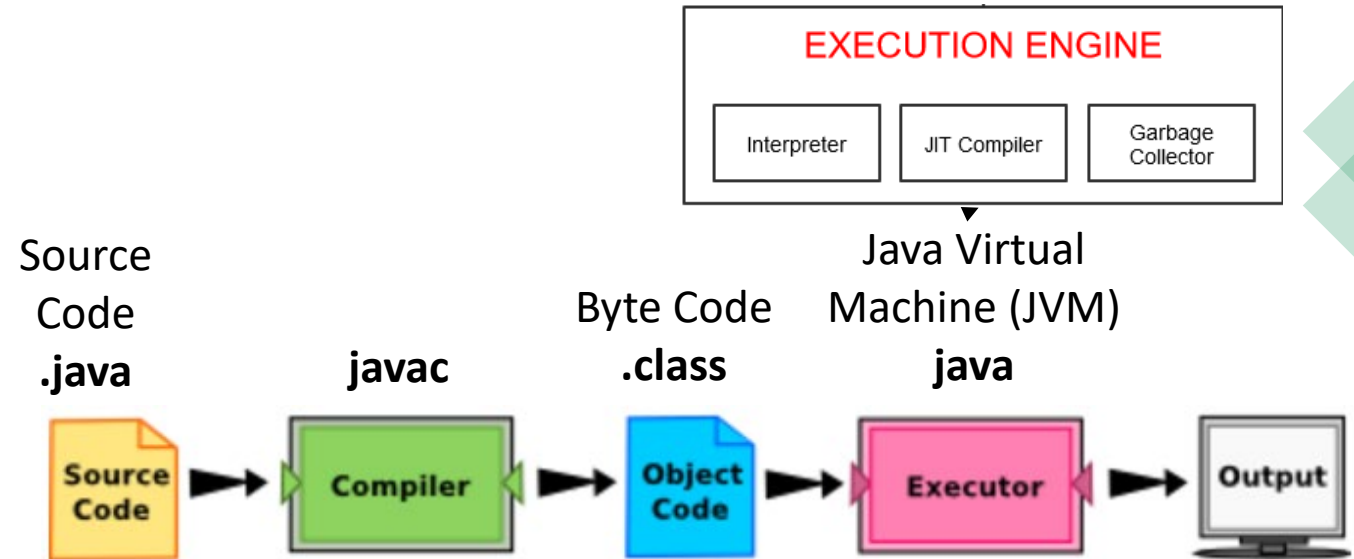
- Flexible and implicit syntax
- Ability to quickly create small programs and applications
  - Implicit syntax makes scaling harder
- Syntax is sparse and clear
- Can be interpreted or compiled to bytecode

## Java

- Formal and explicit syntax
- Designed for any project (no matter the size)
  - Explicit syntax makes scaling clearer to manage (still takes time)
- Syntax more 'computer-like'
- Compiled (code always converted to bytecode before running)

# Compiler (Java)

- A **compiler**:
  - Is like **translating an entire book** and give it to a reader.
  - A compiler reads the program and translates it completely before the program starts running
- For Java the **byte code is stored in .class files**.
  - Unlike in Python where you generally shared your .py files
  - In Java we often just shared these .class files with people who want to run our code
  - (not as easy for a human to read these files)



# Compiled or Interpreted (Both)

---

- Many modern languages use both processes.
- Java uses both processes.
- Java is a **compiled interpreted language**
- Java is first compiled into a lower-level language called byte code and then interpreted by a virtual machine program.
  - (**byte code is higher level than machine code**, we can still move it between machine types, Win/MacOS. Machine code can only move within the same type.)
  - Often, we zip up .class files into a compressed .zip file we rename a **.jar file**
  - A **virtual machine** is created for each operating system type: Windows, Mac, Linux, etc.
- <https://www.baeldung.com/java-compiled-interpreted> (more details if you are interested)

# Just In Time - Compiler

---

- Most modern Java also uses a JIT (Just In Time – Compiler) this recognizes when certain byte code is often re-interpreted over and over (like a function) and converts it into stored machine native code (rather than re-interpreting)
- This is a runtime optimization (makes for interesting runtime speed testing as your program can speed up the longer it runs!)
- One performance result (not universal!)
  - **Java using JIT compiler – 2726 ns – fastest**
  - C++ with O2 optimization – 3639 ns – 33% slower
  - C++ without O2 optimization – 9435 ns – 246% slower
  - **Java without JIT compiler – 17965 ns – 559% slower**
  - *JavaScript (web/browser language) – 22998 ns – 743% slower*

# Running Java Program

---



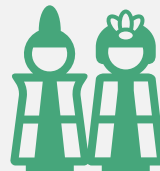
# Command lines and files



You can check your LAPTOP version using  
**java -version**  
**javac -version**



If the result is not 20.X.X+ then  
*Lab environment is 20.0.2*



Then you need to install at least Java 20  
(OpenJDK 20)

# Running a simple Java file from command line

---

**Source code** is a file containing your code often referred to as a *program*. We use words with upper case first letters for Java source code files.

- The filename ends with a *.java* suffix

e.g. **Main.java**

- To execute from terminal/shell (make Main.class via compiling Main.java, then run it):

```
javac Main.java
```

```
java Main
```

- To pipe the output into a file output.txt:

```
java Main > output.txt
```

# Running a simple Java file from command line

---

- Or like Python you can now do this directly:

```
java Main.java
```

# Onward to ... variables.

---

Jonathan Hudson  
[jwhudson@ucalgary.ca](mailto:jwhudson@ucalgary.ca)  
<https://pages.cpsc.ucalgary.ca/~jwhudson/>

