Java Programming

Zheng-Liang Lu

Department of Computer Science & Information Engineering
National Taiwan University

Online Course

Class Information

- Instructor: Zheng-Liang Lu (Arthur)
- Email: arthurzllu@gmail.com
- The course website for the physical course is

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

Prerequisites

- No programming experience required; it would be helpful if you have some.
- In-class examples may involve with high school math.
- Everything is simple and essential 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

- I try to lower the barriers to entry.
- I provide resources as many as possible.
- 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).[2]
- 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.[3]

[2] Try `https://leetcode.com/`.

[3] It sharpens proficiency and you'll need it to get a job.

"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 labs listed in the course page.[4]

[4] See `https://www.csie.ntu.edu.tw/~d00922011/java.html#Programming-Labs`.

Roll Call

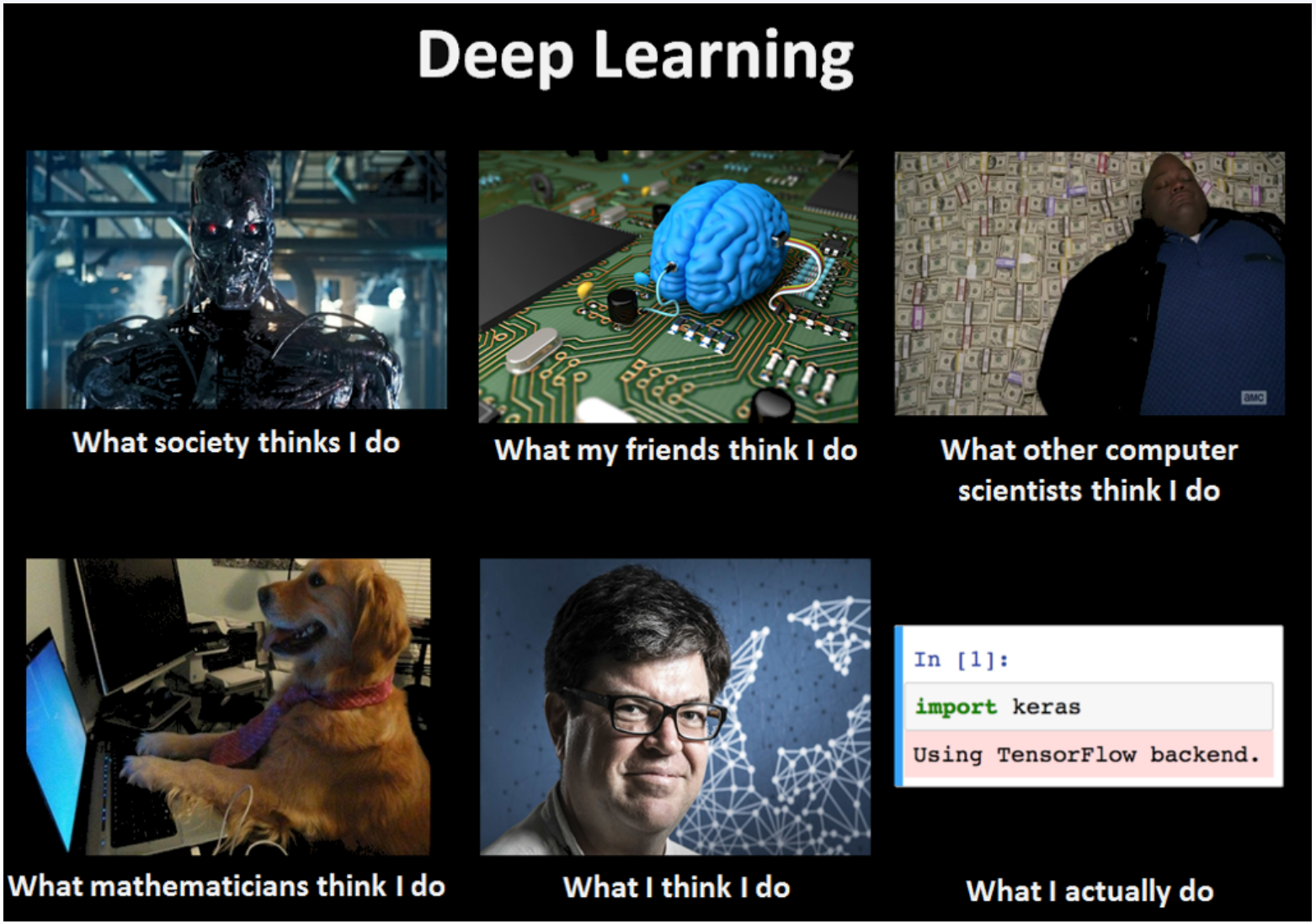
```
class Lecture1 {

                    "Introduction"

}

