

Java Programming

Zheng-Liang Lu

Department of Computer Science & Information Engineering
National Taiwan University

Java 405
Spring 2024

Class Information

- Instructor: 盧政良 (Zheng-Liang Lu, Arthur)
- Email: arthurzllu@gmail.com
- The course website is
<https://www.csie.ntu.edu.tw/~d00922011/java.html>.
- All lecture slides are organized in English and will be modified if necessary.
- In-class examples may involve with high school math.

Teaching Philosophy

- I try to lower the barriers to entry.
 - Everything in my class is simple, essential, and general.¹
- I could provide extra studying materials as many as possible.
- I answer your questions.

¹“Simple is not easy. ... Easy is a minimum amount of effort to produce a result. ... **Simple is very hard.** Simple is the removal of everything except what matters. ...” by Christopher S. Penn.

Learning Tips

- Start with just **one** language and master it.
- Ask lots of questions; [Google](#) or [ChatGPT](#) first.
- Practice makes permanent (and hopefully, perfect).²
- It may take 10000 hours, more or less³; it is never too late.
- Grasp the fundamentals for long-term benefits; code from the bottom.
- Coding by hand sharpens proficiency.

²Try <https://leetcode.com/>.

³Malcolm Gladwell (2008): *Outliers*.

“Knowledge is of no value unless you put it into practice.”

– Anton Chekhov (1860-1904)

“Many roads lead to the path, but basically there are only two: reason and practice.”

– Bodhidharma

Grading Policy

- To acquire the class certificate, you need to finish all labs listed in the course page.⁴

⁴See [the list of programming labs](#).

Roll Call



```
1 class Lecture1 {  
2  
3     "Introduction"  
4  
5 }  
6  
7 // Keywords:  
8 public, class, static, void
```


PROGRAMMER



WHAT MY MOM THINKS I DO



WHAT MY FRIENDS THINK I DO



WHAT SOCIETY THINKS I DO



WHAT ARTISTS THINK I DO



WHAT I THINK I DO

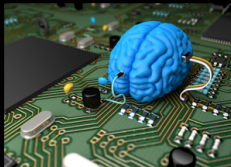


WHAT I ACTUALLY DO

Deep Learning



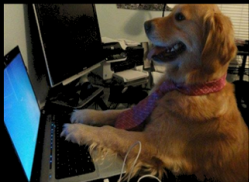
What society think I do



What my friends think I do



What other computer scientists think I do



What mathematicians think I do



What I think I do

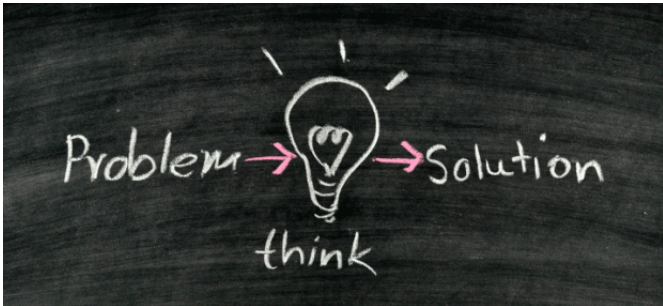
```
In [1]:  
import keras  
Using TensorFlow backend.
```

What I actually do

<http://p.migdal.pl/2017/04/30/teaching-deep-learning.html>

Goal: Problem Solving

- Programming is to provide computational solutions to real-world problems.
- The resulting solution is called a program.



Programs

- A program is a collection of **instructions** to perform a **specified task**.
- They are almost everywhere, for example,
 - video games (e.g. Pokémon Go, Travel Frog, ...);
 - operating systems (e.g. Linux, ...);
 - autonomous driving⁵ / auto-piloting (e.g. MRT, airplane, ...);
 - search engines (e.g. Google, ...);
 - robotics⁶;
 - computer malware⁷;
 - and more.⁸

⁵See <https://www.tesla.com/autopilot>.

⁶See <https://www.bostondynamics.com/> and watch videos from its official YouTube channel [here](#).

⁷See [malware](#) and [computer virus](#).

⁸See [Internet of Things \(IoT\)](#).

