Introduction

Java Programming

Byeongjoon Noh

powernoh@sch.ac.kr

SOON CHUN HYANG UNIVERSITY

Most of the slides are available on Senseable AI Lab homepage: https://sailab.space/courses/

Contents

1. What is a programming?
2. Computing thinking
3. Introduction on OOP in Java
4. (Practice class) Development environment setting
5. Basic data representation in computer

1. What is a programming?

Program

- A sequence of instruction that specifies how to perform a computation
  - computation
    - something mathematical
      - e.g., solving equations or finding the roots of a polynomial
    - can also be symbolic computation
      - e.g., searching and replacing text in a document or something graphical
        - processing an image or playing a video

Concept of programming language

- Programming
  - to command the computer to do what a human thinks
  - all the work to create a program and also referred to as “development”

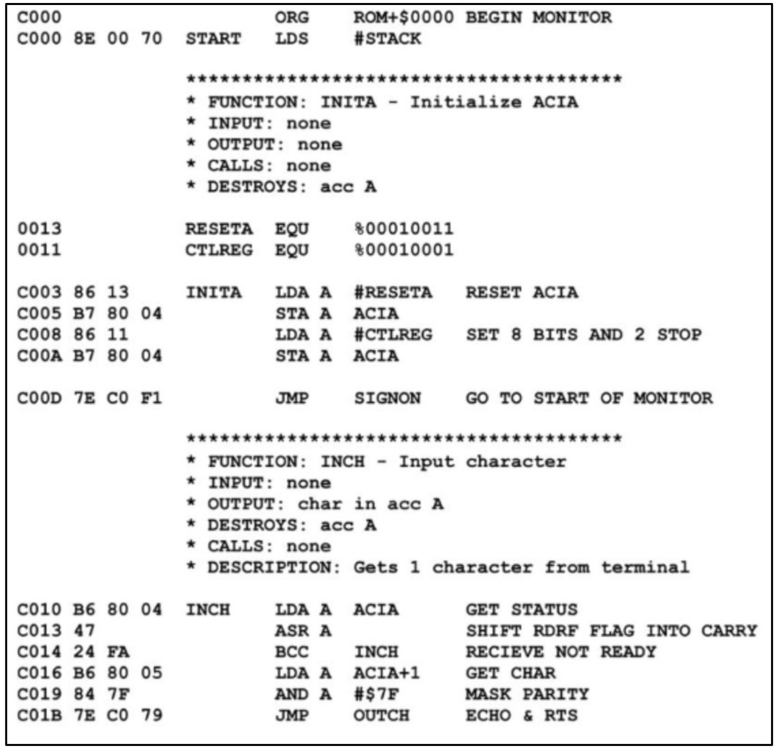
- Programming language
  - a tool to create software (e.g. Excel, League of Legend, etc.) that operates on a computer, using a language the computer can understand
  - “컴퓨터가 실행할 프로그램을 작성하기 위한 언어”

- Programmer
  - a person who uses programming language to create software or apps (applications)

Programming language processing

- Machine language (machine code)
  - only binary commands; can be processed by computer
  - example
    - 1010111 11100111 11101010 0001101 10101011 01111001 10101101 …

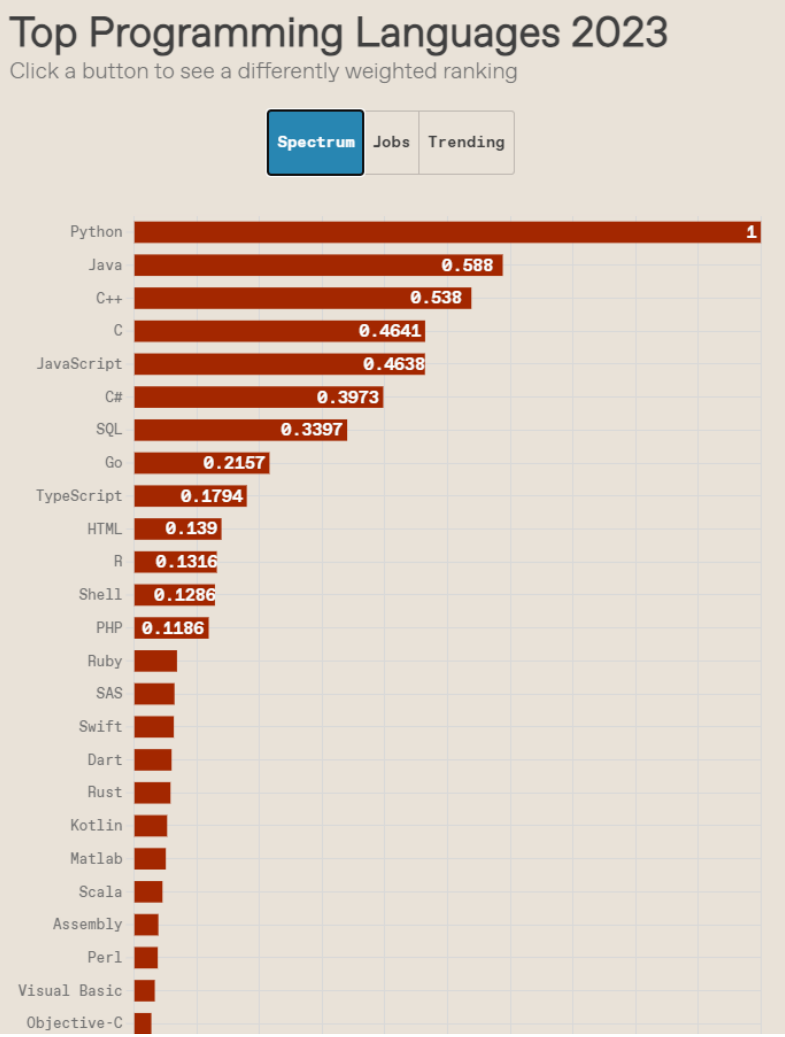
- Assembly code
  - use “Mnemonic” (니모닉) symbols



```
C000                  ORG    ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START  LDS    #STACK

               ****************************************
               * FUNCTION: INITA - Initialize ACIA
               * INPUT: none
               * OUTPUT: none
               * CALLS: none
               * DESTROYS: acc A

0013           RESETA EQU    %00010011
0011           CTLREG EQU    %00010001

C003 86 13     INITA  LDA A  #RESETA   RESET ACIA
C005 B7 80 04         STA A  ACIA
C008 86 11            LDA A  #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04         STA A  ACIA

C00D 7E C0 F1         JMP    SIGNON    GO TO START OF MONITOR

               ****************************************
               * FUNCTION: INCH - Input character
               * INPUT: none
               * OUTPUT: char in acc A
               * DESTROYS: acc A
               * CALLS: none
               * DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04  INCH   LDA A  ACIA      GET STATUS
C013 47               ASR A            SHIFT RDRF FLAG INTO CARRY
C014 24 FA            BCC    INCH      RECIEVE NOT READY
C016 B6 80 05         LDA A  ACIA+1    GET CHAR
C019 84 7F            AND A  #$7F      MASK PARITY
C01B 7E C0 79         JMP    OUTCH     ECHO & RTS
```

Programming language processing

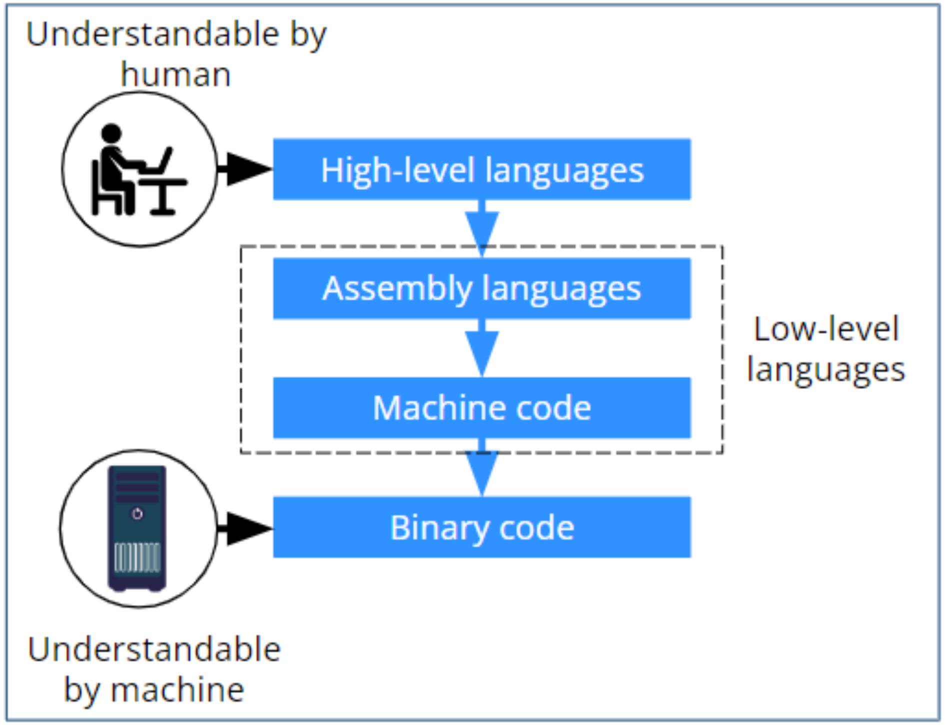
- High-level languages
  - designed to be easily understood by human
  - example
    - X = (2 * X + Z – Y) / 5
  - various programming languages
    - Java, C, C++, Python, etc.



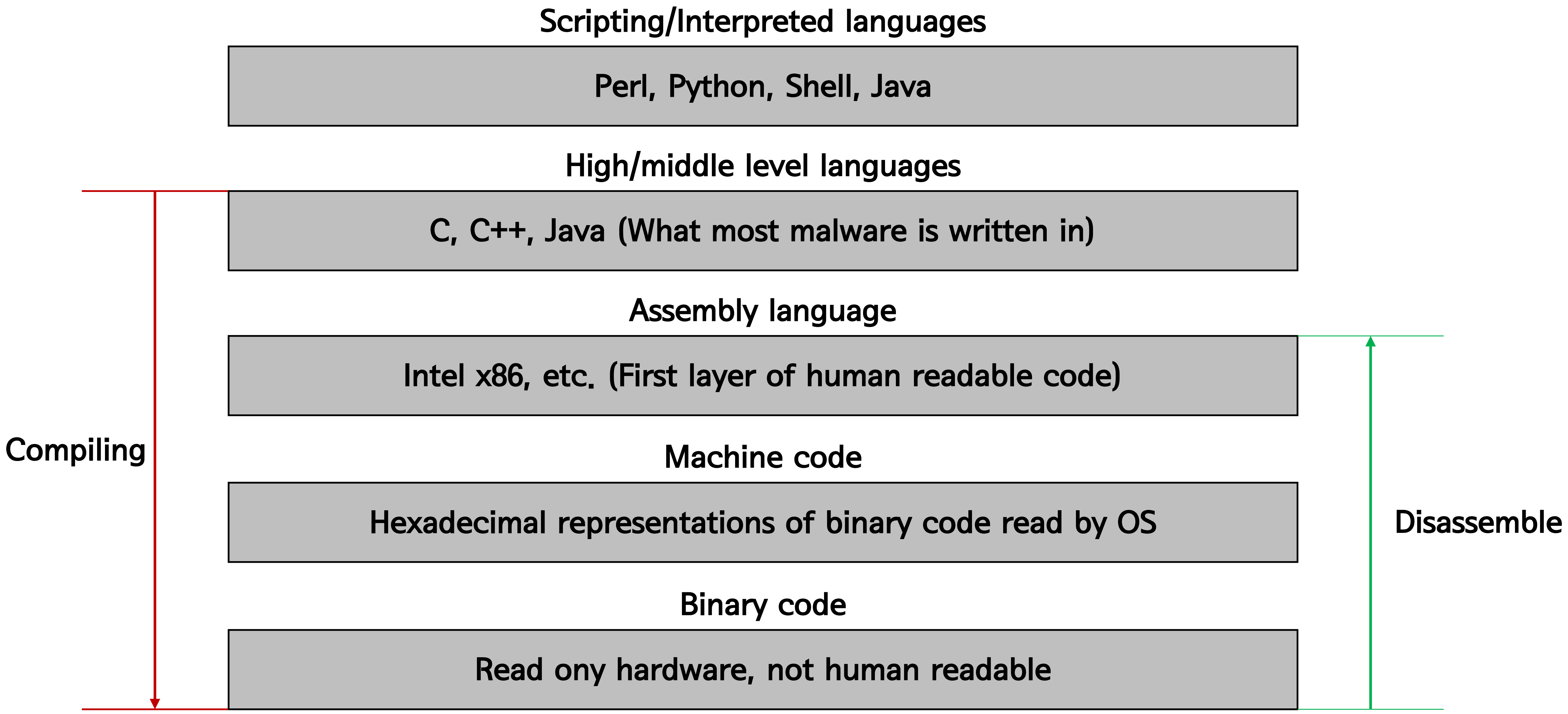
IEEE Spectrum: The Top Programming Languages 2023

Programming language processing

- Compile
  - converting programming language into machine code that can be executed by computer
- Compiler
  - a program that converts to machine language (e.g., Java compiler, gcc, etc.).