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

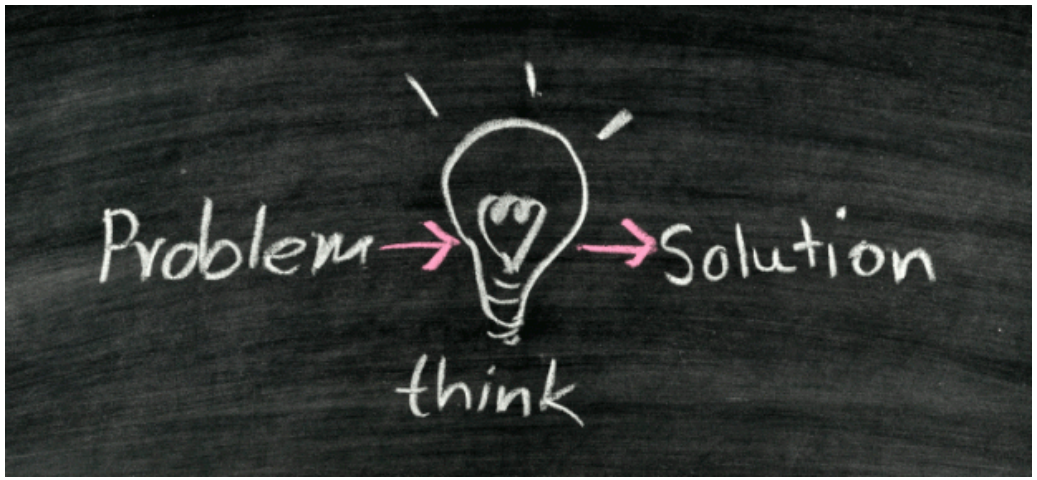

PROGRAMMER
WHAT MY MOM THINKS I DO
WHAT MY FRIENDS THINK I DO
WHAT SOCIETY THINKS I DO
[JavaScript Application]
Hello, world!
OK
WHY WON'T YOU COMPILE?
WHAT ARTISTS THINK I DO
WHAT I THINK I DO
WHAT I ACTUALLY DO



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

Goal

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



Programs

- A program is a collection of instructions, written in an artificial language, to perform a specified task executed by computers.
- 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[5];
  - Computer virus[6];
  - and more.

[5] See `https://www.bostondynamics.com/` and watch `https://www.youtube.com/watch?v=7Q3YW-3KCzU`.

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

How to Run Programs[10]

- 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.[7]
- Then the instructions in the program are scheduled to be executed by the CPU.[8]
  - A CPU contains arithmetic & logic units (ALUs), control units, and registers.[9]
- The immediate result is stored back to the main memory and further written into the disk if necessary.