How to Execute Programs¹²

- Once the program is activated, both data and instructions are loaded from the disk into the **main memory**.
- We now call it a **process**, which is the smallest unit of resource allocation.⁹
- Then the instructions in the program are **scheduled** to be executed by the **CPU**.¹⁰
 - A CPU consists of 6 main components: arithmetic & logic units (ALUs), registers, control units, cache, buses, and clock.¹¹
- The immediate result is stored back to the main memory and further written into the disk if necessary.

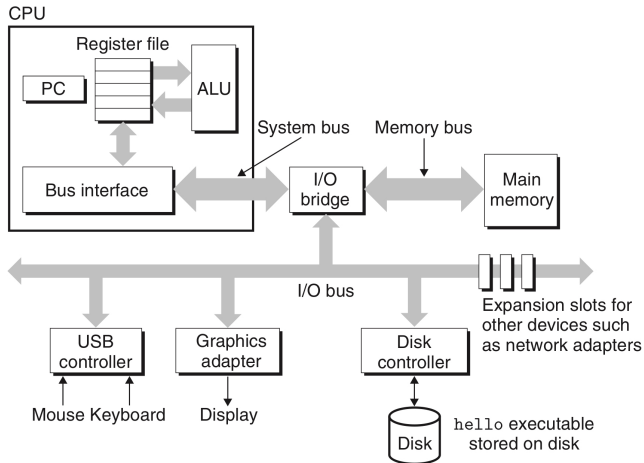
⁹See [https://en.wikipedia.org/wiki/Process_\(computing\)](https://en.wikipedia.org/wiki/Process_(computing)).

¹⁰See [https://en.wikipedia.org/wiki/Scheduling_\(computing\)](https://en.wikipedia.org/wiki/Scheduling_(computing)).

¹¹See https://en.wikipedia.org/wiki/Central_processing_unit.

¹²See [Inside Your Computer](#) by Bettina Bair and [Systems Architecture](#) by BBC.

Hardware Organization: Illustration¹³



¹³See Figure 1-4 in Bryant, p. 44.

Memory Hierarchy¹⁴

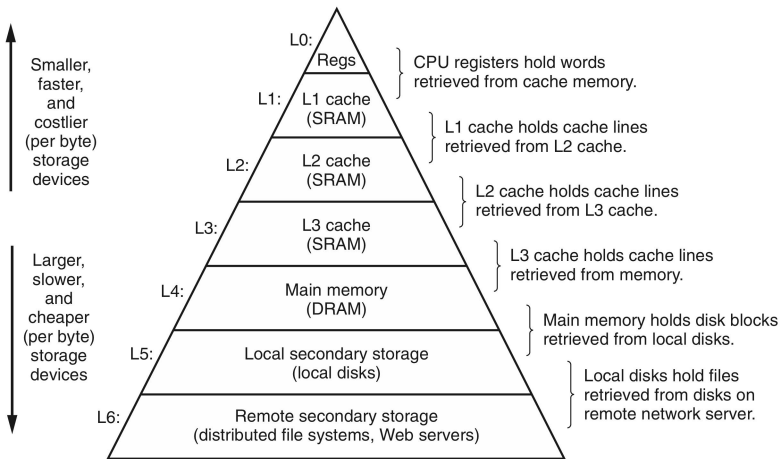


Figure 1.9 An example of a memory hierarchy.

¹⁴See Figure 1-9 in Bryant, p. 50.

Programming Languages

- A programming language is an artificial language to **communicate** with machines.¹⁵
- The elements of programming languages are **syntax** and **semantics**, used to control the behavior of machines.
- Top 20 programming languages can be found in [TIOBE](#).
- Every language originates from some reasons.

¹⁵See https://en.wikipedia.org/wiki/Programming_language.

A Short History¹⁷

- Gen 1: machine code.
- Gen 2: assembly code.
- Gen 3: high-level programming languages.
 - For example, C, C++, Java, C#, Python, JavaScript, etc.
- Gen 4.
 - For example, SQL¹⁶.

¹⁶Edgar Frank Codd (1974).

¹⁷See [generations of language](#) and [computer programming history](#).

