



# Lecture 1

Introduction to Computers, Programs, and Java



- Learn to program in Java
- Acquire the basics of object oriented programming
- Use modern software engineering techniques



- Weekly lectures
  - Tuesdays 2:00 - 3:15pm – CIWW room 317
  - Thursdays 2:00 - 3:15pm – CIWW room 317
- Office hours:
  - Thursdays 3:30 - 4:30pm – 60 Fifth Ave - 5th floor
- Tutors:
  - 60 Washington Square South, 4th Floor (Kimmel Center)
- Final exam
  - Thursday, May 14, 2020 – 2:00 - 3:50pm
- Book

**Introduction to JAVA programming**

Brief version, 11th edition

Y. Daniel Liang

ISBN: 0-13-359220-0



- Part 1: Fundamentals of Programming
  - Command line
  - Types
  - Control flow
  - Looping
  - Input output
  - Strings
  - Functions
  - Arrays
- Midterm 1



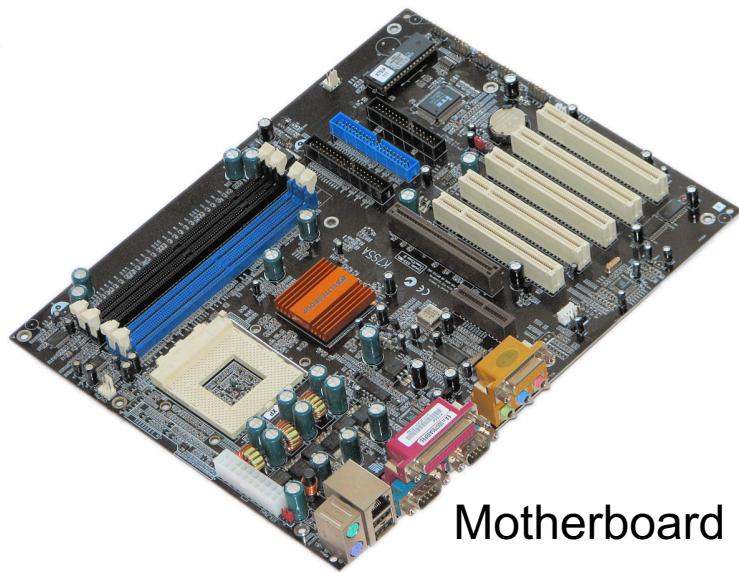
- Part 2: Object oriented programming and design
  - Objects and classes
  - Object oriented design
- Midterm 2
- Part 3: Advanced concepts
  - Inheritance
  - Polymorphism
  - Exception handling
  - Abstract classes and interfaces
  - (Recursion)
- Final exam



- Assignments and quizzes 10%
  - Late assignments count 70%
- Project 10%
- In class participation 5%
- Midterms 20% each
- Final 35%



- You are encouraged to consult with your classmates/friends but collaboration in the assignments is **NOT** allowed.
- You are **NOT** allowed to copy code online or use external libraries (except those provided in the class).
- We will use plagiarism tools to validate all homework. Plagiarism will be punished with a zero-tolerance policy: **the minimal penalty will be a score of 0 points on the assignment, the reduction of 1 letter grade in the final score, and a permanent ban on NYU-related jobs.**
- If you are stuck in an assignment:
  - Google
  - Ask the tutors
  - Come to the office hours



Motherboard



CPU



Memory (RAM)



Hard disk



- The central processing unit (CPU) is the brain of a computer.
- It retrieves instructions from memory and executes them.
- The CPU speed is measured in gigahertz (GHz)
  - 1 megahertz equaling 1 million millions pulses per second.
- The speed of the CPU has been improved continuously.
- If you buy a PC now, you can get an Intel i9 Processor at 4.80GHz





- *Memory* is to store data and program instructions for CPU to execute.
- A memory unit is an ordered sequence of bytes, each holds eight bits.
- A program and its data must be brought to memory before they can be executed.
- A memory byte is never empty, but its initial content may be meaningless to your program.
- The current content of a memory byte is lost whenever new information is placed in it.





- Data of various kinds, such as numbers, characters, and strings, are encoded as a series of bits (zeros and ones).
- Computers use zeros and ones because digital devices have two stable states, which are referred to as zero and one by convention.
- The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme.
- No two data can share or split a same byte.
- A byte is the minimum storage unit.