---

[7] See `https://en.wikipedia.org/wiki/Process_(computing)`.
[8] See `https://en.wikipedia.org/wiki/Scheduling_(computing)`.
[9] See `https://en.wikipedia.org/wiki/Central_processing_unit`.
[10] See `http://ed.ted.com/lessons/inside-your-computer-bettina-bair`.

Memory Hierarchy[11]

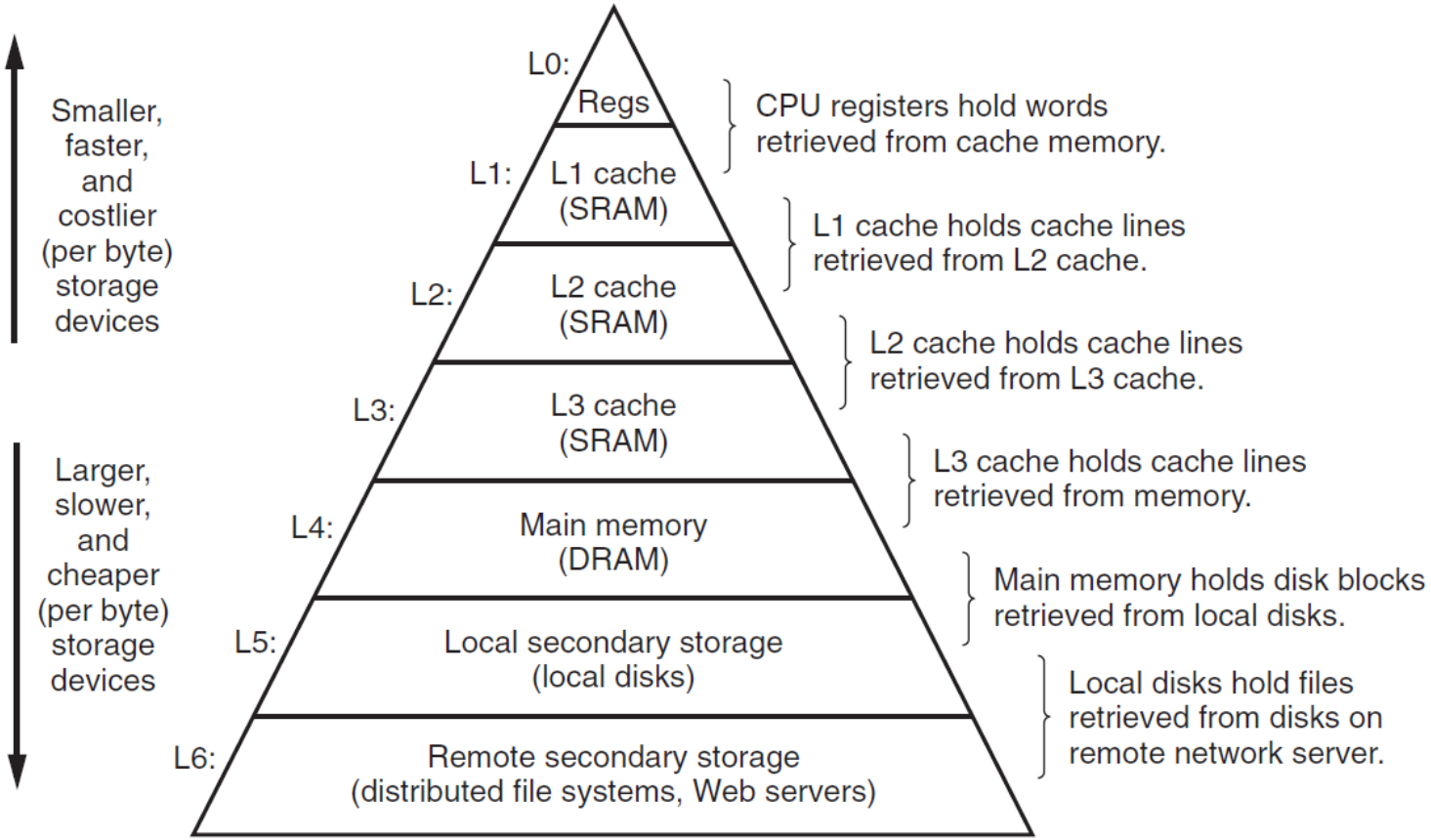


Figure 1.9 **An example of a memory hierarchy.**

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

Programming Languages

- A programming language is an artificial language to communicate with machines.[12]
- 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.

[12] See `https://en.wikipedia.org/wiki/Programming_language`.

Short History[13]

- 1st generation: machine code.
- 2nd generation: assembly code.
- 3rd generation: high-level programming languages.
  - For example, Java.
- 4th generations.
  - For example, SQL.

[13] 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 and https://www.computerhope.com/history/programming.htm.

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:
      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
```

