Java Programming

Zheng-Liang Lu

Department of Computer Science & Information Engineering
National Taiwan University

Java 311
Spring 2019

Class Information

- Instructor: Zheng-Liang Lu
- Email: d00922011@ntu.edu.tw
- The course website is

  http://www.csie.ntu.edu.tw/˜d00922011/java.html.
- All lecture slides are organized in English and will be modified if necessary.

Prerequisites

- This class is organized for students who are not EE/CS majors.
- No programming experience required; it would be helpful if you have some programming experiences.
- Examples may involve with high school math.
- I promise to keep everything **simple** in this class.[1]

[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. . . ." See `http://www.christopherspenn.com/2010/11/simple-is-not-easy/`.

Teaching Philosophy

- First, I try to lower the barriers to entry.
- Second, I provide resources as many as possible.
- Third, I answer your questions.

Learning Tips

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

[2] It sharpens proficiency and you'll need it to get a job. For example, technical interview of Google.

"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 certificate, you need to finish 5 labs listed in the course page[3].

[3] See `https://www.csie.ntu.edu.tw/~d00922011/java.html#lab`

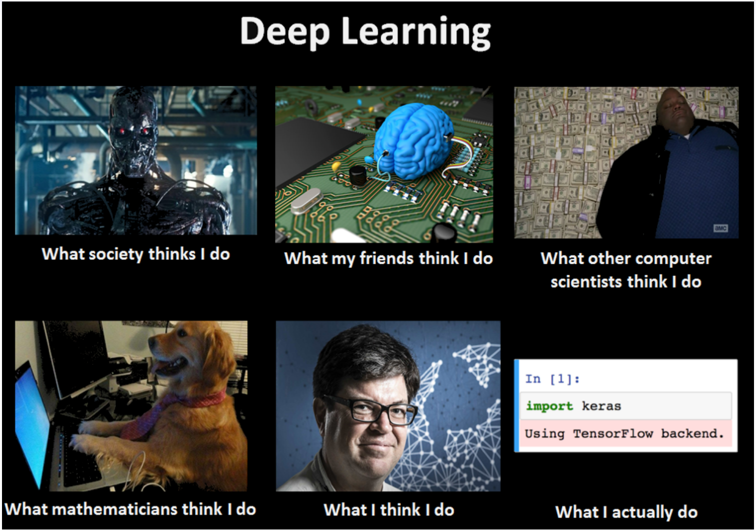
Roll Call

```
class Lecture1 {

                    "Introduction"

}

// Keywords:
public, class, static, void
```

What Is Programming?

- Programming is the activity of writing a sequence of instructions to tell a machine to perform a specific task.
  - A sequence of instructions → program
  - A set of well-defined notations used to write a program → programming language
  - People who write programs → ~~programmer~~ designer
- Writing codes is not what the CS people work for. We are writing codes to make a better world.

PROGRAMMER
WHAT MY MOM THINKS I DO
WHAT MY FRIENDS THINK I DO
WHAT SOCIETY THINKS I DO
[JavaScript Application]
Hello, world!
OK
WHAT ARTISTS THINK I DO
WHAT I THINK I DO
WHY WON'T YOU COMPILE?
WHAT I ACTUALLY DO


Deep Learning
What society thinks 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
What I actually do

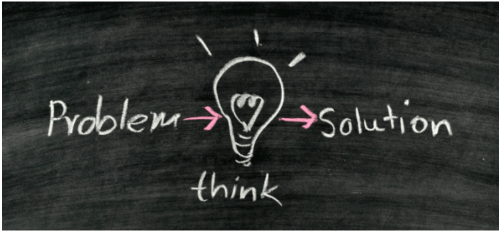
```
In [1]:
import keras
Using TensorFlow backend.
```

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

In Practice

Programming is to provide a solution to a real-world problem using computational models supported by programming languages.

- The computational solution is a program.



Programs

- A program is a sequence of instructions, written in an artificial language, to perform a specified task by a machine.
- They are almost everywhere, for example,
  - Video games (e.g. Pokémon Go, Travel Frog, ...);
  - Operating systems (e.g. Linux, ...);
  - Transportations (e.g. traffic light, MRT, airplane, ...);
  - Search engine (e.g. Google, ...);
  - Robotics[4];
  - Computer virus[5].