- Memory is volatile, because information is lost when the power is off.
- Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them.





- Computer programs, known as software, are instructions to the computer.
- You tell a computer what to do through programs.
- Without programs, a computer is an empty machine.
- Computers do not understand human languages, so you need to use computer languages to communicate with them.
- Programs are written using programming languages.



- Machine language is a set of primitive instructions built into every computer.
- The instructions are in the form of binary code, so you have to enter binary codes for various instructions.
- Program with native machine language is a tedious process.
- Moreover the programs are highly difficult to read and modify.
- Example:
  - to add two numbers, you might write an instruction in binary like this:
  - 1101101010011010



- Assembly languages were developed to make programming easy.
- Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code.
- Example:
  - to add two numbers, you might write an instruction in assembly code like this:
  - ADDF3 R1, R2, R3





- The high-level languages are English-like and easy to learn and program.
- Example:
  - compute the area of a circle with radius 5:
  - `area = 5 * 5 * 3.1415`



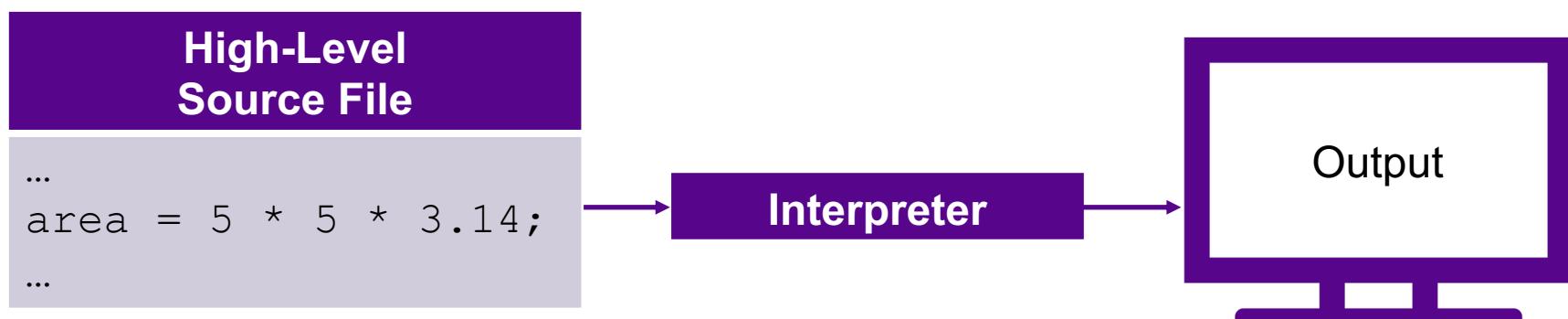
Language	Description
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced “C Sharp”. It is a hybrid of Java and C++ and was developed by Microsoft.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Python	A simple general-purpose scripting language good for writing short programs.
JavaScript	A high-level, interpreted programming language often used for the Web. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.
Swift	A general-purpose, multi-paradigm, compiled programming language developed by Apple for iOS, macOS, watchOS, tvOS, and Linux.



- A program written in a high-level language is called a source program or **source code**.
- Because a computer cannot understand a source program, a source program must be translated into machine code for execution.
- The translation can be done using another programming tool called an interpreter or a compiler.