Assembler

Binary machine language program (for MIPS)

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

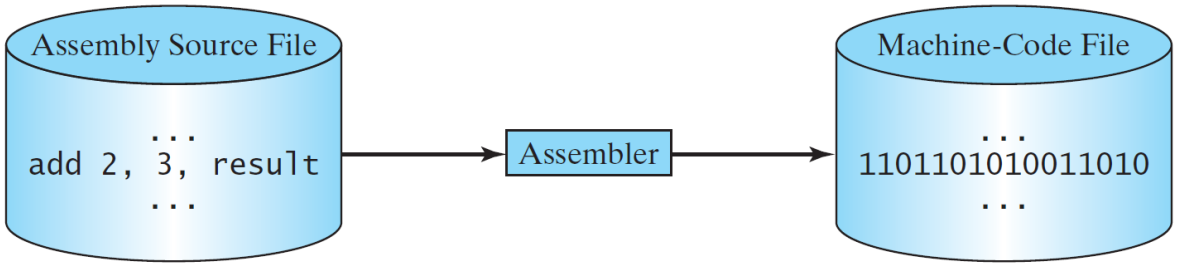
1st-Generation Programming Languages

- The 1st-generation program language is pure machine code, that is just ones and zeros. (Why?)
- Each machine has its own instruction set.[14]
- It means that you may not execute one program on every machine.
- More worse, the machine languages are not human-friendly.

[14] For example, the instruction set of X86 and ARM are incompatible.

2nd-Generation Programming Languages

- An assembly language uses mnemonics to represent instructions as opposed to the machine codes.
- Hence, these codes are easier for human to read and write.
- Assembly code is then converted to machine code.

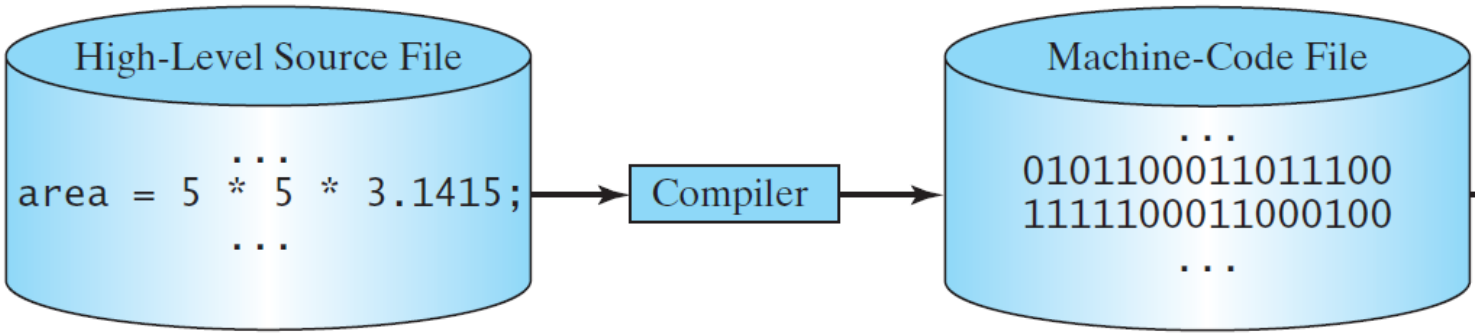


- Note that it presents the execution model very close to the machine.[15]

[15] To be a hacker, you should learn assembly languages.

3rd-Generation Programming Languages

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

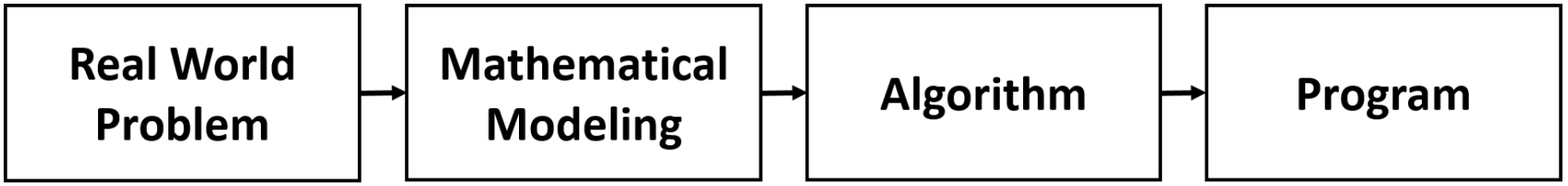


- For example, C[16], C++[17], and Java[18].