Programming language processing



Scripting/Interpreted languages

Perl, Python, Shell, Java

High/middle level languages

C, C++, Java (What most malware is written in)

Assembly language

Intel x86, etc. (First layer of human readable code)

Machine code

Hexadecimal representations of binary code read by OS

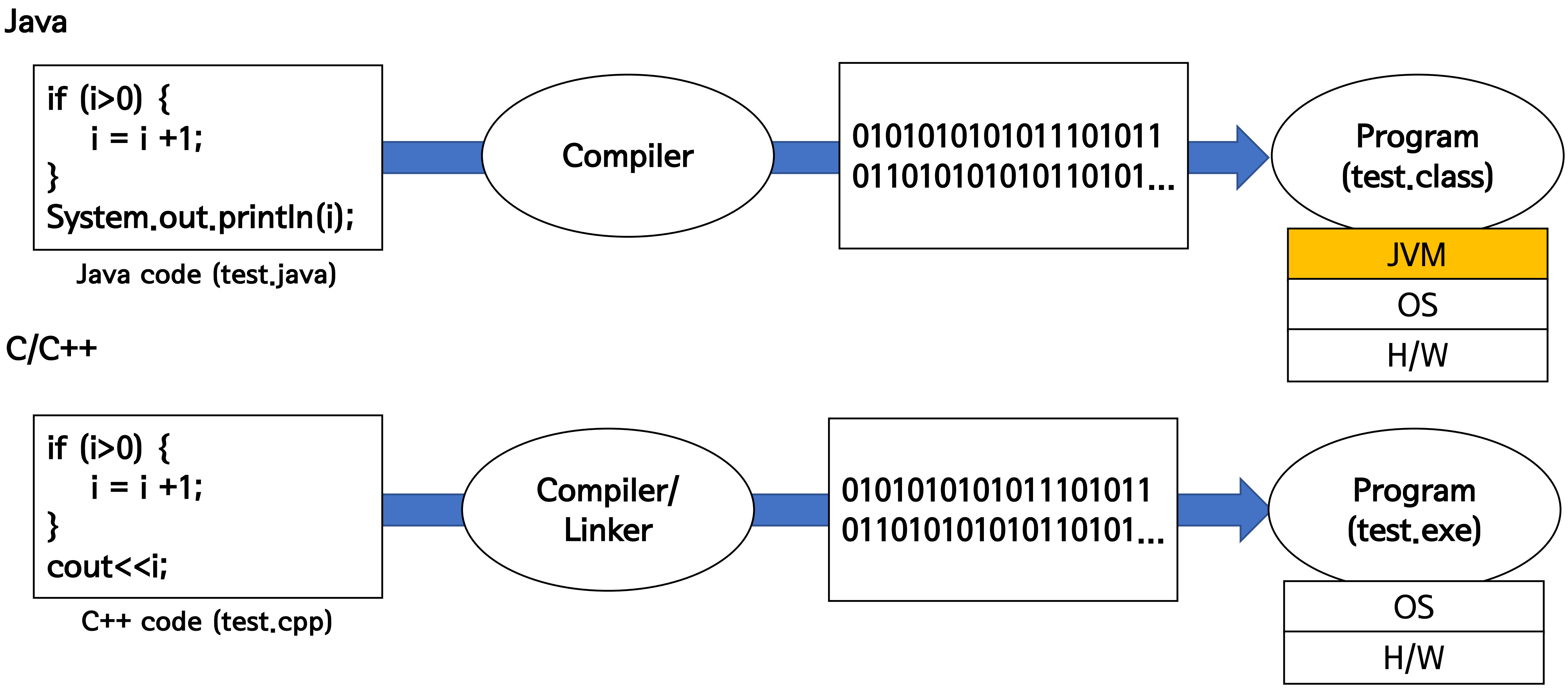
Binary code

Read ony hardware, not human readable

Compiling

Disassemble

Programming language processing



Java

```
if (i>0) {
    i = i +1;
}
System.out.println(i);
```

Java code (test.java)

C/C++

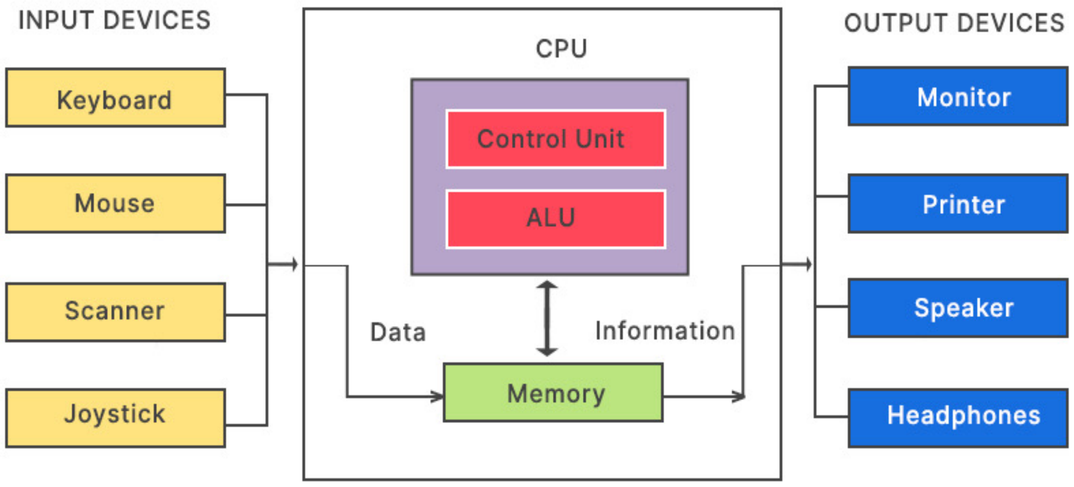
```
if (i>0) {
    i = i +1;
}
cout<<i;
```

C++ code (test.cpp)

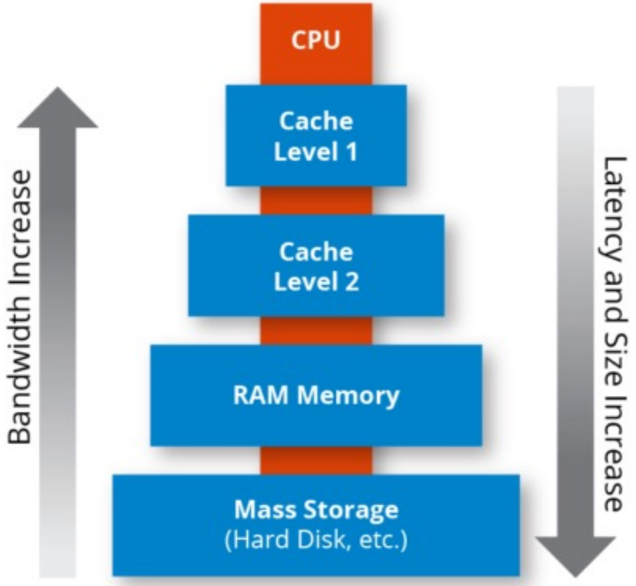
Basic introduction of programming languages

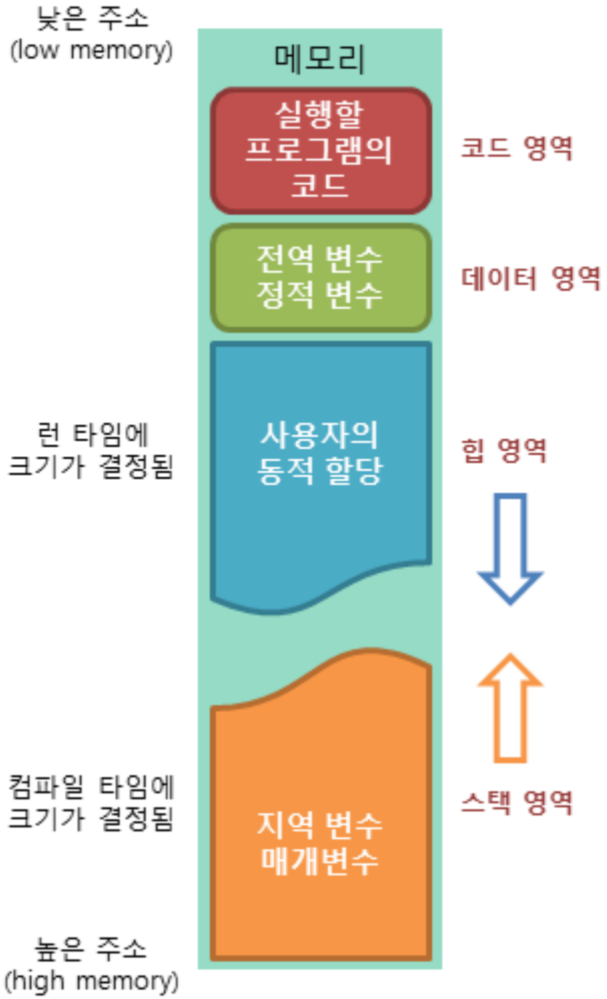
- The details looks different in different languages, but a few basic instructions appear in just about every language
    - input: Get data from the keyboard, a file, the network, or some other device
    - output: Display data on the screen, save it in a file, send it over the network, etc.
    - math: Perform basic mathematical operations like addition and multiplication
    - conditional execution: Check for certain conditions and run the appropriate code
    - repetition: Perform some actions repeatedly, usually with some variations

Note: common knowledge in computer



Basic Architecture Of a Computer





2. Computing thinking

Computing thinking

- A problem-solving process
  - involving a set of skills and techniques computer scientists use to approach problems in a way that could be implemented with a computer
  - approach to solving problem; not just about programming

- Some key practices
  - decomposition, pattern recognition, abstraction, algorithm design, etc.

Key practices

- Decompositions
  - breaking down complex problems into more manageable parts
- Pattern recognition
  - looking for similarities, or patterns, in and among problems
- Abstraction
  - focusing on the important information only, and ignoring irrelevant detail
- Algorithm design
  - developing a step-by-step solution to the problem, or the rules to follow to solve the problem

Real world examples

- Cooking a new recipe:
  - you might break the process down into smaller steps like preparation, cooking, and serving (decomposition)
  - notices patterns in cooking times or ingredient combinations (pattern recognition)
  - focus on the critical steps that will affect the dish's outcome (abstraction)
  - follow a specific set of cooking instruction (algorithm design)

- Healthcare – increasing muscle definition and reducing body fat

In programming

- Problem: Determine the given number is a prime number or not
    - decomposition:
        - understanding what a prime number is
            - a natural number greater than 1 that has no positive divisor other than 1 and itself
        - identifying the steps needed to determine if a number is prime
    - pattern recognition
        - recognizing that no prime number greater than 2 is even, so can skip all even numbers (except 2) when searching for primes
    - abstraction
        - organizing functions; takes a number as input and returns whether the number is prime

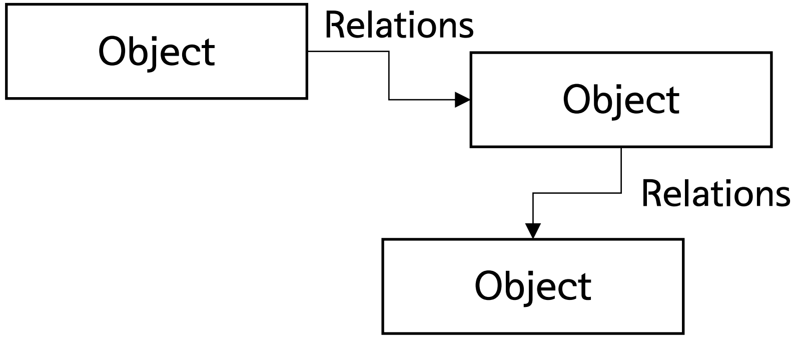
In programming

- Problem: Determine the given number is a prime number or not
    - algorithm design
        - 1. given number = ‘n’
        - 2. if ‘n’ is 2 ➔ prime number
        - 3. if ‘n’ is even ➔ no prime number
        - 4. check divisibility from 3 to ‘n’
            - if any number divides ‘n’ without a remainder ➔ no prime number
        - 5. no divisor are found ➔ ‘n’ is prime number