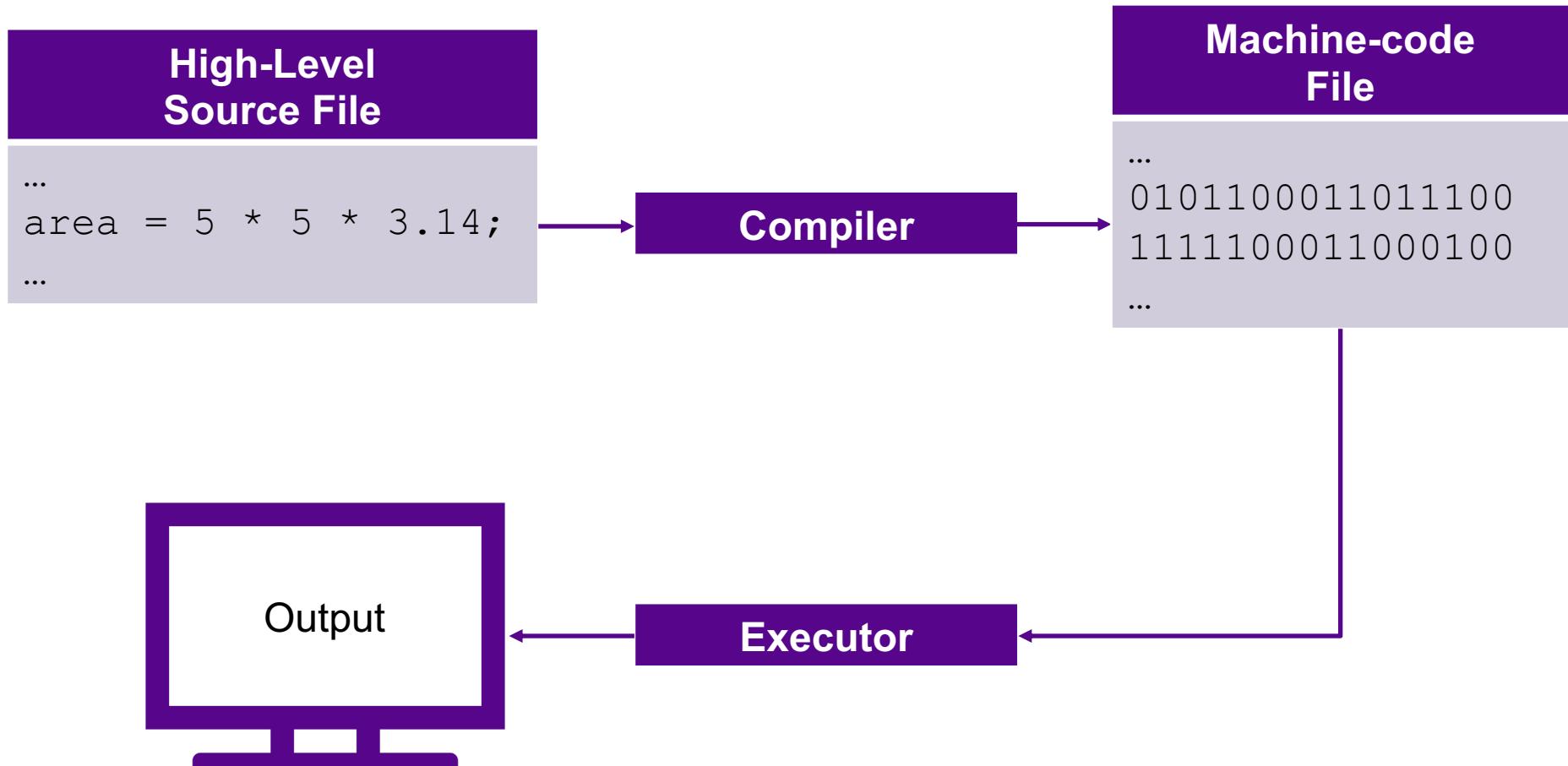


- An interpreter
  - reads one statement from the source code,
  - translates it to the machine code or virtual machine code,
  - then executes it right away.
- Note that a statement from the source code may be translated into several machine instructions.





