School of Computing & Communications | Lancaster University

SCC.111 Software Development – Lecture 28: Introduction to Java

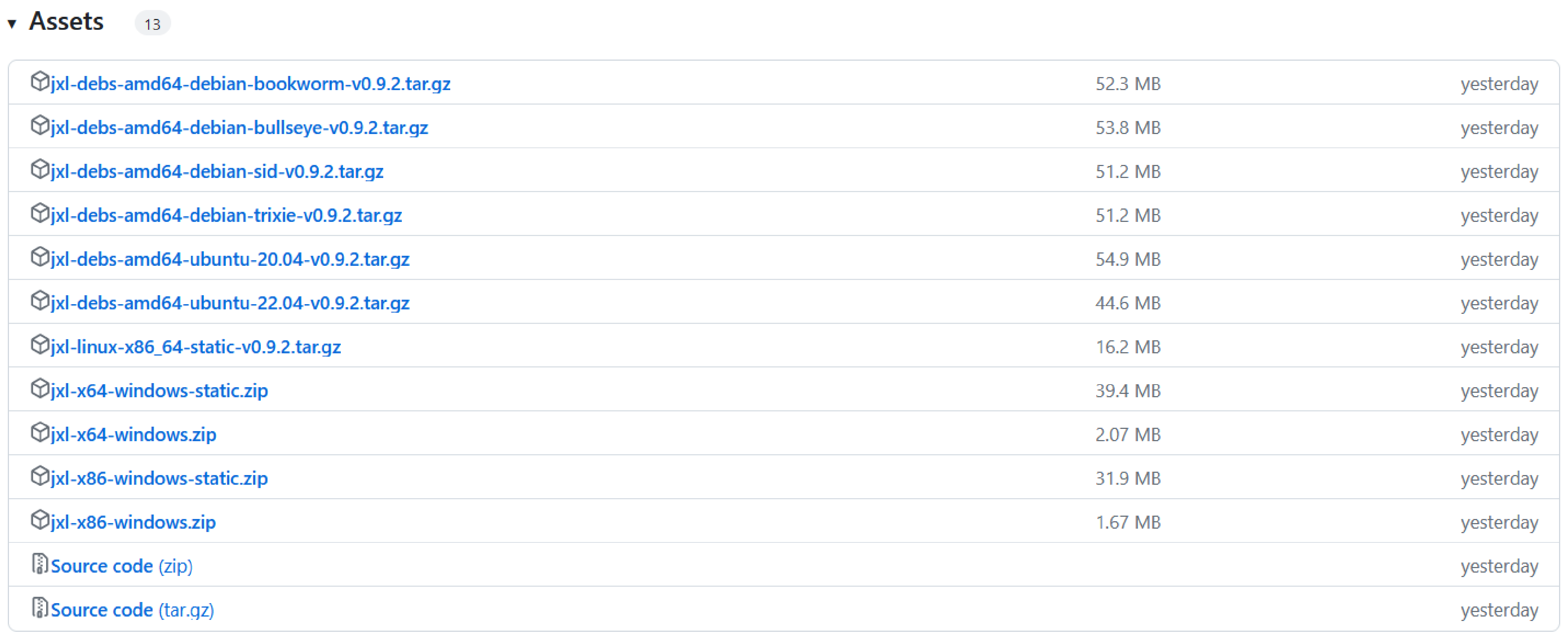
Adrian Friday, Hansi Hettiarachchi and Nigel Davies

Introduction

- Last lecture, we looked at:
  - C/C++ Pros and Cons
  - Too much power to developers
  - Memory Handling and Security Issues

- Today we're going to
  - Explore the alternatives: Virtual Machine based OO languages
  - Introduction to Java