3. Introduction on OOP in Java

Concept of OOP

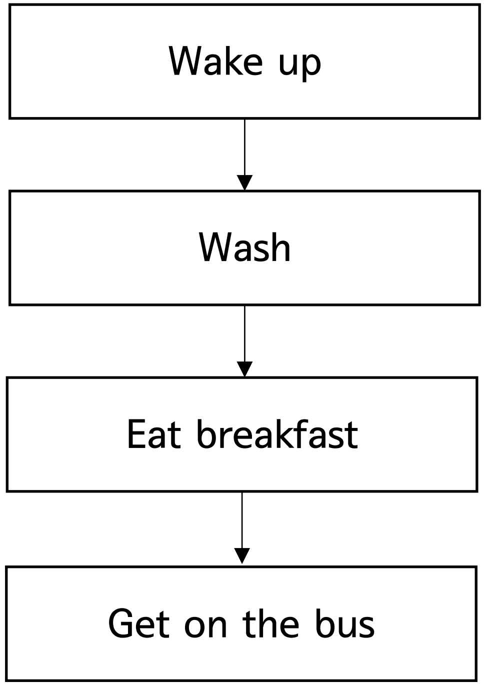
- Object-oriented programming (OOP)
  - A programming paradigm based on concept of "objects"
  - Groups data and behaviors as objects



- What is an object?
  - A thing, both tangible and intangible
  - e.g., account, student, vehicle, product, delivery, order, lecture, smart phone, singer, etc.

Concept of procedural programming

- Procedural programming
  - based on the concept of procedure
  - programs are divided into procedures; as routines or functions
  - diagram examples for the procedural programming

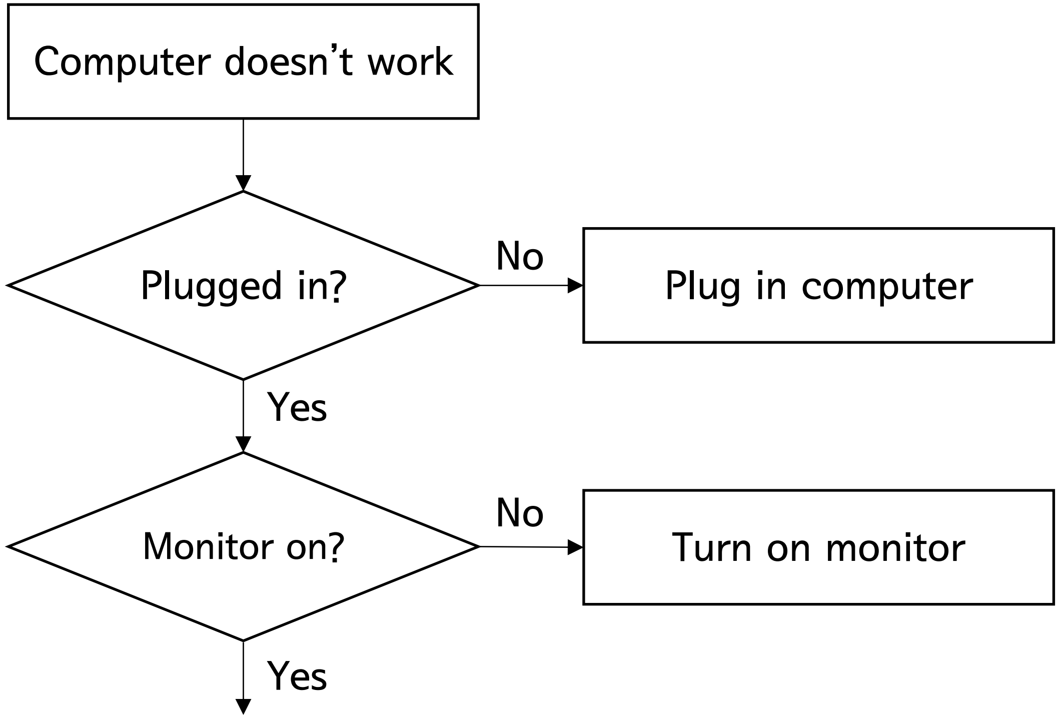


Wake up

Wash

Eat breakfast

Get on the bus



Computer doesn’t work

Plugged in?

No

Plug in computer

Yes

Monitor on?

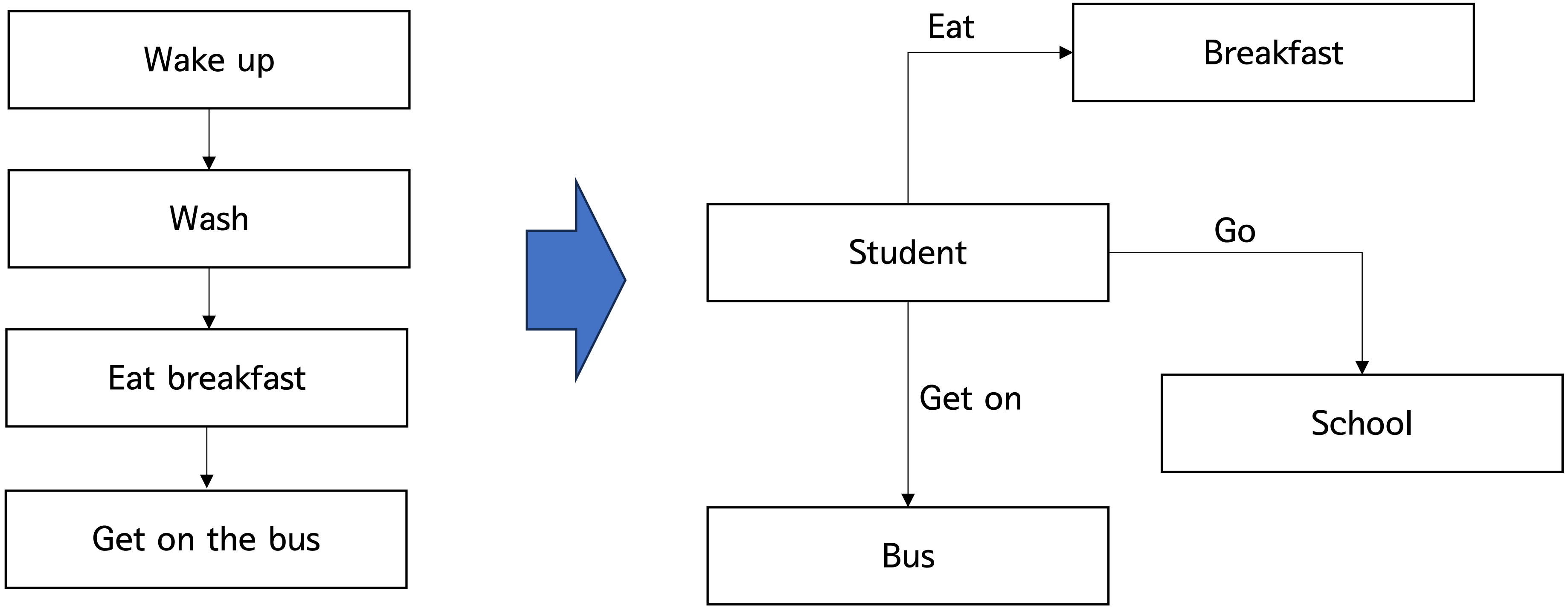
No

Turn on monitor

Yes

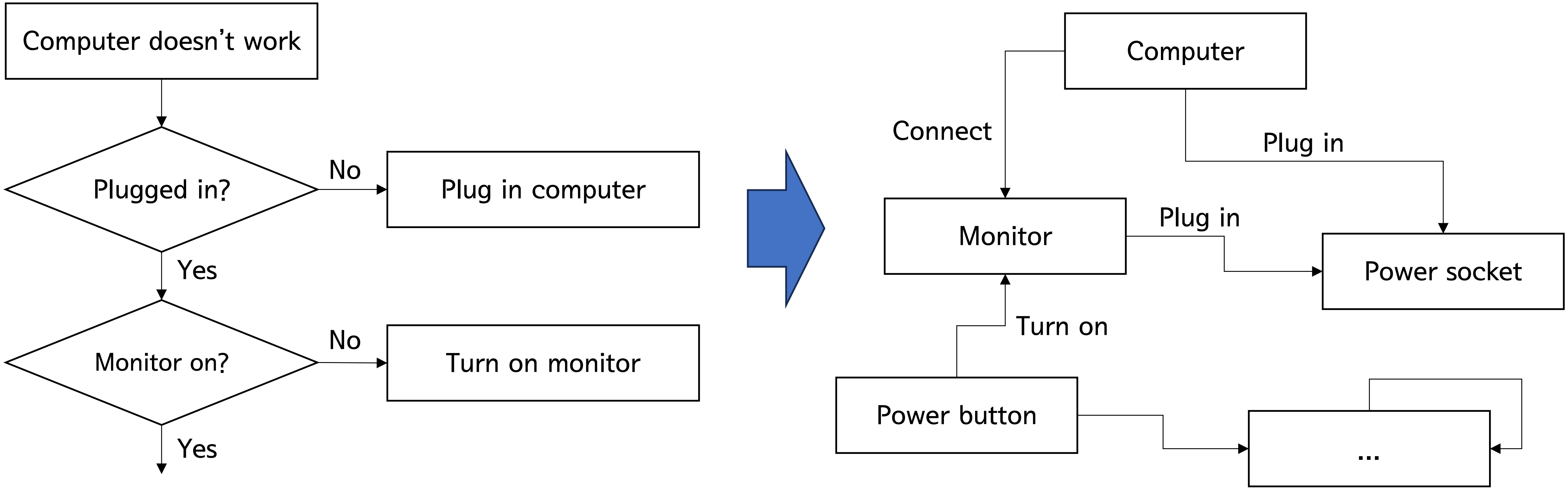
OOP vs. procedural programming

- OOP diagrams



OOP vs. procedural programming

- OOP diagrams



How to develop a program with OOP paradigm

- How to develop OOP?
  - Define objects (after understanding problems and requirements)
  - Design behaviors (methods/functions) for each object or relationships between them

- Example: Purchasing a product in an online shopping mall
  - 1) log in as a member to an online shopping mall
  - 2) search for product by navigating the website
  - 3) select the product and add cart
  - 4) checkout
  - 5) enter shipping information and payment method

How to develop a program with OOP paradigm

- Example: Taking coffee in Starbucks
  - 1) enter Starbucks
  - 2) choose your coffee
  - 3) place your order
  - 4) make payment
  - 5) wait
  - 6) pick up your order
  - 7) enjoy your coffee

Why Java?