[16] Dennis Ritchie (1973).
[17] Bjarne Stroustrup (1983).
[18] James Gosling (1995).

What Can A Program Do?

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



Algorithms In A Nutshell[19]

- An algorithm is a well-defined computational procedure that takes necessary information as input and produces an correct answer as output.
- Simply put, an algorithm is a procedure that solves a specific class of problems, like a recipe or a cookbook.

[19] Also see `http://ed.ted.com/lessons/your-brain-can-solve-algorithms-david-j-malan`.

- An algorithm has properties as follows:
  - Definiteness: all steps are precisely defined.
  - Finiteness: for any input, the algorithm must terminate after a finite number of steps (time).
  - Effectiveness: operations are basic enough (e.g. $+ - \times \div$) to be able to done exactly and in a finite number of steps.
- Note that an algorithm could be expressed not only in programming languages, but also in human languages, flow charts, and pseudo codes.

Example: Greatest Number

- Let A be a list of numbers.
- For example, consider $A = \{1, 7, 9, -2, 4\}$.
- Then it is clear that the answer is 9.
- Now propose an algorithm which finds the greatest element in for any list of numbers.

Input: A.
Output: the greatest element in A.

- Try a top-down approach in your native language?

Optimal Solution

- Let $A(1)$ be the first element of A and so on.
- The symbol $\leftarrow$ is a copy operator from right to left.

```
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
```

- In Line 1, why not $\boxed{\text{max} \leftarrow 0}$ but $\boxed{\text{max} \leftarrow A(1)}$?
- You may extend this solution to more questions:
  - Smallest element?
  - Location of the greatest element?

"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[20], which can be considered a model of a general-purpose computer.
- Proposed the famous question: "*Can machines think?*"[21]
  - Well-known as the Turing test.
- Turing Award is recognized as the highest distinction in computer science and the "Nobel Prize of computing".[22]

[20] Turing (1936). Try `http://www.google.com/doodles/alan-turings-100th-birthday`.

[21] Turing (1950). You could find the paper here: `https://phil415.pbworks.com/f/TuringComputing.pdf`. Also see `https://en.wikipedia.org/wiki/Turing_test`.

[22] See `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.[23]

[23] See https://www.nytimes.com/2019/07/15/business/alan-turing-50-pound-note.html.

About Java

- Java is one of general-purpose programming languages, supporting object-oriented programming (OOP).
- The first version of the Java platform was released by Sun Microsystems in 1995, now owned by Oracle Corporation from 2010.
- It is intended to let application developers write once, run anywhere (WORA).

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]