C++ based project releases



▾ Assets 13

jxl-debs-amd64-debian-bookworm-v0.9.2.tar.gz	52.3 MB	yesterday
jxl-debs-amd64-debian-bullseye-v0.9.2.tar.gz	53.8 MB	yesterday
jxl-debs-amd64-debian-sid-v0.9.2.tar.gz	51.2 MB	yesterday
jxl-debs-amd64-debian-trixie-v0.9.2.tar.gz	51.2 MB	yesterday
jxl-debs-amd64-ubuntu-20.04-v0.9.2.tar.gz	54.9 MB	yesterday
jxl-debs-amd64-ubuntu-22.04-v0.9.2.tar.gz	44.6 MB	yesterday
jxl-linux-x86_64-static-v0.9.2.tar.gz	16.2 MB	yesterday
jxl-x64-windows-static.zip	39.4 MB	yesterday
jxl-x64-windows.zip	2.07 MB	yesterday
jxl-x86-windows-static.zip	31.9 MB	yesterday
jxl-x86-windows.zip	1.67 MB	yesterday
Source code (zip)		yesterday
Source code (tar.gz)		yesterday

Virtual Machine Based Languages

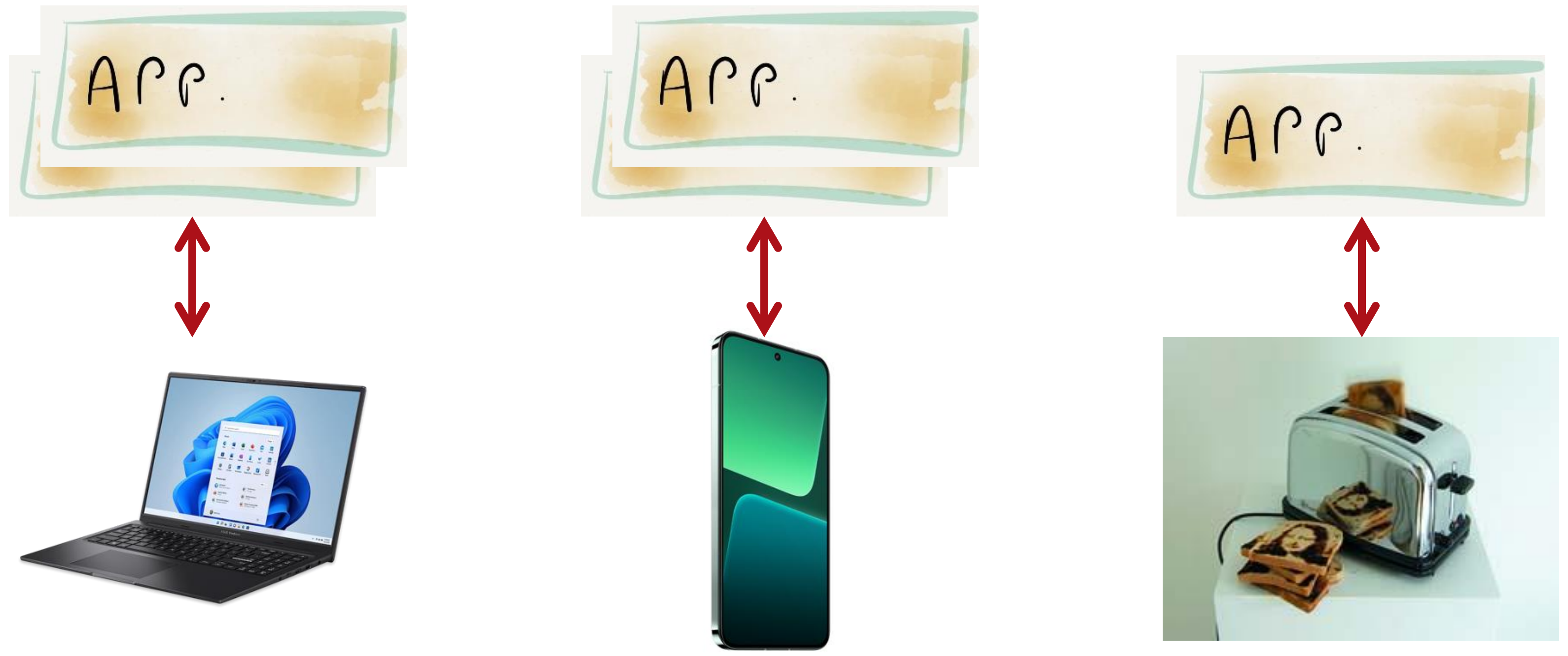
- A Virtual Machine (VM) is a software emulation of a physical computer that runs programs in an isolated environment
  - Instead of directly running on hardware, VM-based languages run on a software-based virtual environment.
- WHY? Platform Independence. **WORA (Write Once, Run Anywhere)**
  - Code written in VM-based languages can run on any system with the appropriate VM installed
  - VMs handle memory allocation and garbage collection, reducing the risk of memory-related errors
  - Execution within a VM provides a layer of isolation
- Java (JVM), C# (CLR), Python (Cpython, Jython, IronPython), Kotlin, JS, Ruby…

