Java Programming

Zheng-Liang Lu

Department of Computer Science & Information Engineering
National Taiwan University

Java 310
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/](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 at least 70 points:
  • Programming assignments (30 pts)
    • Practice makes perfect.
  • Final exam (70 pts)
    • On-site programming.
    • 2 hours.
    • Open everything (honor code).
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

WHAT ARTISTS THINK I DO

WHAT I THINK I DO

WHAT I ACTUALLY DO

JavaScript Application

Hello, world!

OK

WHY WON'T YOU COMPILE?
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.
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\(^3\)
- Computer virus\(^4\)

\(^3\)See https://www.bostondynamics.com/.
\(^4\)See http://en.wikipedia.org/wiki/Computer_virus
Once the program is activated, both data and instructions are loaded from the disk into the main memory.

Now we call them the processes.\(^5\)

CPUs include the arithmetic & logic units (ALUs) and a limited number of registers.

- ALU is responsible for the computational power.
- Registers store the data to be used instantaneously.

The results are written back to the main memory and further stored into the disk if necessary.

\(^5\)The “process” is a formal term used in operating systems.

\(^6\)You may refer to any class for an introduction to computer system. For example, [www.csie.ntu.edu.tw/~d00922011/ics.html].
Memory Hierarchy\textsuperscript{7}

![Memory Hierarchy Diagram]

\textit{Figure 1.9} An example of a memory hierarchy.

\textsuperscript{7}See Figure 1-9 in Bryant, p. 14.
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

- 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.

---

High-level language program (in C)

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

Assembly language program (for MIPS)

```assembly
swap:
    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)

```
00000000010100010000000000100011000
000000000100000001000000000100010001
1001110111100010000000000000000000000
1000110000100100000000000000000100
1010111000010010000000000000000000
1010110111100010000000000000000000100
0000001111100000000000000000010000
```
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.\(^9\)
- 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.

\(^9\) For example, X86 and ARM.
2nd-Generation Programming Languages

- An **assembly language** uses mnemonics 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.
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\textsuperscript{11}, C++\textsuperscript{12}, and Java\textsuperscript{13}.

\textsuperscript{11}Dennis Ritchie (1973).
\textsuperscript{12}Bjarne Stroustrup (1983).
\textsuperscript{13}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.\(^\text{14}\)

\(^{14}\)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.
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.
An algorithm must possess the following properties:\footnote{Alan Turing (1912–1954)}:\footnote{Donald E. Knuth (1938–)}:

- **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.
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 situations?
My Solution

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

```plaintext
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**.
  - Also first proved that there exist problems which are undecidable by Turing machine.
- Father of **computing theory and artificial intelligence**.
- **Turing Award** of ACM
- You may watch **The Imitation Game** (2014)

---

17 Try this [toy example](#) by Google for celebration of Turing’s birthday.
18 See [Halting problem](#).
19 See [Turing test](#).
20 See [Pretty sure Google's new talking AI just beat the Turing test](#).
21 Association for Computing Machinery.
Alan Turing
What Is Java?

• Java is a general purpose programming language.
• It has features to support programming based on the object-oriented paradigms.
• The initial version of the Java platform was released by Sun Microsystems in 1995.\textsuperscript{22}
• At the very early stage, this language was called Oak and it was meant to be used in set-top boxes for televisions.
• Slogan: “Write once, run anywhere.”
• That is, Write a Java program once and run it on any platform. (How?)

\textsuperscript{22}Now owned by Oracle Corporation, since January 2010.
Java Virtual Machine (JVM)\textsuperscript{26}

- Java Virtual Machine (JVM) is used to translate Java bytecodes into machine codes according to the platform\textsuperscript{23} it is running on.
- Clearly, JVM is a software program, not a physical machine.
- To enhance the security, the JVM verifies all bytecodes before the program is executed.\textsuperscript{24}
- No user program can crash the host machine.\textsuperscript{25}

\textsuperscript{23}For example, Windows, Linux, MacOS, Android, iOS, et cetera.
\textsuperscript{24}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.
\textsuperscript{25}Also see https://en.wikipedia.org/wiki/Virtualization.
\textsuperscript{26}See http://en.wikipedia.org/wiki/Java_virtual_machine.
Compiling and Running A Java Program

See Figure 2-19 in Sharan, p. 59.
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: A Simple Template

Write a program which says hello to Java.

```java
public class HelloJava {
    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
• 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\(^{28}\), 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 java file.

\(^{28}\)We will visit the access control later when it comes to encapsulation.
How To Run A Java Program

Source code (developed by the programmer)

```java
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

```java
Welcome to Java!
```

See Figure 1.14 in YDL, p.20.
### Table of Special Characters

<table>
<thead>
<tr>
<th>Character</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{}</code></td>
<td>Opening and closing braces</td>
<td>Denote a block to enclose statements.</td>
</tr>
<tr>
<td><code>()</code></td>
<td>Opening and closing parentheses</td>
<td>Used with methods.</td>
</tr>
<tr>
<td><code>[]</code></td>
<td>Opening and closing brackets</td>
<td>Denote an array.</td>
</tr>
<tr>
<td><code>//</code></td>
<td>Double slashes</td>
<td>Precede a comment line.</td>
</tr>
<tr>
<td><code>&quot;&quot;&quot;&quot;</code></td>
<td>Opening and closing quotation marks</td>
<td>Enclose a string (i.e., sequence of characters).</td>
</tr>
<tr>
<td><code>;</code></td>
<td>Semicolon</td>
<td>Mark the end of a statement.</td>
</tr>
</tbody>
</table>

---

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 the programmer by implementing the functional requirement incorrectly

Note that logic errors are the obscurest in programs 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.”

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](https://google.github.io/styleguide/javaguide.html).