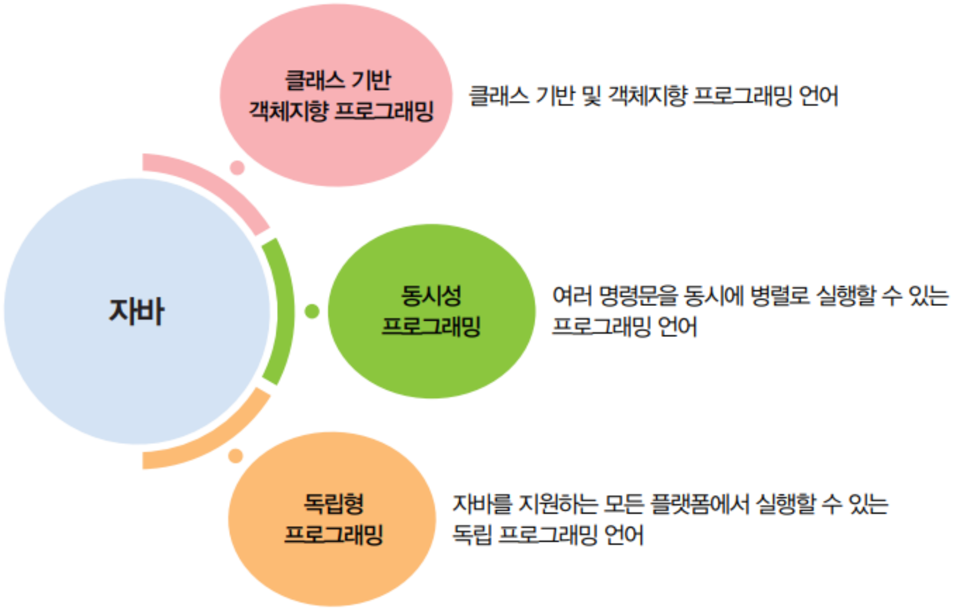
- By James Gosling at Sun Microsystems in 1995

- A high-level, object-oriented programming language
  - inheritance, polymorphism, encapsulation

- 현재 가장 널리 사용되는 프로그래밍 언어 중 하나임

- 문법과 구문은 C, C++과 거의 유사함

- Platform independent
  - Java program can be executable in any platforms



클래스 기반 및 객체지향 프로그래밍 언어

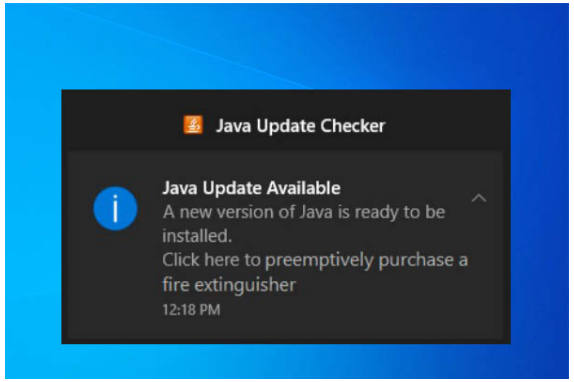
여러 명령문을 동시에 병렬로 실행할 수 있는 프로그래밍 언어

자바를 지원하는 모든 플랫폼에서 실행할 수 있는 독립 프로그래밍 언어

[그림 1-1] 자바의 개요

Java applications

- Java application의 유형
  - 독립 실행형 애플리케이션
    - 별도의 컴퓨터 프로세스에서 실행되는 응용 프로그램
    - e.g., 미디어 플레이어, 백신 프로그램, 그림판, POS 결제 소프트웨어 등
  - 웹 애플리케이션
    - 클라이언트에 의해 실행되는 클라이언트-서버 소프트웨어 응용 프로그램
    - e.g., 이메일, 전자상거래 웹사이트, 은행 웹사이트
  - 엔터프라이즈 애플리케이션
    - 기업 전체에서 소프트웨어 및 하드웨어 시스템과 서비스를 사용함
    - e.g., 전자상거래, 회계, 은행 정보 시스템 등



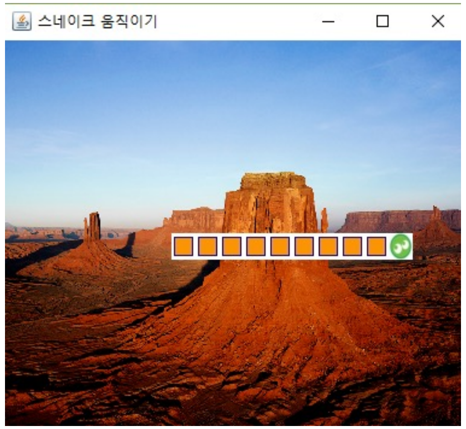
Java Update Checker

Java Update Available

A new version of Java is ready to be installed.

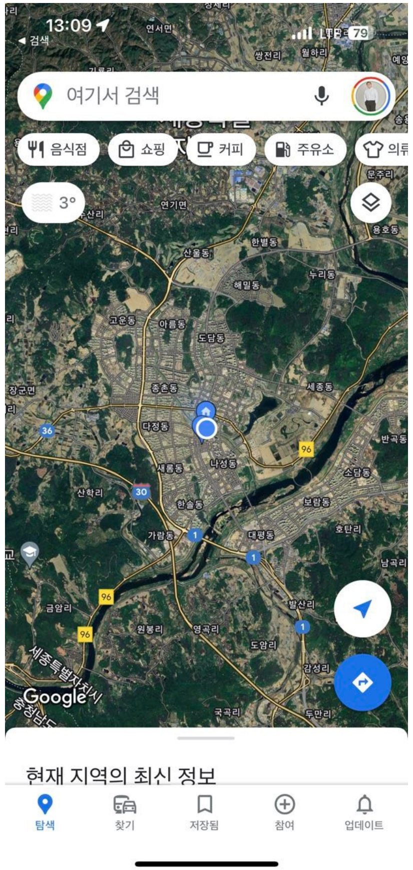
Click here to preemptively purchase a fire extinguisher

12:18 PM



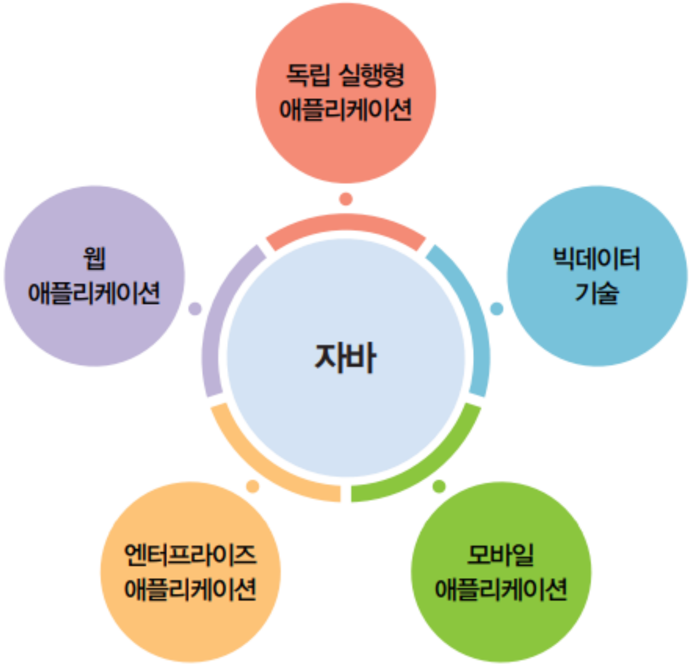
Java applications

- Java application의 유형
  - 모바일 애플리케이션
    - 스마트폰에서 다양한 애플리케이션을 실행하는 플랫폼
    - e.g., Google map 등
  - 빅데이터 기술
    - 대용량 자료를 처리할 수 있는 컴퓨터 클러스터에서 동작하는 분산 애플리케이션 지원
    - e.g., Hadoop (하둡)



Java의 주요 특징

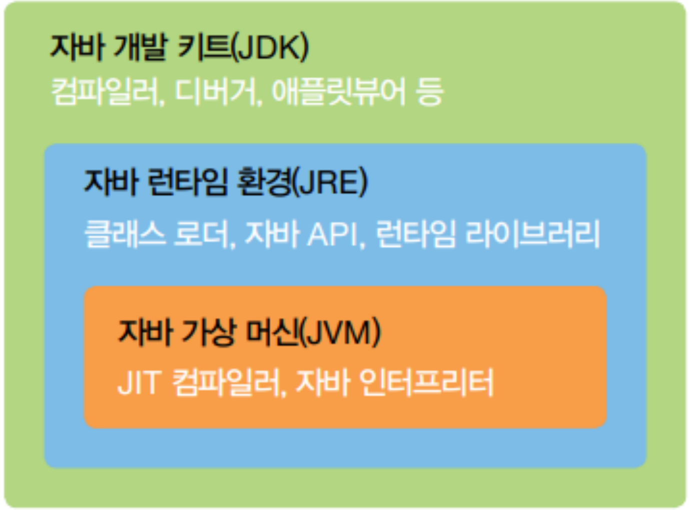
- 단순함
- 객체지향
- 강력한 기능
- 플랫폼 독립적
- 보안
- 멀티스레딩
- 아키텍처 중립적
- 휴대성
- 고성능
- 분산



4. Development environment setting

Java platform

- Java platform 구성 요소
  - Java Development Kit (JDK)
    - Java에서 제공되는 개발용 라이브러리. 계속 버전이 올라가고 있음
  - Java Runtime Environment (JRE)
    - Java 프로그램이 실행되는 환경. 8.0까지 무료로 제공됨
  - Java Virtual Machine (JVM)
    - Java 가상 머신으로 프로그램이 실행되는 환경인 JRE가 설치되어 있어야 함


자바 개발 키트(JDK)
컴파일러, 디버거, 애플릿뷰어 등
자바 런타임 환경(JRE)
클래스 로더, 자바 API, 런타임 라이브러리
자바 가상 머신(JVM)
JIT 컴파일러, 자바 인터프리터

Java platform

- Java Development Kit (JDK)
  - Java application을 만드는데 사용되는 소프트웨어 개발 환경
  - Windows, MacOS, Solaris, Linux 등 다양한 운영체제(OS)에서 JDK 사용을 지원
  - 동일한 컴퓨터에 둘 이상의 JDK 버전을 설치할 수도 있음 (별도의 설정 필요)

  - JDK의 특징
    - JDK에는 Java 프로그램을 작성하는 데 필요한 도구와 이를 실행하는 JRE가 포함됨
    - 컴파일러, Java application 실행기 등이 포함됨
      - * Compiler: Java로 작성된 코드를 byte code로 변환함

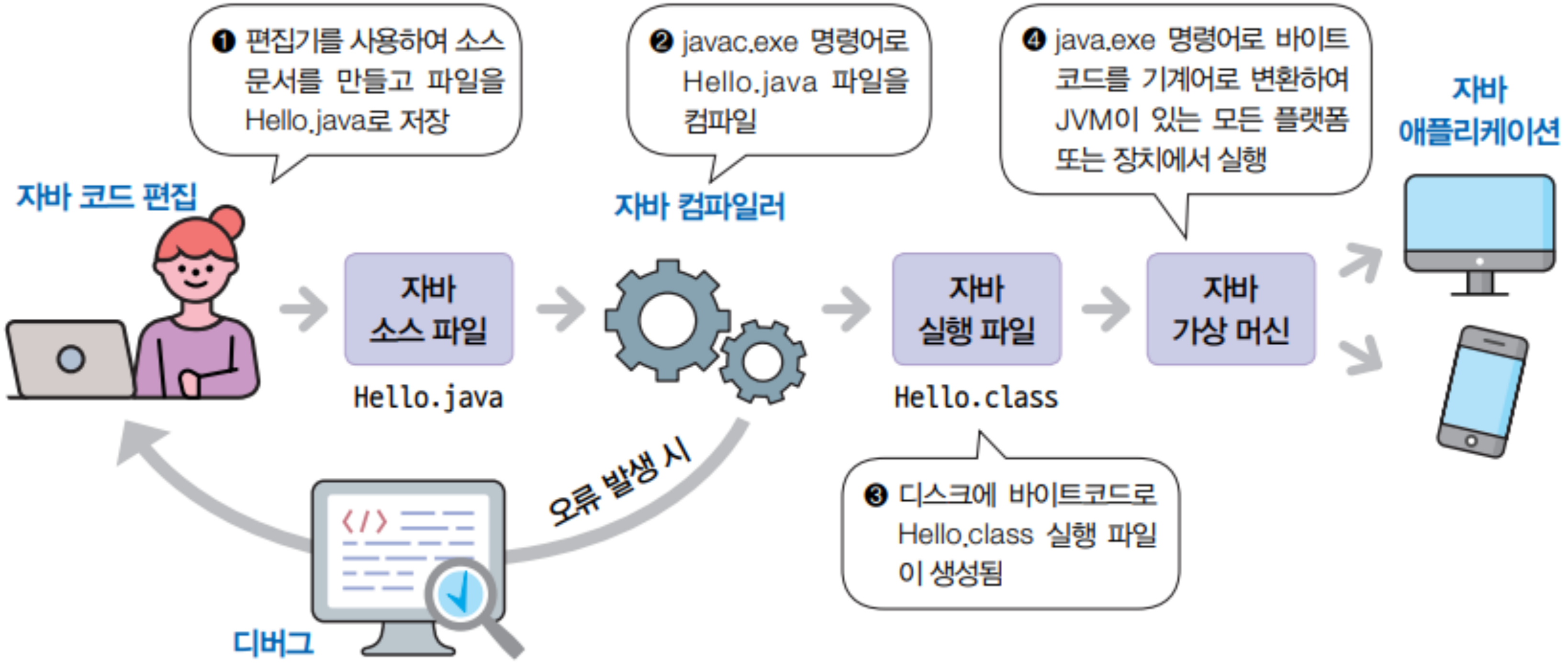
Java platform

- Java Runtime Environment (JRE)
  - JRE에는 runtime 라이브러리, 클래스 로더, JVM이 포함됨
  - ➔ Java program 실행을 위해서 JRE가 반드시 필요함