Introducing Java

- Java is a **modern**, platform independent, object-oriented programming language
  - Developed by SUN microsystems in 1995 (James Gosling). Java is now owned by Oracle.
  - Overarching design goal: **simplicity** and **reuse**
  - Originally designed for embedded devices… the first envisioned application was the Java toaster ☺
  - Next it was the web language of choice (applets)
  - Since evolved into a programming language of choice for **high reliability** and **good performance.** Now common place in:
    - Enterprise Systems (IBM/Oracle)
    - Mobile Devices (Android)
    - Embedded Devices (SmartTVs, IoT)
    - Financial, Medical, and Automotive domains

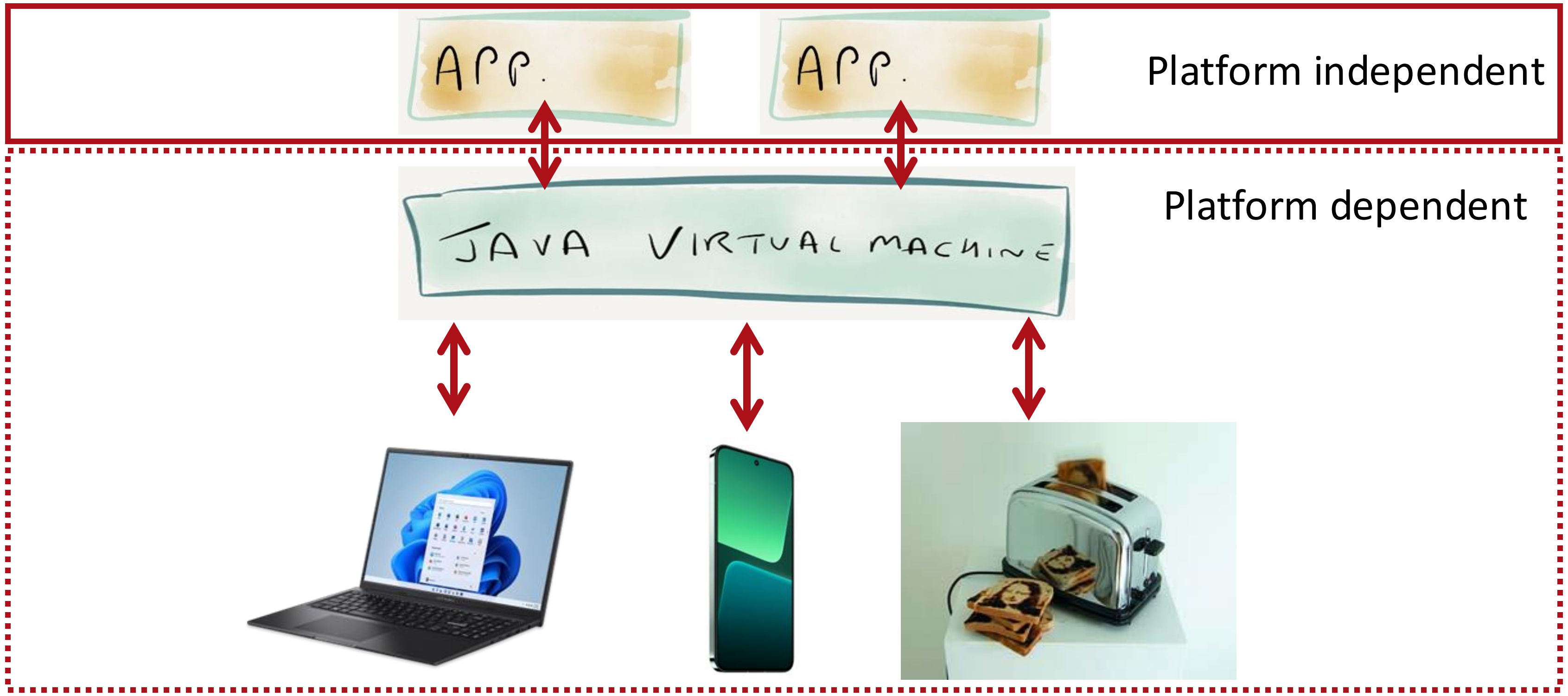
Introducing Java: open

- Java is a modern, **platform independent**, object-oriented programming language
    - **Open standards** allow interoperation and promote innovation