[4] See https://www.bostondynamics.com/.

[5] See http://en.wikipedia.org/wiki/Computer_virus.

How and Where The Programs Run

- 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.[6]
- Then the instructions in the program are scheduled to be executed by the CPU.[7]
  - A CPU contains arithmetic & logic units (ALUs), control units, and registers.[8]
- The immediate result is stored back to the main memory and further written into the disk if necessary.

[6] See `https://en.wikipedia.org/wiki/Process_(computing)`.

[7] See `https://en.wikipedia.org/wiki/Scheduling_(computing)`.

[8] See `https://en.wikipedia.org/wiki/Central_processing_unit`.

Memory Hierarchy[9]

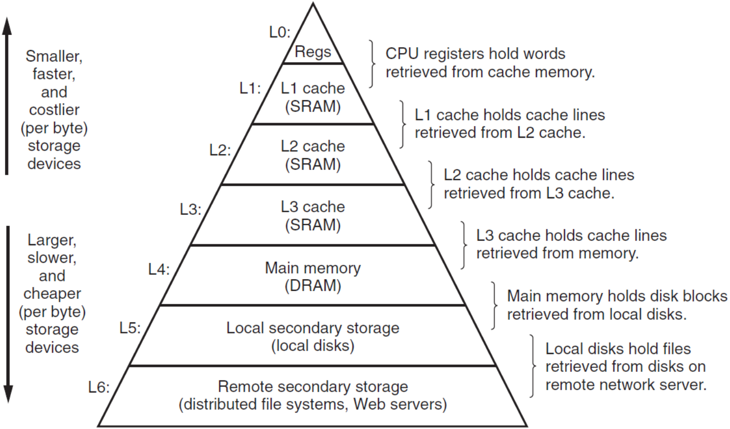

Smaller,
faster,
and
costlier
(per byte)
storage
devices
Larger,
slower,
and
cheaper
(per byte)
storage
devices
L0:
Regs
L1:
L1 cache
(SRAM)
L2:
L2 cache
(SRAM)
L3:
L3 cache
(SRAM)
L4:
Main memory
(DRAM)
L5:
Local secondary storage
(local disks)
L6:
Remote secondary storage
(distributed file systems, Web servers)
CPU registers hold words retrieved from cache memory.
L1 cache holds cache lines retrieved from L2 cache.
L2 cache holds cache lines retrieved from L3 cache.
L3 cache holds cache lines retrieved from memory.
Main memory holds disk blocks retrieved from local disks.
Local disks hold files retrieved from disks on remote network server.

Figure 1.9 **An example of a memory hierarchy.**

[9] See Figure 1-9 in Bryant, p. 14.

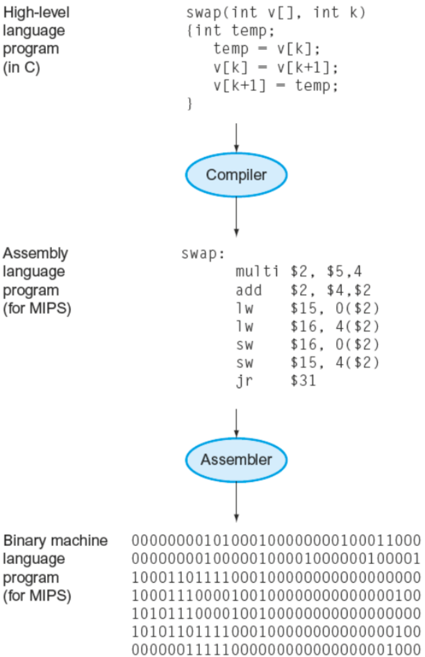
Programming Languages

- A programming language is an artificial language to communicate with machines.
- Recall how you learned the 2nd nature language when you were a kid.
- Programming languages → syntax and semantics
  - Used to express algorithms
  - Used to control the behavior of machines
- How many programming languages in the world?
  - More than 1000.
  - Top 20 programming languages can be found in TIOBE.
  - Java: top 3
- Note that every language originates from reasons.

History[10]

- 1st generation: machine code
- 2nd generation: assembly code
- 3rd generation: high-level programming languages
- Post 3rd generations
- Java is one of the 3rd-generation programming languages.