- Java Virtual Machine (JVM)
  - Byte code를 machine code로 변환함
  - JVM은 JRE의 일부
  - Java compiler는 JVM을 통해서 machine code로 변환
    - 다른 프로그래밍 언어는 특정 시스템에 대한 machine code를 생성 (플랫폼 독립적이지 않음)

Java platform

- Java program 실행 과정



❶ 편집기를 사용하여 소스 문서를 만들고 파일을 Hello.java로 저장

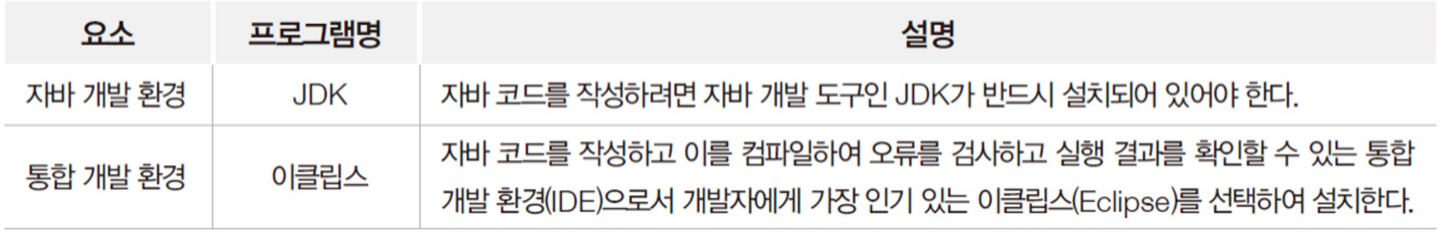
❷ javac.exe 명령어로 Hello.java 파일을 컴파일

❸ 디스크에 바이트코드로 Hello.class 실행 파일이 생성됨

❹ java.exe 명령어로 바이트 코드를 기계어로 변환하여 JVM이 있는 모든 플랫폼 또는 장치에서 실행

Java development environment setting

- Java development environment tools



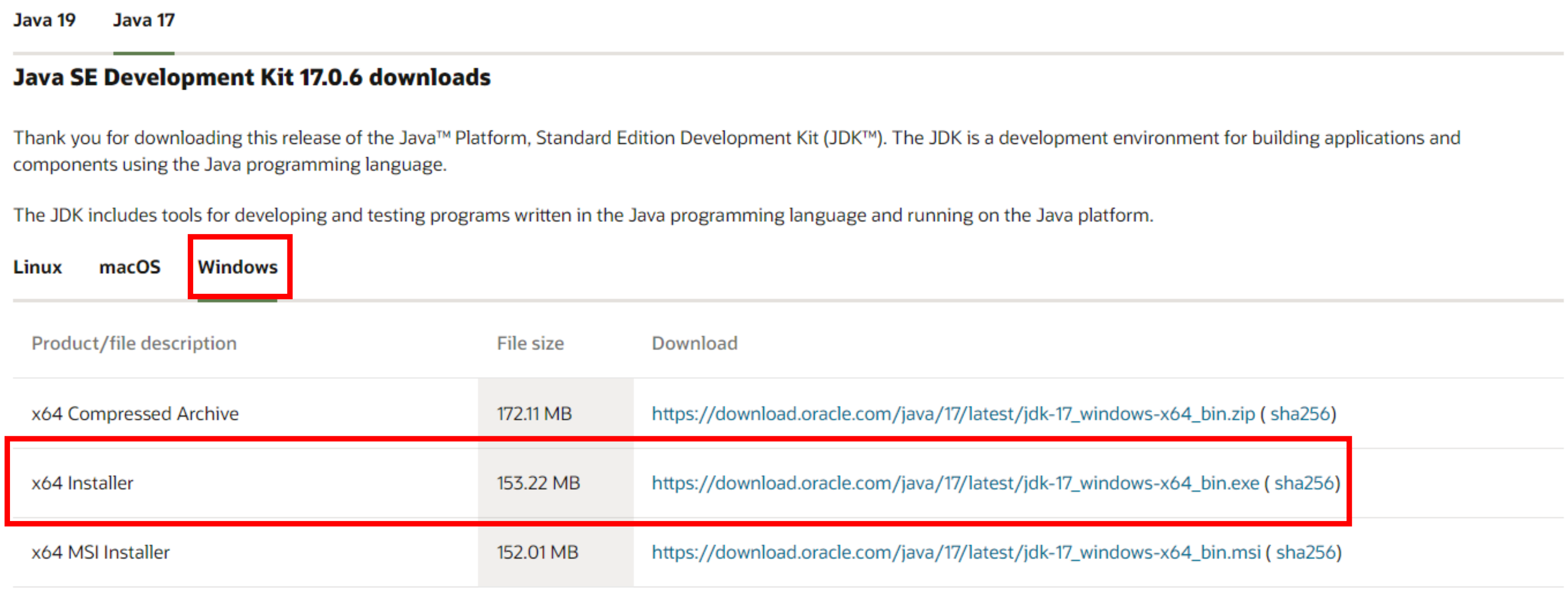
요소	프로그램명	설명
자바 개발 환경	JDK	자바 코드를 작성하려면 자바 개발 도구인 JDK가 반드시 설치되어 있어야 한다.
통합 개발 환경	이클립스	자바 코드를 작성하고 이를 컴파일하여 오류를 검사하고 실행 결과를 확인할 수 있는 통합 개발 환경(IDE)으로서 개발자에게 가장 인기 있는 이클립스(Eclipse)를 선택하여 설치한다.

- Integrated development environment (통합 개발 환경)
    - IDE가 없으면?

- IDE의 종류
    - Visual studio: mainly for C/C++/C#
    - Visual studio code (VS code), Jupyter Notebook: mainly for Python
    - Eclipse: mainly for Java
    - R Studio: mainly for R

Step 1: Java JDK installation

- JDK download link
  - https://www.oracle.com/java/technologies/downloads/



Java SE Development Kit 17.0.6 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

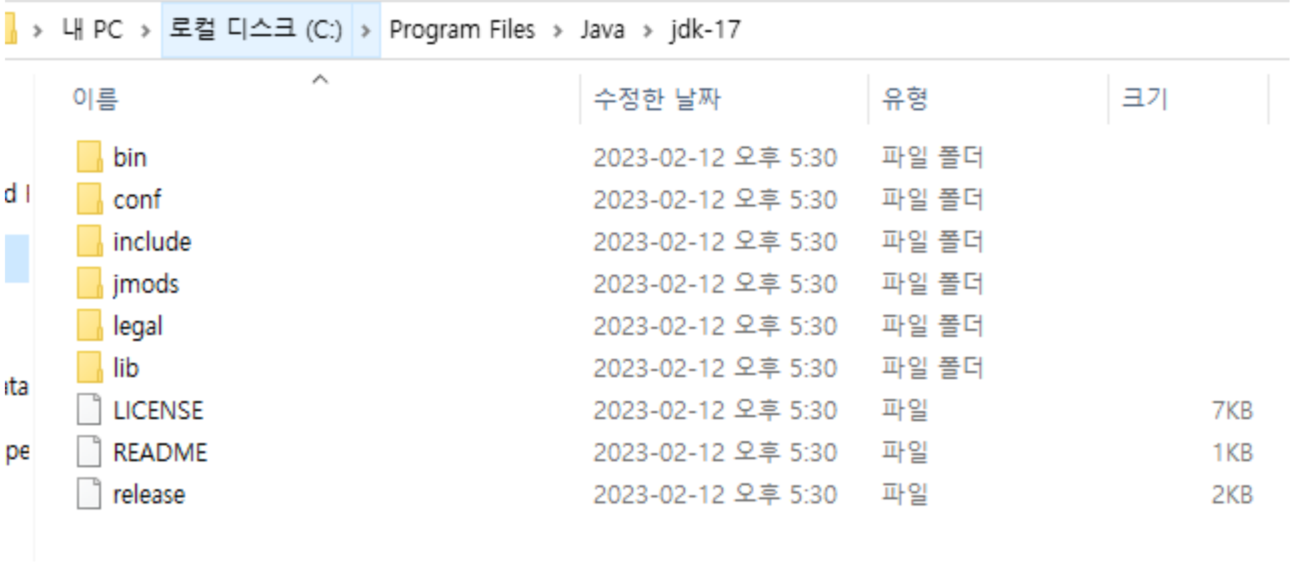
The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

Product/file description	File size	Download
x64 Compressed Archive	172.11 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256)
x64 Installer	153.22 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256)
x64 MSI Installer	152.01 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256)

Step 1: Java JDK installation

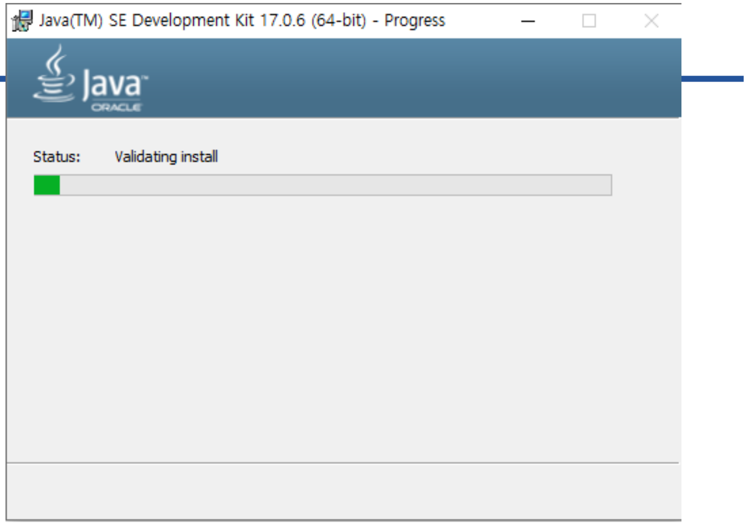
- Installing…
  - Next > Next > Close

- Check path
  - C:/Program Files/Java/jdk-17/bin



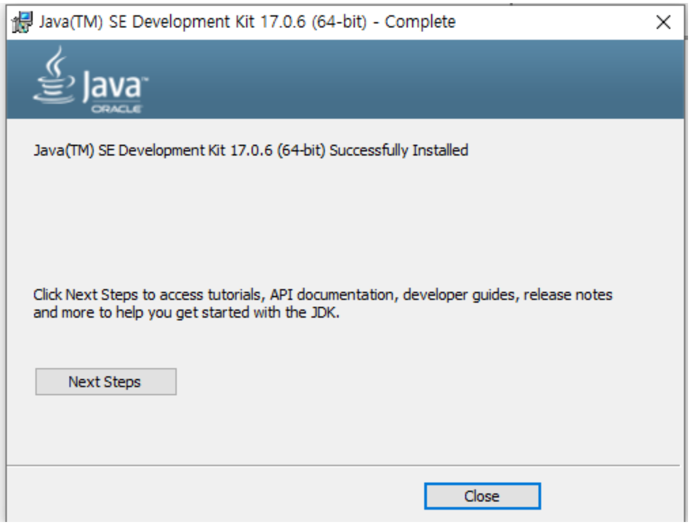
› 내 PC › 로컬 디스크 (C:) › Program Files › Java › jdk-17

이름	수정한 날짜	유형	크기
bin	2023-02-12 오후 5:30	파일 폴더	
conf	2023-02-12 오후 5:30	파일 폴더	
include	2023-02-12 오후 5:30	파일 폴더	
jmods	2023-02-12 오후 5:30	파일 폴더	
legal	2023-02-12 오후 5:30	파일 폴더	
lib	2023-02-12 오후 5:30	파일 폴더	
LICENSE	2023-02-12 오후 5:30	파일	7KB
README	2023-02-12 오후 5:30	파일	1KB
release	2023-02-12 오후 5:30	파일	2KB



Java(TM) SE Development Kit 17.0.6 (64-bit) - Progress

Status: Validating install



Java(TM) SE Development Kit 17.0.6 (64-bit) - Complete

Java(TM) SE Development Kit 17.0.6 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