---

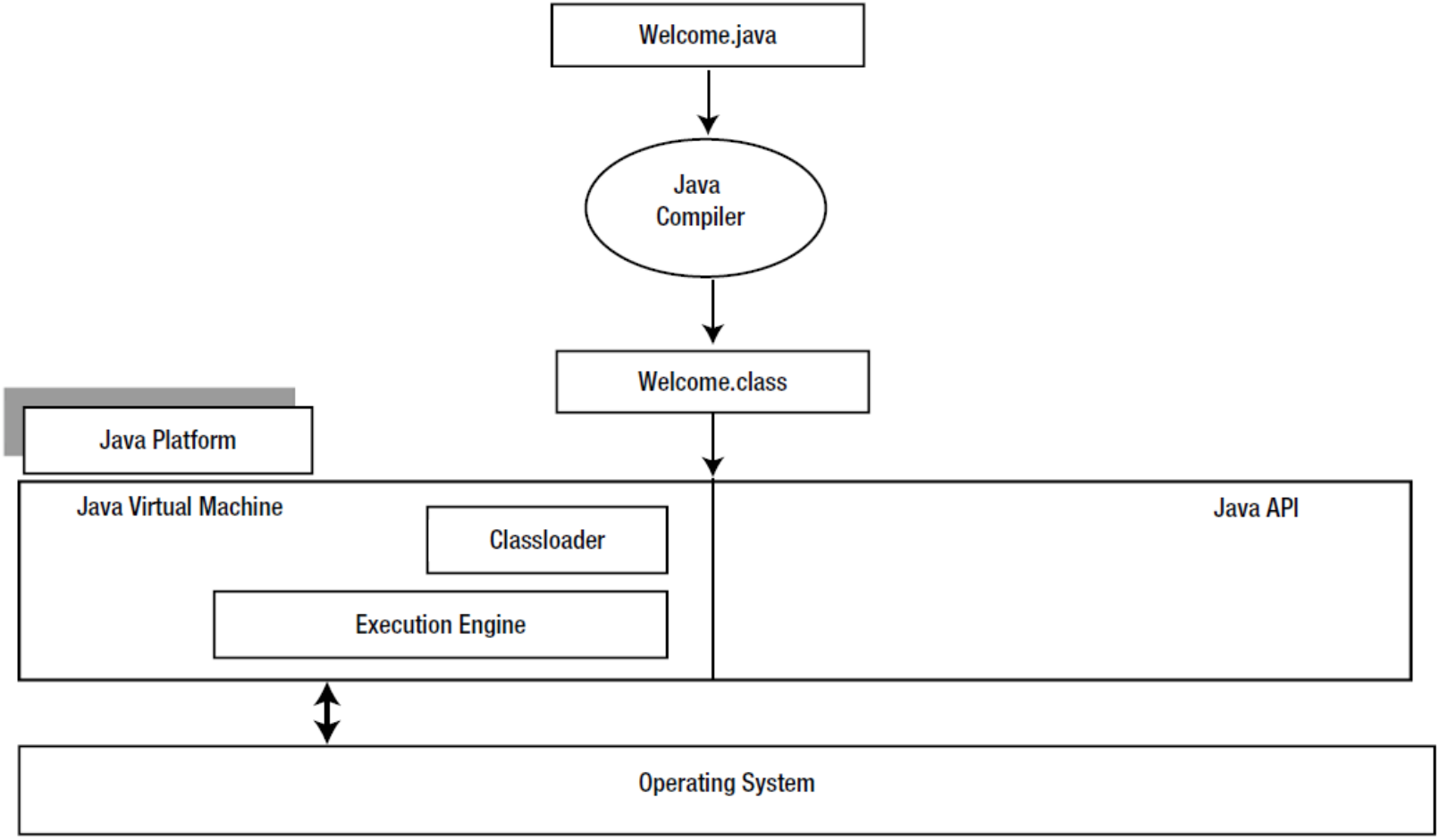
[24] For example, Windows, Linux, MacOS, Android, iOS, et cetera.

[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.

Software Installation

- First, we need Java Development Kit 11 (JDK11).[29]
- Second, we need an integrated development environment (IDE) which provides comprehensive facilities, say code completion, debugger, and build automation tools.
  - We use Eclipse in this course.[30]
  - You may try other IDEs, for example, NetBeans, IntelliJ IDEA, or Visual Studio Code with proper packages.

[29] Try `https://www.oracle.com/java/technologies/javase-jdk11-downloads.html`.

[30] Try `https://www.eclipse.org/downloads/`.

First Program: Hello, World[31]

```java
public class HelloJavaDemo {

    public static void main(String[] args) {

        // Print "Hello, Java." on the screen.
        System.out.println("Hello, Java.");

    }

}
```

- class: declare a new class followed a distinct class name.
- public: can be accessed by any other class.
- static: can be called without having any object.
- void: do not return a value.

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

- A class is an entity of Java programs.
- The class may have a special method[32] called main() used as the entry point of the program.
- **System**.out refers to the standard output device, say the screen.
- The method println() is used to output a string to the screen.
- Every statement ends with a semicolon (;).