[10] See https://en.wikibooks.org/wiki/A-level_Computing_2009/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Fundamentals_of_Computer_Systems/Generations_of_programming_language.



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;
}
```

Assembly language program (for MIPS)

```
swap:
      multi $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
```

Binary machine language program (for MIPS)

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

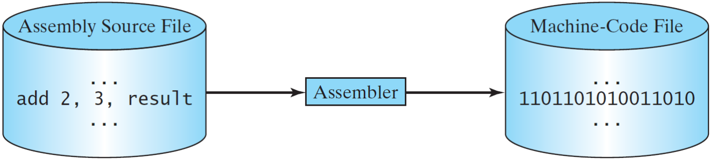
1st-Generation Programming Languages

- Computers understand instructions only in binary, which is a sequence of 0's and 1's. (Why?)
- Each computer has its own set of instructions.[11]
- So the programs at the very early stage were machine-dependent.
- These are so-called the machine language, aka machine code.
- Pros:
  - Most efficient for machines
- Cons:
  - Hard to program for human
  - Not portable
- Still widely used in programming lower level functions of the system, such as drivers, interfaces with firmware and hardware.

[11] For example, X86 and ARM.

2nd-Generation Programming Languages

- An assembly language uses mnemonics[12] to represent instructions as opposed to the machine codes.
- Hence, the code can be read and written by human programmers.
- Yet, it is still machine-dependent.
- To run on a computer, it must be converted into a machine readable form, a process called assembly.



- More often used in extremely intensive processing such as games, video editing, graphic manipulation/rendering.
- Note that machine languages and assembly languages are also known as low-level languages.

[12] Easy to recognize and memorize.

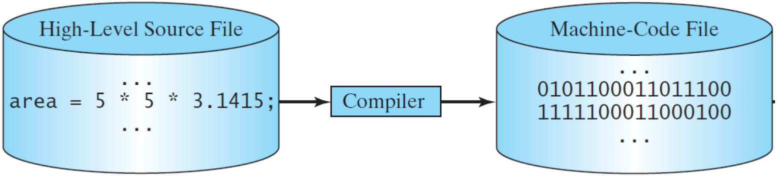
3rd-Generation Programming Languages

- High-level programming languages use English-like words, mathematical notation, and punctuation to write programs.
- They are closer to human languages.
- Pros:
  - Portable, machine-independent
  - Human-friendly
- For example, C[13], C++[14], and Java[15].

[13] Dennis Ritchie (1973).

[14] Bjarne Stroustrup (1983).

[15] James Gosling (1995).

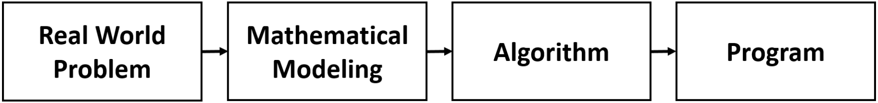


- Note that the machines understand and execute only the machine codes as before.
- The translation is accomplished by a compiler, an interpreter, or a combination of both.[16]

[16] If you've learned C, you should take a look at the design of compiler.

What Can A Program Do?

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


Real World Problem

Mathematical Modeling

Algorithm

Program

Algorithms In A Nutshell

- An algorithm is a well-defined computational procedure that takes a set of values as input and produces a set of values as output.
- Simply put, an algorithm is a procedure that solves a particular class of problems, such as a cookbook.

Properties of Algorithms[18]

- An algorithm must possess the following properties[17]:
  - Input and output.
  - Correctness.
  - Definiteness: basic instructions provided by a machine, e.g. $+ - \times \div$.
  - Effectiveness: action which can be completed by combination of basic instructions.
  - Finiteness: resource requirement, especially time and space.
- Note that an algorithm is not necessarily expressed in a specific programming language.
  - Could use human languages, graphs, and pseudo codes.

[17] Alan Turing (1912–1954).

[18] Donald E. Knuth (1938–).

Example

- Organize an algorithm that finds the greatest element in the input list, say A.

Input: A (a list of *n* numbers)
Output: max (the greatest element in A)

- Can you provide a procedure to determine the greatest element?
- For all cases?

My Solution

- The first element of A can be fetched by calling A(1).
- Let ← be the assignment operator in the following pseudo code.