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for MIPS)

```
swap:
  mult $2, $5, $4
  add  $2, $4, $2
  lw   $15, 0($2)
  lw   $16, 4($2)
  sw   $16, 0($2)
  sw   $15, 4($2)
  jr   $31
```

Assembler

Binary machine
language
program
(for MIPS)

```
000000001010001000000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
0000001111100000000000000001000
```

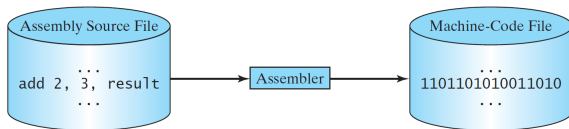
1st-Generation Languages

- The 1st-generation language is pure machine code, which represents instructions in just ones and zeros. (Why?)
- Each machine has its own instruction set.¹⁸
- So machine code is platform-dependent!
- More worse, the machine language is not human-readable.

¹⁸For example, the instruction set of the [x86](#) family and the [ARM](#) family are incompatible.

2nd-Generation Languages

- An **assembly language** uses mnemonics to represent instructions as opposed to the machine codes.
- Hence, assembly code is easier for human to read and write.
- When the assembly code is executed, it is then converted to the resulting machine code.

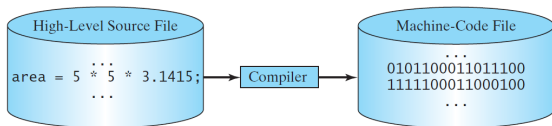


- It is worth to mention that the program flow of assembly code is almost close to the execution steps in the machine.¹⁹

¹⁹To be a hacker, you should learn assembly languages.

3rd-Generation Languages

- High-level languages are much closer to human languages by using English-like words, mathematical notations, and punctuation to write programs.



- For example, C²⁰, C++²¹, Python²², and Java²³.

²⁰Dennis Ritchie (1972).

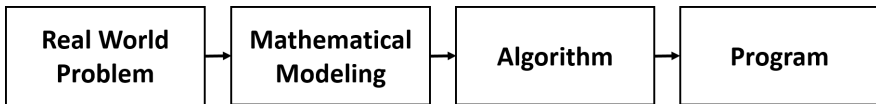
²¹Bjarne Stroustrup (1983).

²²Guido van Rossum (1991).

²³James Gosling (1995).

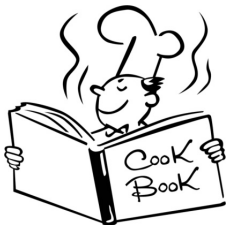
What Can A Program Do?

- A **program** is an implementation of an **algorithm** expressed in a specific **programming language**.



First Glance at Algorithms²⁴

- Simply put, an algorithm is a well-defined **procedure** that solves a specific category of problems, like a recipe or a cookbook.
- In general, an algorithm may take adequate information as its **input** and produce the correct result as its **output**.



²⁴See also [What's an Algorithm?](#) by David J. Malan.

Example: Largest Number

- Let A be a nonempty list of any real numbers.
 - Now try to propose an algorithm to find the largest number for any list of numbers.
-

Input: A .

Algorithm: your work here...

Output: the largest number in A .

- For example, assume that $A = \{1, 7, 9, -2, 4\}$.
- It is clear that the answer is 9.
- Can you teach the computer to repeat your idea?

Optimal Algorithm (Pseudo Code)

- For convenience, $A(i)$ denotes the i -th number of A and the symbol \leftarrow denotes a copy operator from right to left.
- Then the optimal algorithm could be as follows:

```
1 max ← A(1) // Initial guess, without loss of generality!
2 for i ← 2 ~ n
3     if A(i) > max
4         max ← A(i)
5     end
6 end
7 return max
```

Discussions

- In Line 1, why not `max ← 0`?
- You may extend this algorithm for following questions:
 - smallest number;
 - location of the largest (also smallest) number;
 - ... (more and more)
- We are eager for **efficient** algorithms.²⁵
 - Will “sorting the list and then the largest number could be the last number” be more efficient than the previous one?

²⁵We will show you the **complexity analysis** in the end of the chapter of loops.