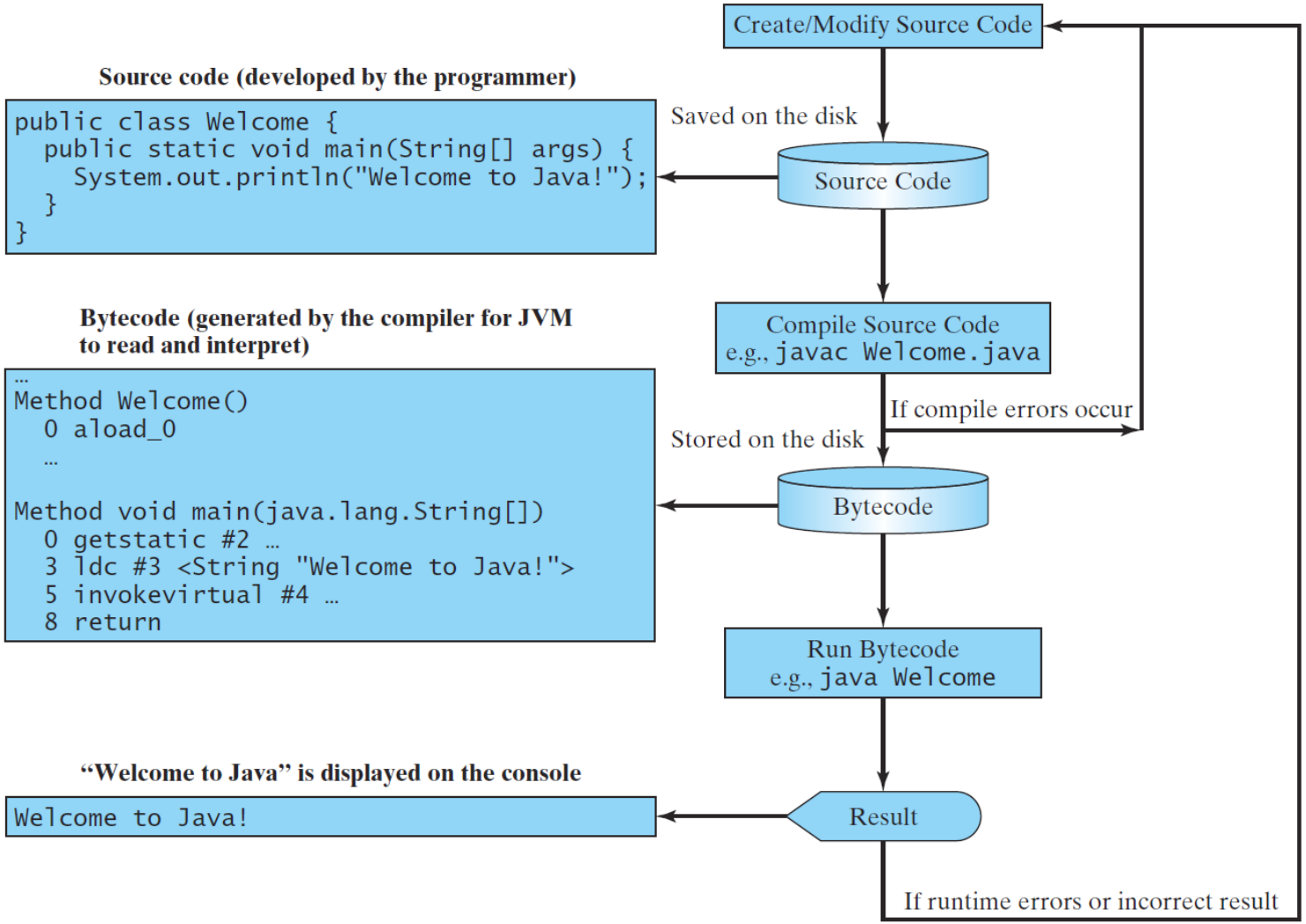
[32] Aka functions and subroutines.

Public Classes

- The keyword public is one of access modifiers[33], allowing the programmer to control the visibility of classes and also members.
- The name of public class is identical to the file name.
- There must be at most one public class in one file.

[33] We will visit the access controls later.

How To Run A Java Program[34]



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

Table of Special Characters

Symbol	Name	Description
{ }	Opening/closing braces	Denote a block to enclose statements.
( )	Opening/closing parentheses	Mostly used with methods.
[ ]	Opening/closing brackets	Denote an array.
//	Double slashes	Precede a comment line.
" "	Opening/closing quotation marks	Enclose a string.
;	Semicolon	Mark the end of a statement.

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 because 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)

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

– Alfred Pennyworth, *Batman Begins* (2005)

Programming Style

- Good programming style makes a program easy to read and helps programmers prevent from errors.
  - For example, Google Java Style.
- In particular, we use indentation to enhance the structural relationships by visual.
- Be consistent through the whole program!