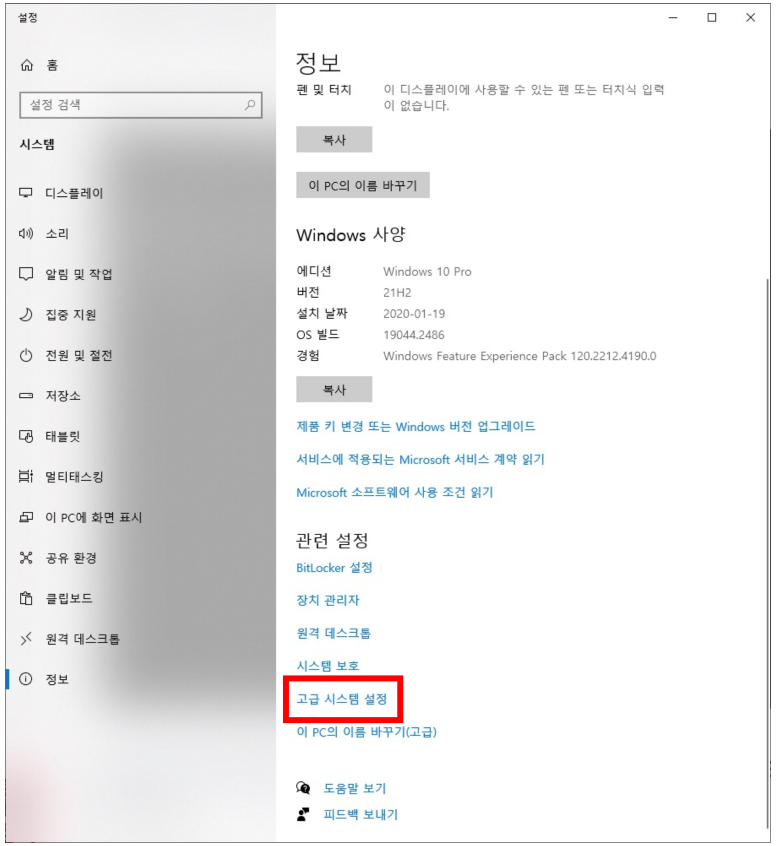
Step 1: Java JDK installation

- Environment variables 설정
  - Add the Java compiler to your PATH variable
    - 1) Select Start ➔ Control Panel ➔ Search control panel, for “environment”

      ➔ Edit the system environment variables -> Environment Variables
    - 2) Under “System variables”, find the one named “Path”. Select Edit
    - 3) Add location of bin folder in your system
      - (example) C:/Program Files/Java/jdk-17/bin

Step 1: Java JDK installation

- Environment variables 설정
    - Recommend to make “JAVA_HOME” variable,
      and add like “%JAVA_HOME%bin” in variable editing



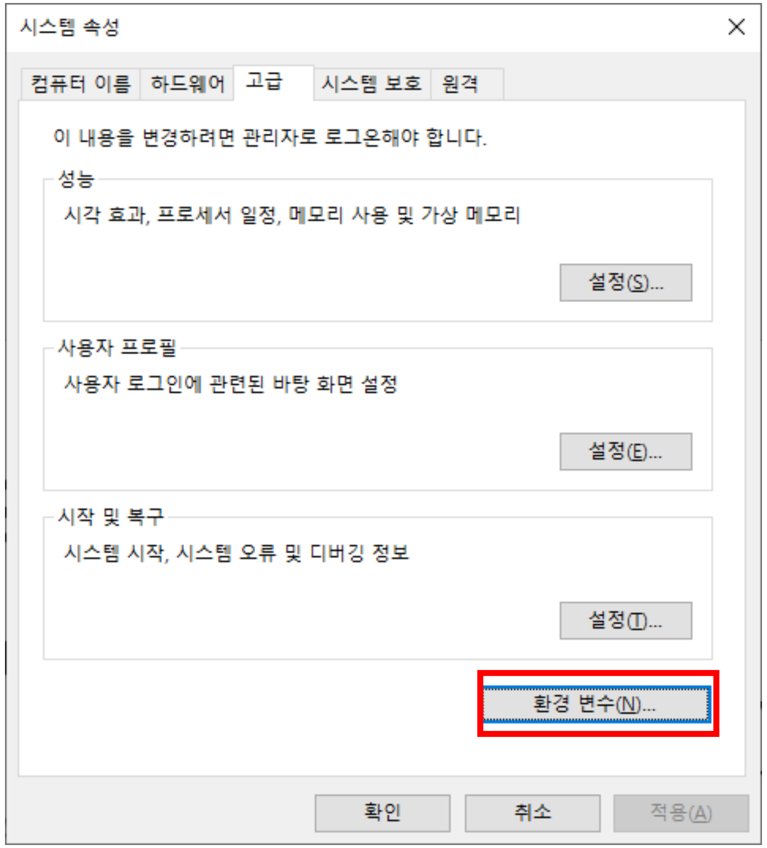
정보

펜 및 터치 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

Windows 사양

에디션	Windows 10 Pro
버전	21H2
설치 날짜	2020-01-19
OS 빌드	19044.2486
경험	Windows Feature Experience Pack 120.2212.4190.0

관련 설정



이 내용을 변경하려면 관리자로 로그온해야 합니다.

성능

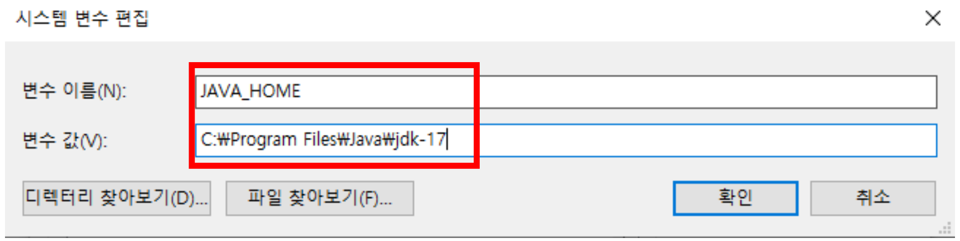
시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리

사용자 프로필

사용자 로그인에 관련된 바탕 화면 설정

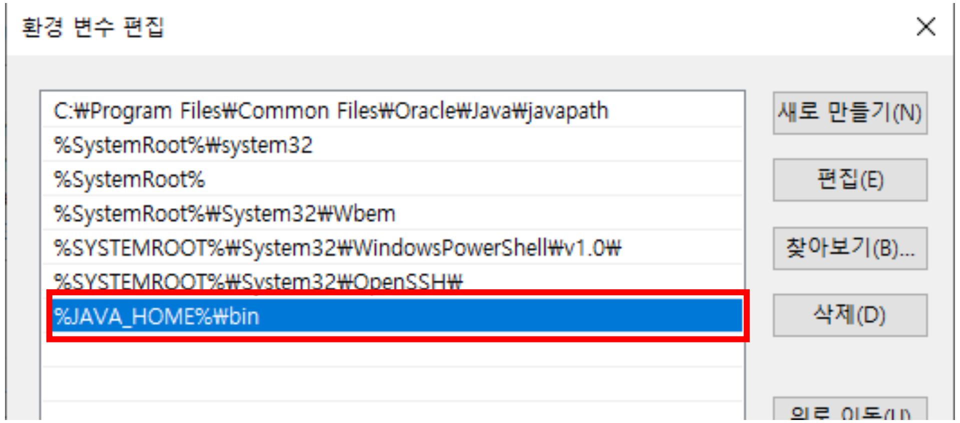
시작 및 복구

시스템 시작, 시스템 오류 및 디버깅 정보



변수 이름(N): JAVA_HOME

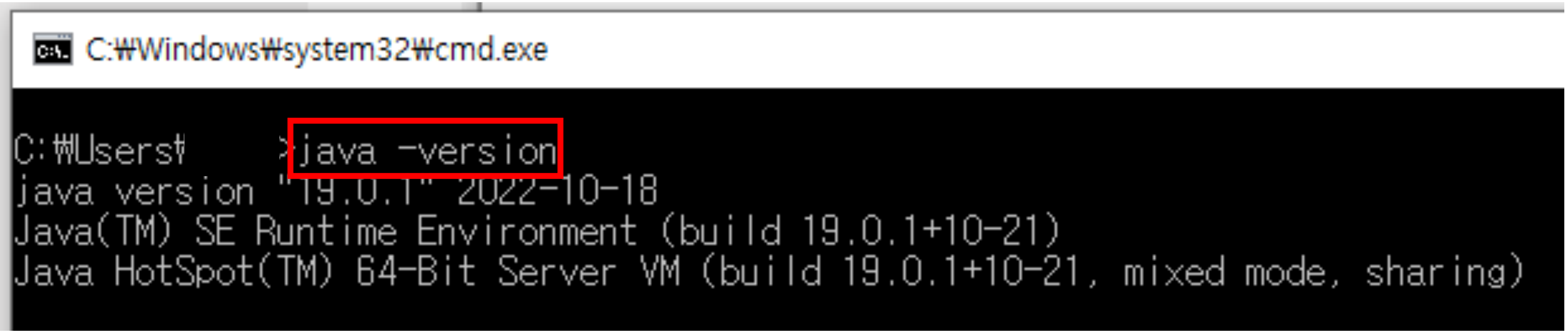
변수 값(V): C:₩Program Files₩Java₩jdk-17



```
C:₩Program Files₩Common Files₩Oracle₩Java₩javapath
%SystemRoot%₩system32
%SystemRoot%
%SystemRoot%₩System32₩Wbem
%SYSTEMROOT%₩System32₩WindowsPowerShell₩v1.0₩
%SYSTEMROOT%₩System32₩OpenSSH₩
%JAVA_HOME%₩bin
```

Step 1: Java JDK installation

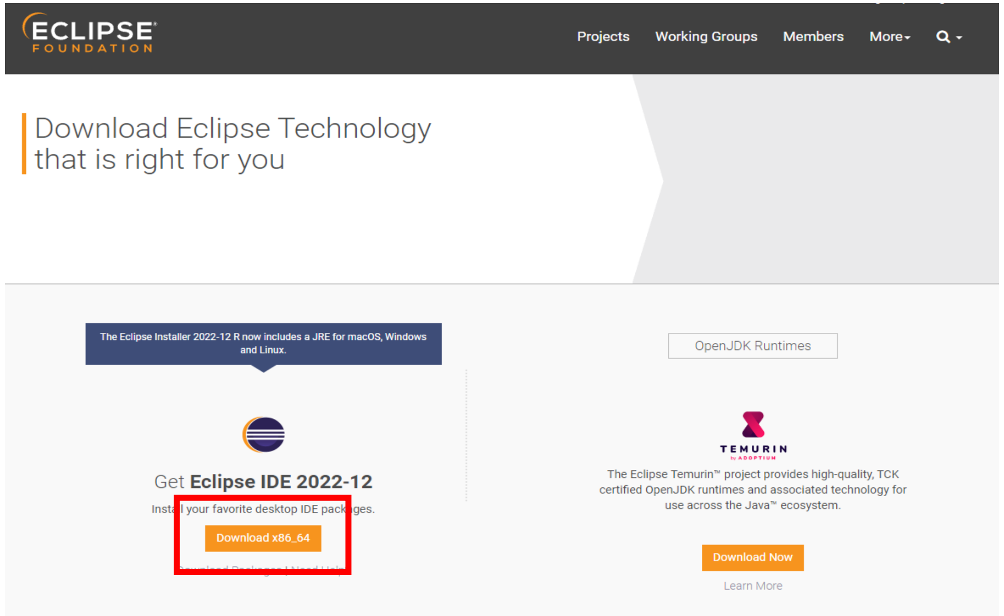
- Test
  - 1) win-key + R
  - 2) type “cmd”
  - 3) type “java -version” (Please be care space between “java” and “-version”)



```
C:₩Users₩[redacted]>java -version
java version "19.0.1" 2022-10-18
Java(TM) SE Runtime Environment (build 19.0.1+10-21)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.1+10-21, mixed mode, sharing)
```