Introducing Java: abstract

- Java is a modern, **platform independent**, object-oriented programming language
  - Java uses a virtual machine to abstract over device operating systems



Java Virtual Machine (JVM)

- Abstracts over a device hardware
  - Processor
  - Memory
  - Input / Output
  - Graphical Interfaces…
- Contains a **virtual** computer processor!
  - Executes its own machine language known as **bytecode**
  - Very simple instructions, such as add, multiply, compare (c.f. machine language)
  - Java programs are compiled into bytecode by the developer
  - The JVM interprets these into whatever the hardware understands **at run time!**
  - Java bytecode is an example of an **intermediate language** (neither something you write in, nor something that is directly executed by a computer)

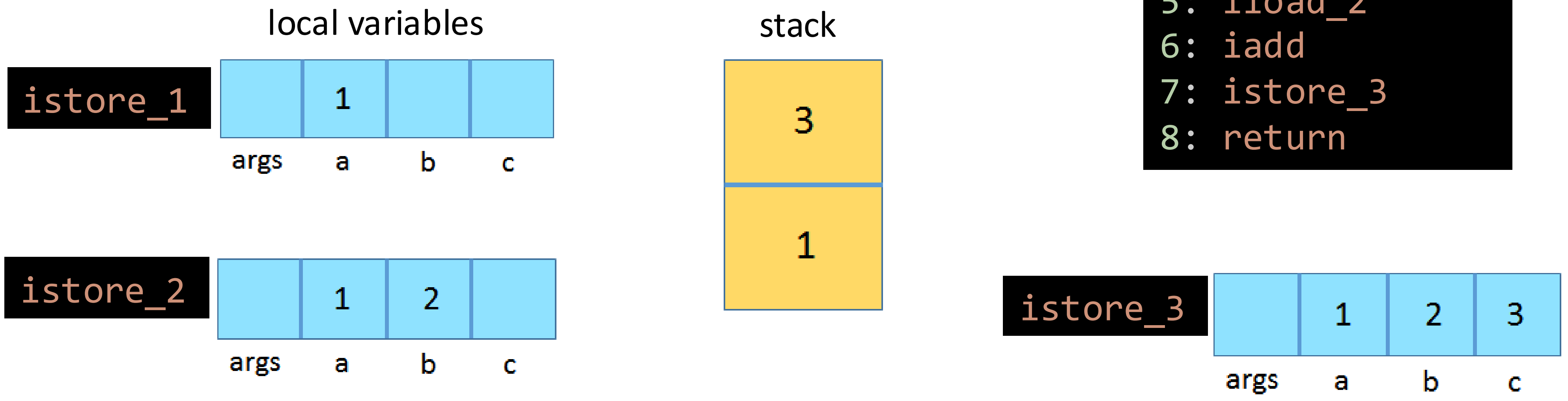
Java Machine Language: Bytecode

Simple Java Code

```
public static void main(String[] args) {
    int a = 1;
    int b = 2;
    int c = a + b;
}
```

ByteCode

```
0: iconst_1
1: istore_1
2: iconst_2
3: istore_2
4: iload_1
5: iload_2
6: iadd
7: istore_3
8: return
```



Head-to-Head: Performance Comparison

- **C++ vs Java**
  - I ran a **Bubble Sort** algorithm on an array of 13 integers…
  - Results shown in nanoseconds (1000 nsec = 0.001 msec)
    - Java: 4279 nsec
    - C++: 666 nsec (6 times faster)

“There are only two types of languages: the ones people complain about and the ones nobody uses” – Bjarne Stroustrup