```
max <- A(1)
for i <- 2 ~ n
    if A(i) > max
        max <- A(i)
    end
end
return max
```

- How to find the minimal element?
- How to find the location of the greatest element?
- Why not max ← 0?

"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 universal computation model for general-purpose computers.
  - As known as Turing machine.[19]
  - Also first proved that there exist problems which are undecidable by Turing machine.[20]
- Father of computing theory and artificial intelligence.[21,22]
- The Turing Award is generally recognized as the highest distinction in computer science and the"Nobel Prize of computing".[23]
- You may watch The Imitation Game (2014).

[19] Try this toy example by Google for celebration of Turing's birthday.

[20] See Halting problem.

[21] See Turing test.

[22] See Pretty sure Google's new talking AI just beat the Turing test.

[23] See https://en.wikipedia.org/wiki/Turing_Award#Recipients.

Alan Turing

What Is Java?

- Java is one of general-purpose programming languages.
- It has features to support programming based on the object-oriented programming.
- The initial version of the Java platform was released by *Sun Microsystems* in 1995, now owned by Oracle Corporation since January 2010.
- Slogan: “Write once, run anywhere, ” that is, write a Java program once and run it on any platform. (How?)

Java Virtual Machine (JVM)[27]

- Java Virtual Machine (JVM) is used to translate Java bytecodes into machine codes according to the host platform.[24]
- Clearly, JVM is a software program, not a physical machine.
- To enhance the security, the JVM verifies all bytecodes before the program is executed.[25]
- No user program can crash the host machine.[26]

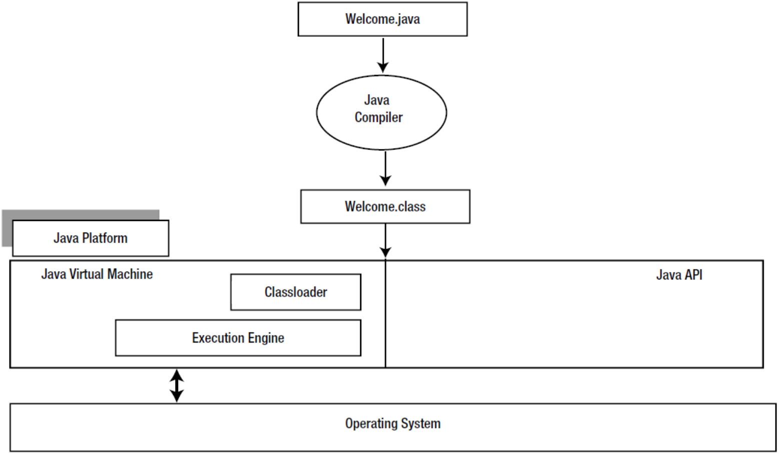
[24] For example, Windows, Linux, MacOS, Android, iOS, et cetera.

[25] However, there are a number of possible sources of security vulnerabilities in Java applications. See https://en.wikipedia.org/wiki/Java_security#Potential_sources_of_security_vulnerabilities_in_Java_applications.

[26] Also see https://en.wikipedia.org/wiki/Virtualization.

[27] See http://en.wikipedia.org/wiki/Java_virtual_machine.

Compiling and Running A Java Program[28]



[28] See Figure 2-19 in Sharan, p. 59.

Integrated Development Environment (IDE)

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

- An IDE normally consists of a source code editor, build automation tools and a debugger.
- Most modern IDEs offer the intelligent code completion.

In this class, we need Java Development Kit (JDK) and Eclipse IDE for Java Developers.

Example: Hello, Java[29]

Write a program which says hello.