Step 2: Eclipse IDE installation

- Eclipse download link
  - https://www.eclipse.org/downloads/



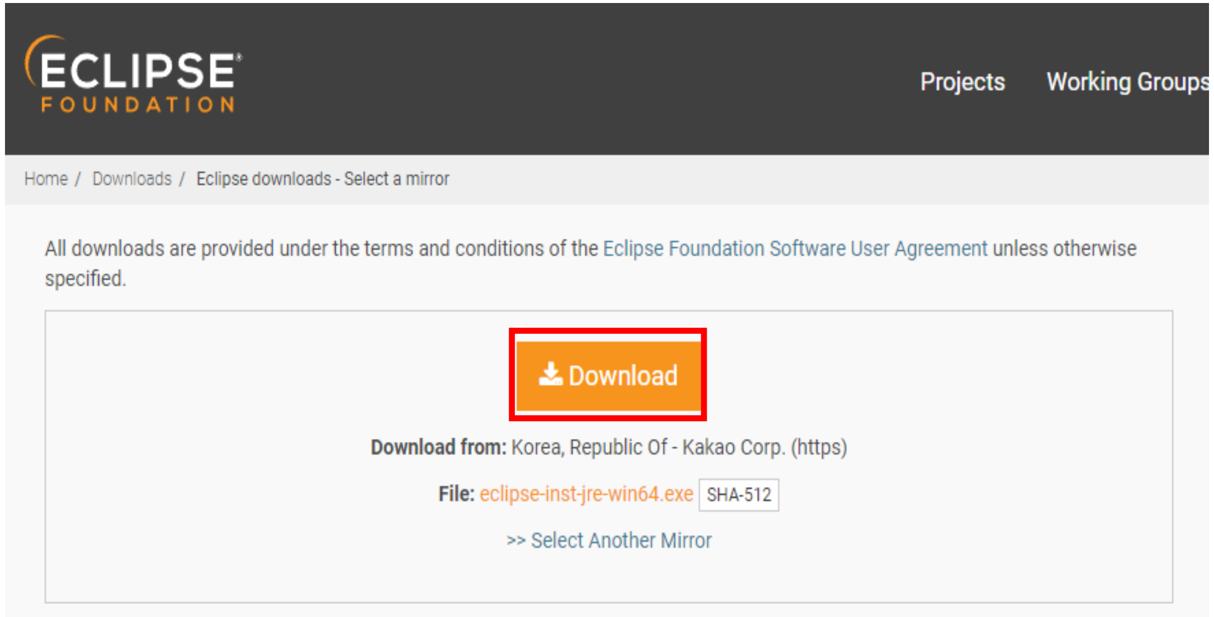
Download Eclipse Technology that is right for you

The Eclipse Installer 2022-12 R now includes a JRE for macOS, Windows and Linux.

Get **Eclipse IDE 2022-12**

Install your favorite desktop IDE packages.

The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem.



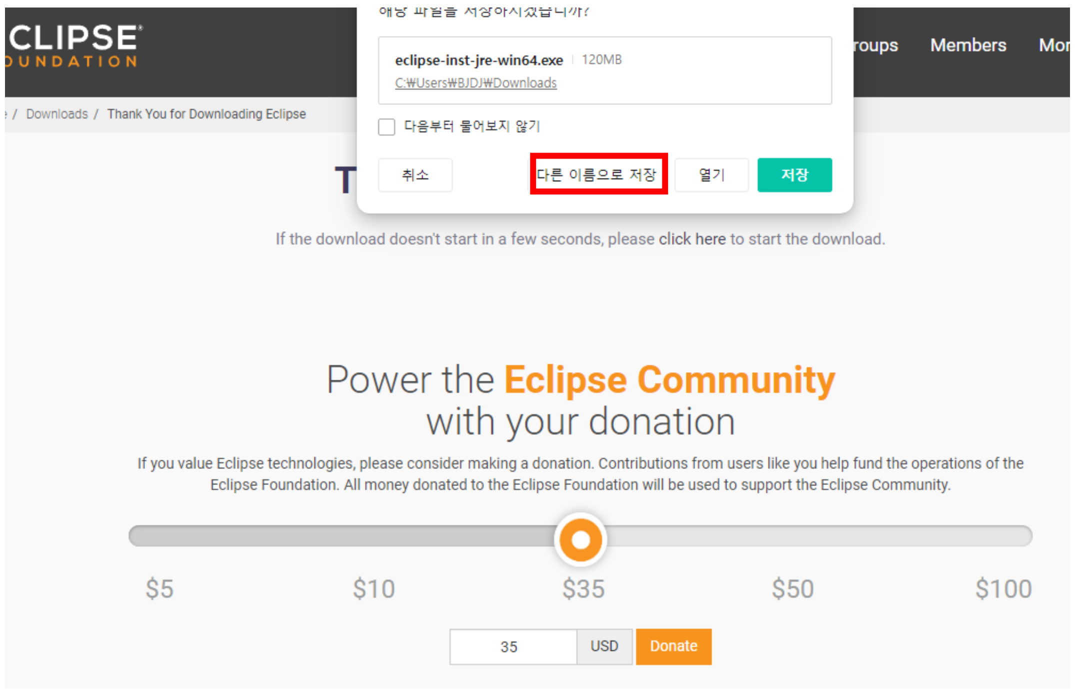
All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download from: Korea, Republic Of - Kakao Corp. (https)

File: eclipse-inst-jre-win64.exe

Step 2: Eclipse IDE installation

- Installing…
  - Don’t donate it
  - Please select “Eclipse IDE for Java Developers”



해당 파일을 저장하시겠습니까?

eclipse-inst-jre-win64.exe | 120MB

C:₩Users₩BJDJ₩Downloads

□ 다음부터 물어보지 않기

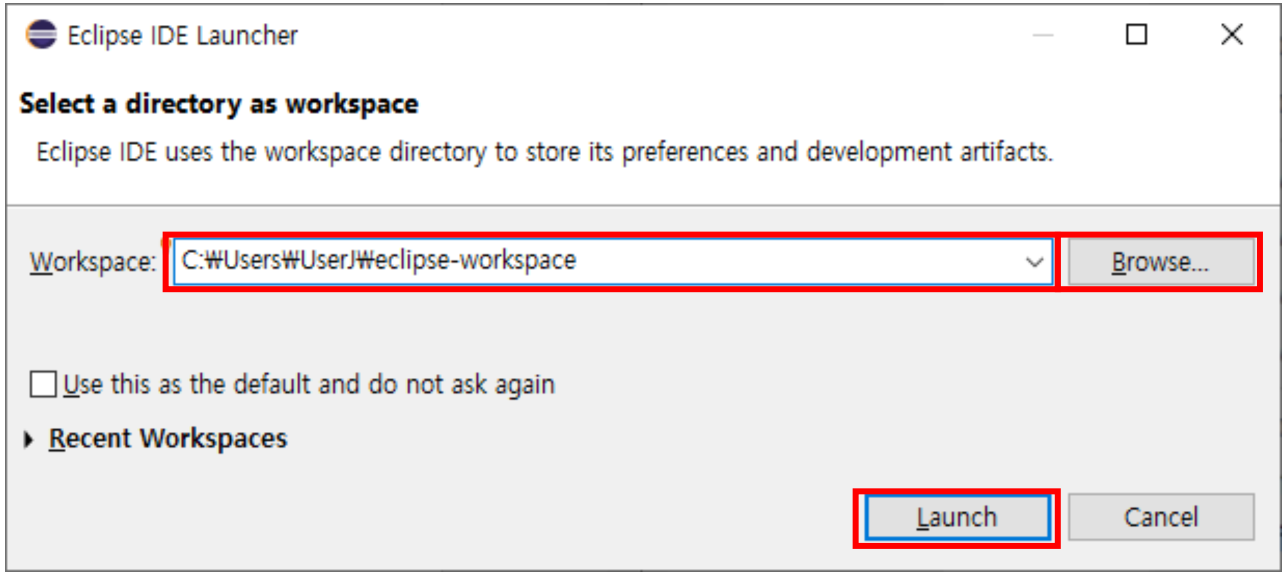
If the download doesn't start in a few seconds, please click here to start the download.

Power the **Eclipse Community** with your donation

If you value Eclipse technologies, please consider making a donation. Contributions from users like you help fund the operations of the Eclipse Foundation. All money donated to the Eclipse Foundation will be used to support the Eclipse Community.

Step 3: “Hello World!” test

- Setup
  - Set the workspace by using “Browse…” button
  - Click Launch



Eclipse IDE Launcher

Select a directory as workspace

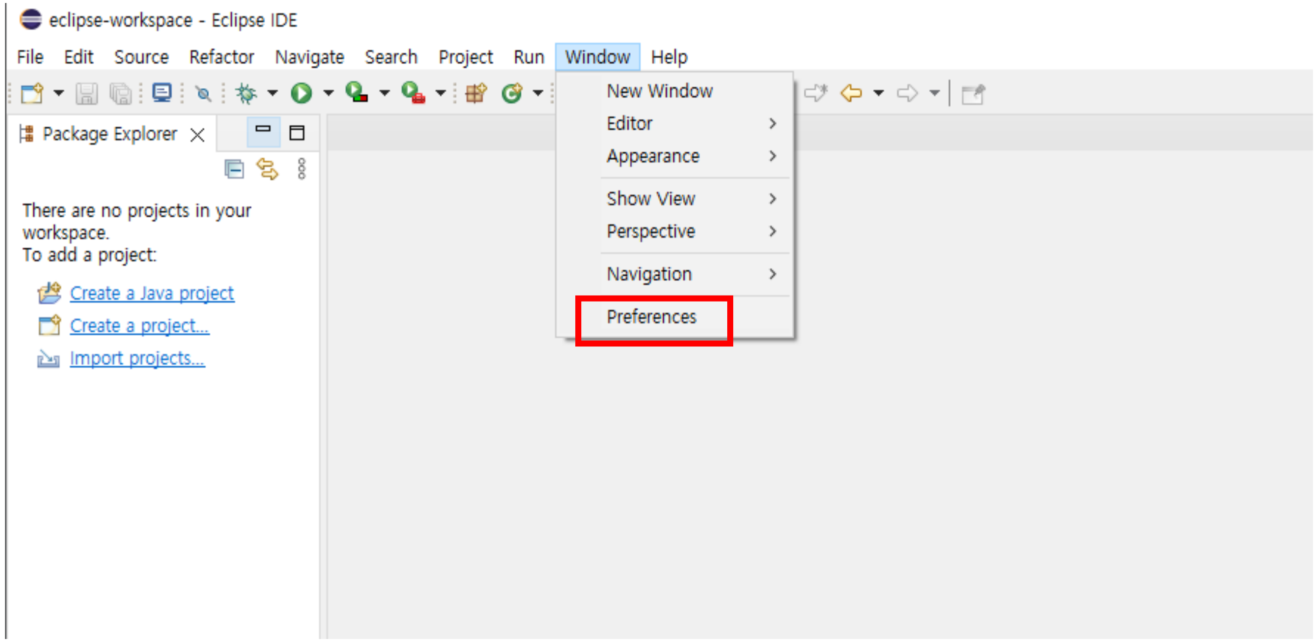
Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: C:₩Users₩UserJ₩eclipse-workspace

□ Use this as the default and do not ask again

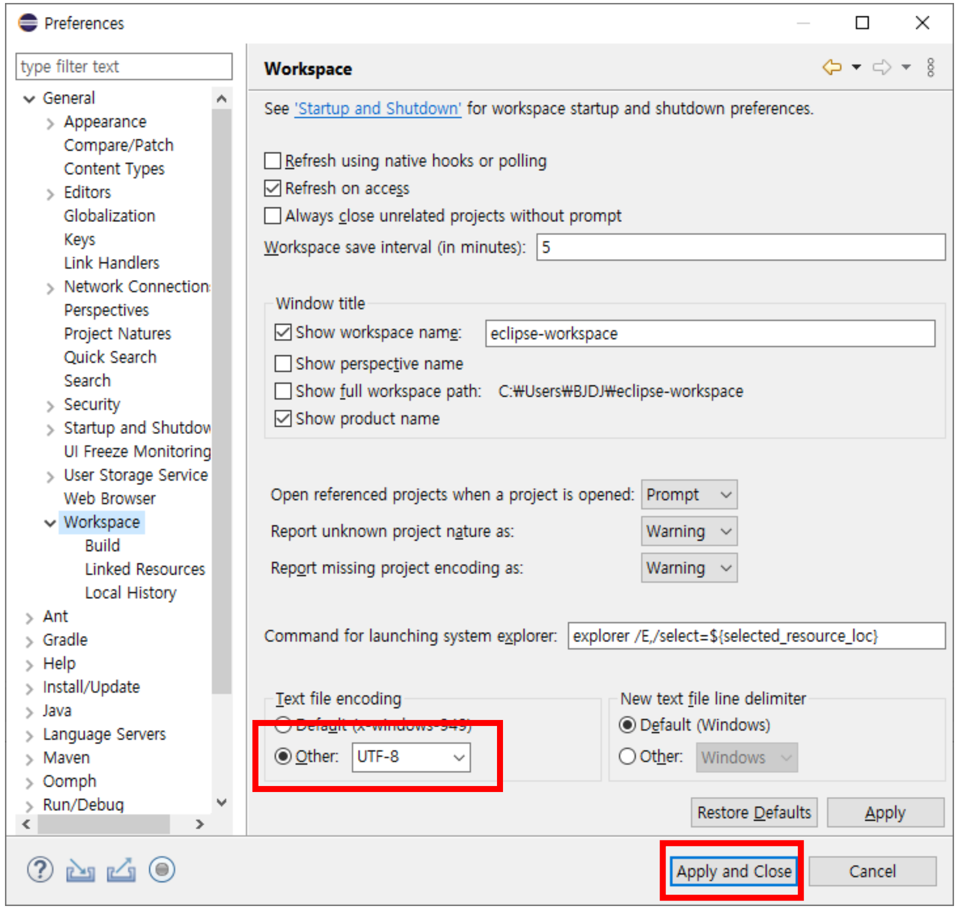
Step 3: “Hello World!” test

- Setup
    - Language encoding setting



There are no projects in your workspace.
To add a project:

Create a Java project
Create a project...
Import projects...