- A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed





- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

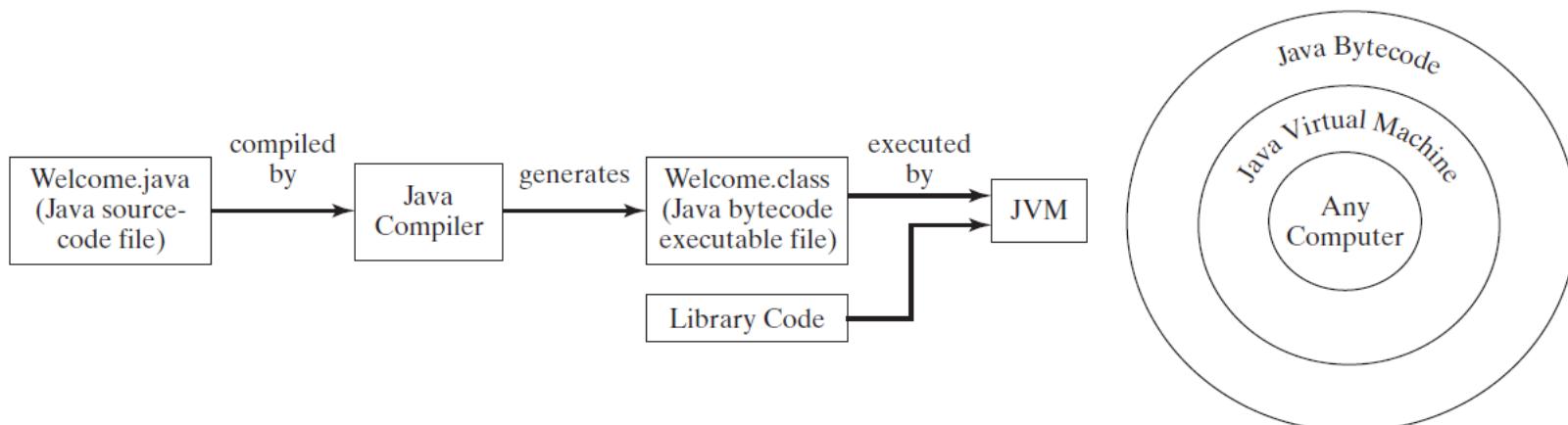


```
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

```
##### python
if __name__ == "__main__":
    print("Welcome to Java!")
```



- You can port a source program to any machine with appropriate compilers.
- The source program must be recompiled, however, because the object program can only run on a specific machine.
- Java was designed to run object programs on any platform.
- With Java, you write the program once, and compile the source program into a special type of object code, known as bytecode.
- The bytecode can then run on any computer with a Java Virtual Machine.
- Java Virtual Machine is a software that interprets Java bytecode.





<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

The screenshot shows the Oracle Java SE Downloads page. At the top, there's a navigation bar with the Oracle logo, a menu icon, a search bar, sign-in options, and a country/region selector. Below the navigation is a breadcrumb trail: Oracle Technology Network / Java / Java SE / Downloads.

The main content area has tabs for Overview, Downloads (which is selected), Documentation, Community, Technologies, and Training. On the left, a sidebar lists categories like Java SE, Java EE, Java ME, etc. The central part features two download cards: "Java Platform (JDK 9)" and "NetBeans with JDK 8".

A large section titled "Java Platform, Standard Edition" contains a "Java SE 9.0.4" update notice and a list of links: Installation Instructions, Release Notes, Oracle License, Java SE Licensing Information User Manual (which includes a note about third-party licenses), Certified System Configurations, and Readme.

To the right, under "Java SDKs and Tools", are links for Java SE, Java EE and Glassfish, Java ME, Java Card, NetBeans IDE, Java Mission Control, Java Resources, Java APIs, Technical Articles, Demos and Videos, Forums, Java Magazine, Developer Training, Tutorials, and Java.com. The "JDK DOWNLOAD" button is highlighted with a purple rectangle.

At the bottom, a section titled "Which Java package do I need?" provides guidance for different user roles: Software Developers (JDK), Administrators running applications on a server (Server JRE), and End users running Java on a desktop (JRE). It also includes a note about browser integration for the Java plug-in.

- Set path to JDK bin directory
- Set classpath to include the current directory

- Compile
  - javac Welcome.java



```
teseo@216-165-71-208 ~/java $ javac Welcome.java
```

A screenshot of a Mac OS X Terminal window titled "Terminal — 49x5". The window shows a command-line session where the user has just run the "javac" command on a file named "Welcome.java". The output is displayed in green text.

- Run
  - java Welcome



```
teseo@216-165-71-208 ~/java $ java Welcome
Welcome to Java!
teseo@216-165-71-208 ~/java $
```

A screenshot of a Mac OS X Terminal window titled "Terminal — 49x5". The window shows a command-line session where the user has run the "java" command on the "Welcome" class. The program's output, "Welcome to Java!", is displayed in green text.



Mac	Windows
mkdir	mk
cd <folder>	cd <folder>
rm <file>	erase <file>
ls	dir
ls   grep	dir   findstr -i
cat	type
cp <file> <out>	copy <file> <out>
mv <file> <out>	rename <file> <out>
javac	javac
java	java
pwd	