The Java language

- The good news… **Java is based on the syntax of C And C++!**

```
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println( "Hello world" );
    }
}
```

The Java language: classes

- Every Java program is made up of one or more classes.
  - Classes are Java’s unit of modularity… they define **objects**
  - Remember the OO vision - so programs normally have loads of classes!
- The **class** keyword defines a unique name for the class you’re writing. Curly braces define the code that is part of that class.
- **One class per file, with filename matching as the class name!**

```
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println( "Hello world" );
    }
}
```

The Java language: methods

- The **main method** defines the start point for your program
  - This method always returns **void** in Java (they have no return value)
  - Parameter is an array of strings (command line arguments, like C)
- Methods are like functions in C
  - They are blocks of code that have names, parameters and return types

```java
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println( "Hello world" );
    }
}
```

The Java language: comments

- A single line comment begins with **//**
  - All text on the line after the **//** characters are ignored
- Java also supports the C style **/* */** comment blocks

```
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println( "Hello world" );
    }
}
```

The Java language: statements

- Method invocation (like calling a function in C) statements end with a semicolon, as they do in C and C++
- For example:
  - **System.out** is the name of an object (the console output stream)
  - **println** is the name of a method – note there's also a **print** method

```
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println("Hello world");
    }
}
```

The Java language: code blocks

- The end of the method and end of the class (respectively)
- As with C, its very important to maintain good code indentation
- **In Java, methods are always defined inside classes**

```java
public class HelloWorld
{
    public static void main( String[] arguments )
    {
        // this is where you say what you want the computer to do
        System.out.println("Hello world");
    }
}
```

Compiling a Java Program

- Create a text file with your favourite editor (e.g. VS Code)
  - Filename **must** match the name of the class it contains
  - Filename **must** end in .java
- Open a command line prompt
  - **shell** in Unix
  - **cmd** in Windows
- Change directory to the location of your file, then compile your program into bytecode:
  - **javac HelloWorld.java**
- Now start a Java virtual machine that interprets your program:
  - **java HelloWorld**

Lancaster University

Live demo…

Development tools

- **To develop in java, you need a Java Development Kit (JDK) and a text editor**
    - We recommend keeping it simple with Visual Studio Code
        - Available pre-installed on SCC Ubuntu image: https://mylab.lancaster.ac.uk
    - If you prefer to install on your laptop:
        - License free Open JDK: https://microsoft.com/openjdk
        - License free VS Code: https://code.visualstudio.com/
        - VS Code Java Extension Pack: https://code.visualstudio.com/docs/java/java-tutorial

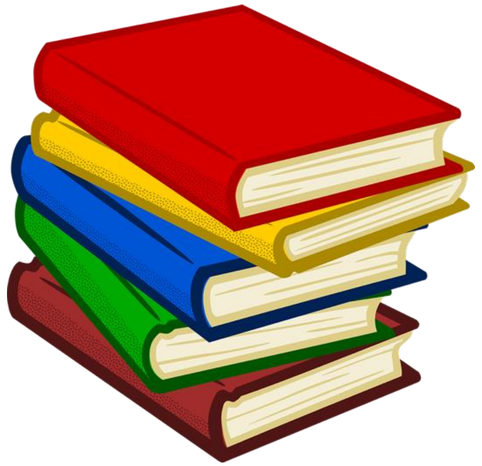
Syntax comparison

- Syntax comparison table:
  - Contains languages **C, C++, Java** and will include **Python**
  - Fill it as you go
  - We will come back to C and C++ equivalents for new concepts

Additional reading

- These books provide good additional reading:
  - **Head First Java** (2nd Edition) - ISBN: 1449331440
  - **Teach Yourself Java in 21 Days** - ISBN: 0134663667
- Both are **available for free** in the library's electronic collection
- Use these to reinforce your studies

- **Java Syntax Reference** PDF



Summary

- Today we learned that:
  - Java uses similar syntax to C and C++
  - Use “**javac** MyFile.java” to compile java programs (C equivalent: gcc)
  - Use “**java** MyFile” to run the compiled program (C equivalent: ./myfile)
  - Java is pure OO. All Java programs use **at least one class**.
  - One class per source file.