```java
public class HelloJavaDemo {
    public static void main(String[] args) {
        // Print "Hello, Java." on the screen.
        System.out.println("Hello, Java.");
    }
}
```

Keywords are marked in violet.

- class: declare a new class followed a distinct class name.
- public: can be accessed by any other class.
- static: can be called without having to instantiate a particular instance of the class.
- void: do not return a value.

[29] See urlhttps://en.wikipedia.org/wiki/%22Hello,_World!%22_program/.

- Every statement ends with a semicolon (;).
- A special method **main** is used as the entry point of the program.
- **System.out** refers to the standard output device, normally the screen.
- **println**() is a method within *System.out*, which is automatically imported by default.

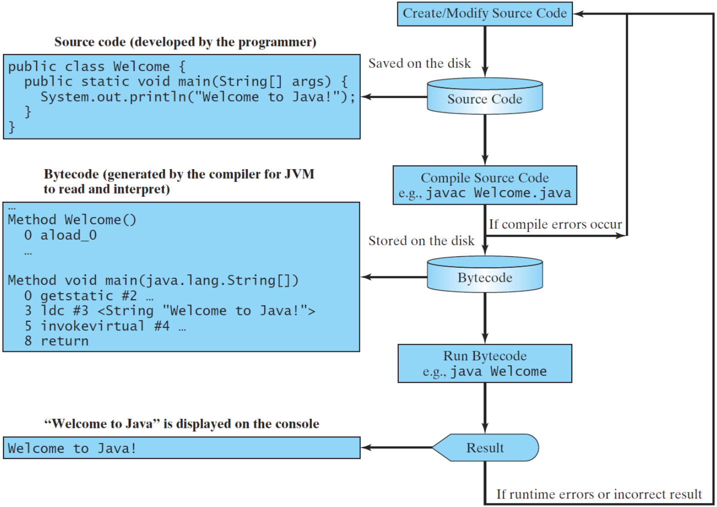
Public Classes

The public keyword is one of access modifiers[30], which allows the programmer to control the visibility of classes and also members.

- One public class in the java file whose filename is identical to that of the public class.
- There must be at most one public class in one jave file.

[30] We will visit the access control later when it comes to encapsulation.

How To Run A Java Program[31]



Source code (developed by the programmer)

```
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Bytecode (generated by the compiler for JVM to read and interpret)

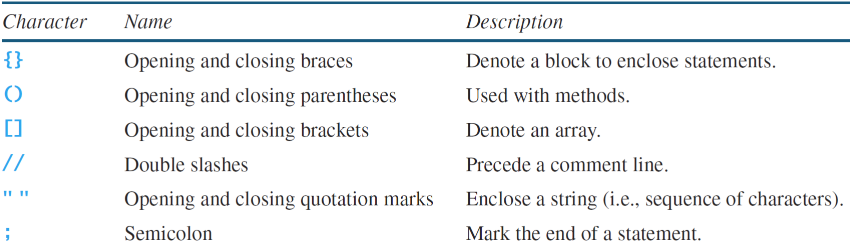
```
…
Method Welcome()
  0 aload_0
  …

Method void main(java.lang.String[])
  0 getstatic #2 …
  3 ldc #3 <String "Welcome to Java!">
  5 invokevirtual #4 …
  8 return
```

"Welcome to Java" is displayed on the console

```
Welcome to Java!
```

[31] See Figure 1.14 in YDL, p.20.

Table of Special Characters[32]



Character	*Name*	*Description*
`{}`	Opening and closing braces	Denote a block to enclose statements.
`()`	Opening and closing parentheses	Used with methods.
`[]`	Opening and closing brackets	Denote an array.
`//`	Double slashes	Precede a comment line.
`" "`	Opening and closing quotation marks	Enclose a string (i.e., sequence of characters).
`;`	Semicolon	Mark the end of a statement.

[32] See Table 1.2 in YDL, p.18.

Bugs

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 since they are hard to be found.

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

– Edsger W. Dijkstra (1930–2002)

Programming Style

- Good programming style makes a program easy to read and helps programmers prevent from errors.
    - Indentation: enhance the structural relationships by visual
    - Curly braces by: next-line style or end-of-line style
        - Be consistent through the whole program!
- For example, Google Java Style.

```
class Lecture2 {

            "Data types, Variables, and Operators"

}

// Keywords:
byte, short, int, long, char, float, double, boolean, true,
    false, import, new
```

Example

Given the radius of a circle, say 10, determine the area.

Recall that a program comprises data and algorithms.

- How to store the data?
  → variables, data types
- How to compute the area?
  → arithmetic operators
- How to show the result?
  → System.out.println()

```java
public class ComputeAreaDemo {
    public static void main(String[] args) {

        // input
        int r = 10;

        // algorithm
        double area = r * r * 3.14;

        // output
        System.out.println(area);
    }
}
```

- The type int and double are two of primitive data types.
- We use two variables *r* and *area*.