*“Computers are good at following instructions, but **not at reading your mind.**”*

– Donald Knuth (1938-)

*“There are two ways of constructing a software design: One way is to make it so **simple** that **there are obviously no deficiencies**, and the other way is to make it so **complicated** that **there are no obvious deficiencies**. The first method is far more difficult.”*

– Tony Hoare (1934-)

Alan Turing

- Provided a formalization of the concepts of **algorithm** and computation with the **Turing machine**²⁶, which can be considered a model of a general-purpose computer.
- Proposed the famous question: “*Can machines think?*”²⁷
 - Well-known as the Turing test.
- Turing Award is recognized as the highest distinction in computer science and the “Nobel Prize of computing”.²⁸

²⁶Turing (1936). Try [this](#) google doodle.

²⁷Turing (1950). You could find the paper [here](#). See also [Turing Test](#).

²⁸See https://en.wikipedia.org/wiki/Turing_Award#Recipients. 



- You may watch [The Imitation Game](#) (2014).
- Britain's £50 note will honor computing pioneer Alan Turing.²⁹

²⁹See [Alan Turing to be the Face of New £50 Note](#) by Bank of England.

About Java³⁰

- Java is one of general-purpose programming languages, supporting the **object-oriented programming (OOP)** paradigm.
- Java was first released by Sun Microsystems back in 1995 and is now maintained by [Oracle \(ORCL\)](#) after the year of 2010.
- Java was intended to let application developers **write once, run anywhere (WORA)**.
- Nowadays, Java is widely used in web apps (especially the back-end services), mobile apps (for example, Android apps) and embedded systems.

³⁰See [Java Version History](#).

Java Virtual Machine (JVM)³³

- JVM is a software program, not a physical machine.
- JVM translates Java **bytecodes** into machine codes which can be executed by the host platform.³¹
- JVM verifies all bytecodes before the program is executed, so that no user program can crash the host machine.³²
- You can find further details about JVM from its [specification](#).

³¹For example, Windows, Linux, and MacOS for desktop/laptop computers; Android and iOS for mobile devices.

³²See [Security Vulnerabilities of Java](#).

³³See http://en.wikipedia.org/wiki/Java_virtual_machine.

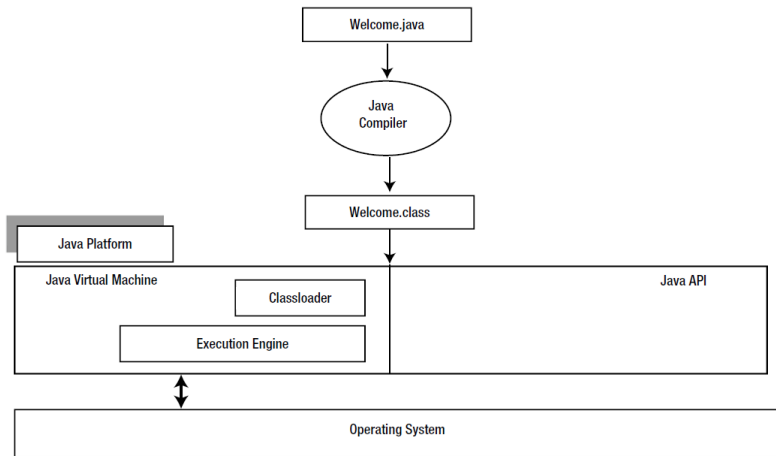
Software Installation

- First download and install the **Java Development Kit (JDK)**.³⁴
- Then choose an **integrated development environment (IDE)** which may provide comprehensive facilities, say debugger, code completion, and build automation tools.
- In this course, I will use Eclipse which is a free and powerful IDE.³⁵
- Uncheck `Create module-info.java file` for every new project; simply remove it when the “module-info.java” file is already present in the project.

³⁴See <https://www.oracle.com/java/technologies/downloads/>.

³⁵You may try the following alternatives: [NetBeans](#), [IntelliJ IDEA](#) and [Visual Studio Code](#) with proper packages.

Workflow³⁶



³⁶See Figure 2-19 in Sharan, p. 59.

First Program: Hello, World³⁷

```
1 public class HelloWorldDemo {
2
3     public static void main(String[] args) {
4
5         // Print "Hello, Java." on the screen.
6         System.out.println("Hello, Java.");
7
8     }
9
10 }
```

- **class**: define a new class.
- **public**: accessible to anyone.
- **static**: can be called without having any object.
- **void**: no return value.

³⁷See [Hello, World!"_program">https://en.wikipedia.org/wiki/"Hello, World!"_program](https://en.wikipedia.org/wiki/).

Discussions (1/3)

- A class is an entity of Java programs.
- A Java program can consist of many classes.³⁸
- To be executable, a program needs one special **method**³⁹ called the **main** method as the **entry point** of the program.
- **System.out** refers to the standard output device, say the monitor screen.
- The `println` method is used to output the **string** you offer.
- **Every statement should end with a semicolon (;).**

³⁸We will discuss more later in the OOP chapter.

³⁹Also called functions and subroutines in other programming languages.

Discussions (2/3)

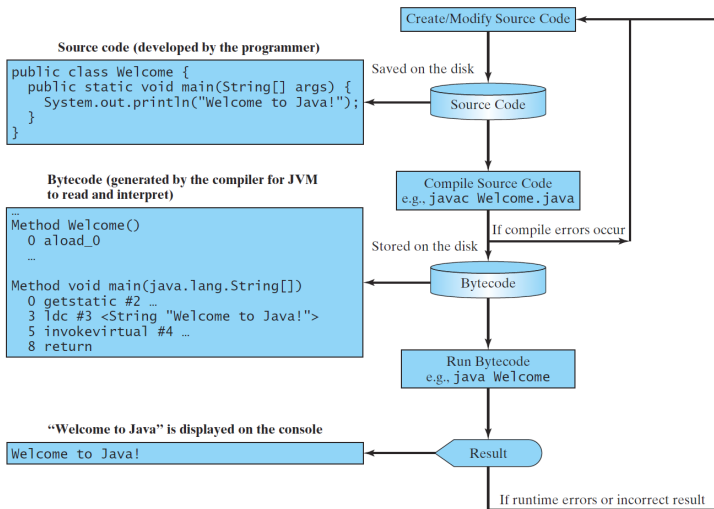
- For now, one program has only one **public** class.
- The only **public** class **should have the same name as the filename**.
 - For example, the **public HelloWorldDemo** class should be in the source file whose filename must be *HelloWorldDemo.java*.
- Note that you cannot have more than one **public** class in single java file.
- The keyword **public** is one of **access modifiers**, allowing you to control the **visibility** of classes and also its members.⁴⁰

⁴⁰We will visit more access controls later.

Discussions (3/3): Table of Special Characters

Symbol	Description
{ }	Denote a scope to enclose statements; enumerate a set of objects.
()	Mostly used with expressions and methods.
[]	Mostly used with arrays.
" "	Enclose a string.
;	Mark the end of a statement.
//	Precede a comment line.
/* ... */	Used as block comments.

Recall How To Run A Java Program⁴¹



⁴¹See Figure 1.14 in YDL, p.20.

Digression: Bugs⁴²

- A bug is an error, flaw, failure, or fault in a computer program or system, producing an incorrect or unexpected result, or misbehaving in unintended ways.
 - **Compile-time error**: most of them are syntax errors.
 - **Runtime error**: occurs when Java program runs, e.g. 1/0.
 - **Logic error**: introduced by implementing the functional requirement incorrectly.
- Note that logic (semantic) errors are the obscurest because they are hard to be found.

⁴²See [Computer Bugs](#).

*“If debugging is the process of **removing** software bugs, then programming must be the process of **putting** them in.”*

– Edsger W. Dijkstra (1930–2002)

“Why do we fall sir? So that we can learn to pick ourselves up.”

– Alfred Pennyworth, *Batman Begins* (2005)

Programming Style

- A good programming style makes programs easy to read and helps programmers prevent from errors.
 - For example, [Google Java Style Guide](#) and [Code Style of Java](#) from IntelliJ IDEA.
- In particular, we use **indentation** to enhance the structural relationships by visual.
 - For Eclipse users, select the code snippet and press ctrl + I to make correct indentation.
- Be consistent through the whole program, even in the project!