Workspace

See 'Startup and Shutdown' for workspace startup and shutdown preferences.

□ Refresh using native hooks or polling
☑ Refresh on access
□ Always close unrelated projects without prompt
Workspace save interval (in minutes): 5

Window title
☑ Show workspace name: eclipse-workspace
□ Show perspective name
□ Show full workspace path: C:₩Users₩BJD₩eclipse-workspace
☑ Show product name

Open referenced projects when a project is opened: Prompt
Report unknown project nature as: Warning
Report missing project encoding as: Warning

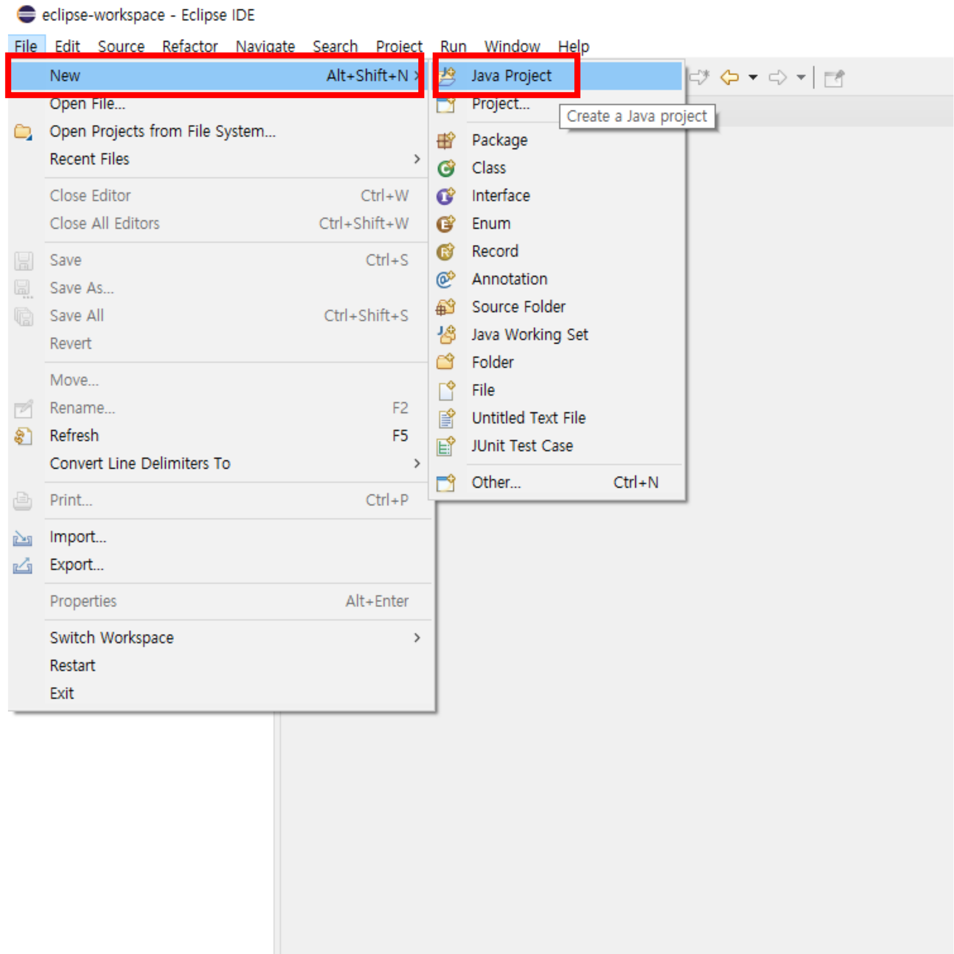
Command for launching system explorer: explorer /E,/select=${selected_resource_loc}

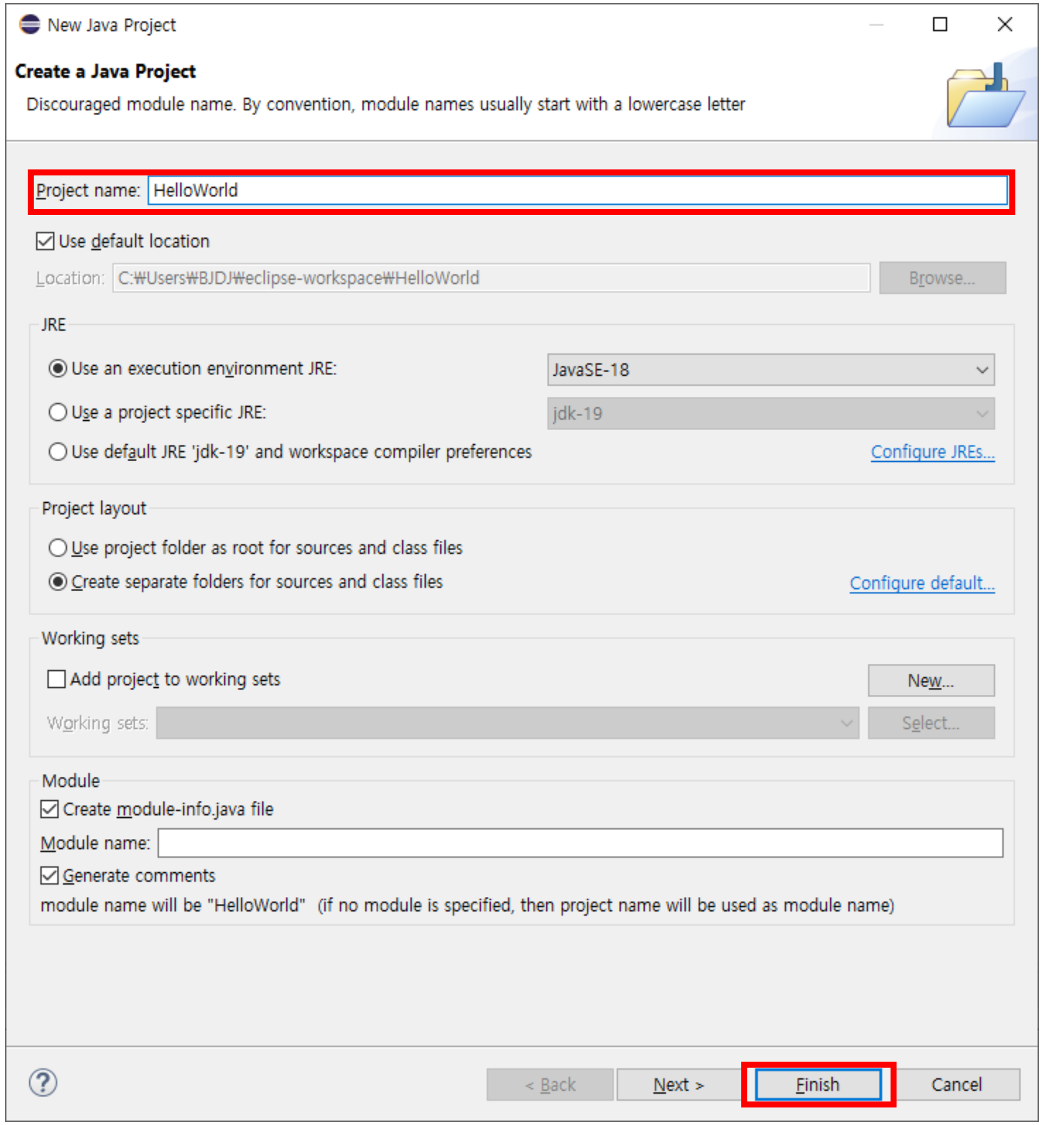
Text file encoding
○ Default (x-windows-949)
● Other: UTF-8

New text file line delimiter
● Default (Windows)
○ Other: Windows

Step 3: “Hello World!” test

- Create Java Project




New Java Project

Create a Java Project

Discouraged module name. By convention, module names usually start with a lowercase letter

Project name: HelloWorld

☑ Use default location

Location: C:₩Users₩BJDJ₩eclipse-workspace₩HelloWorld

JRE

● Use an execution environment JRE: JavaSE-18

○ Use a project specific JRE: jdk-19

○ Use default JRE 'jdk-19' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

Working sets

☐ Add project to working sets

Working sets:

Module

☑ Create module-info.java file

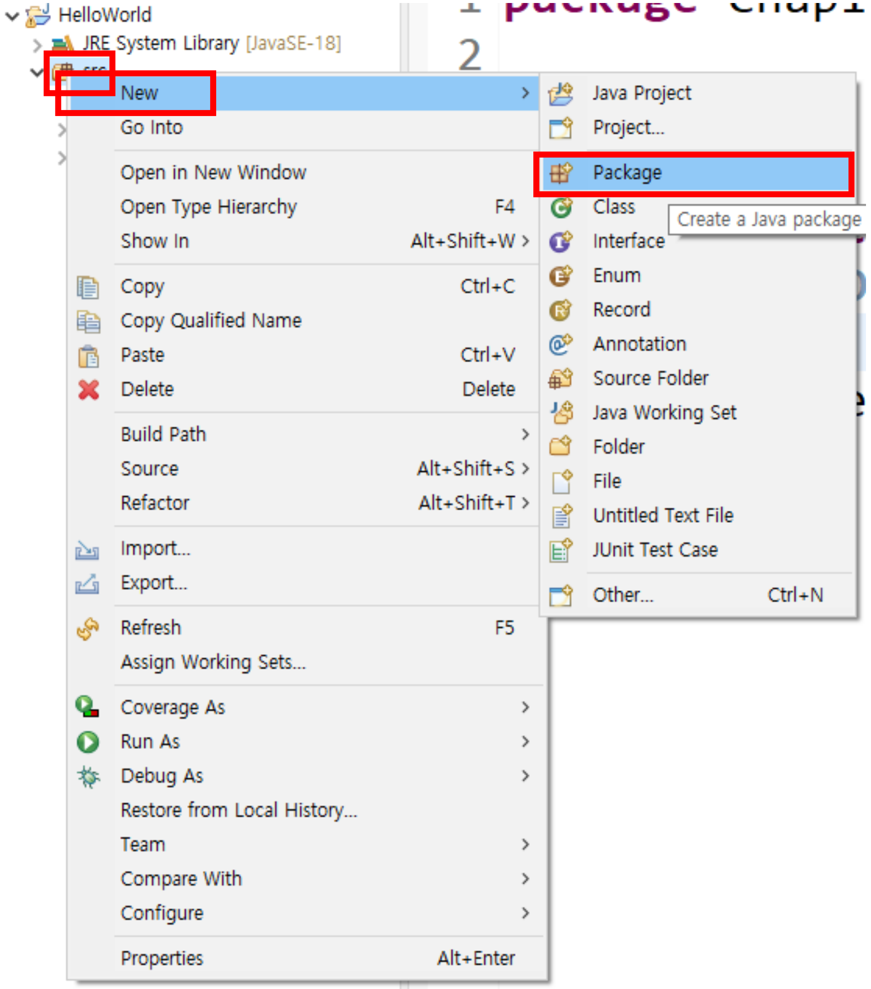
Module name:

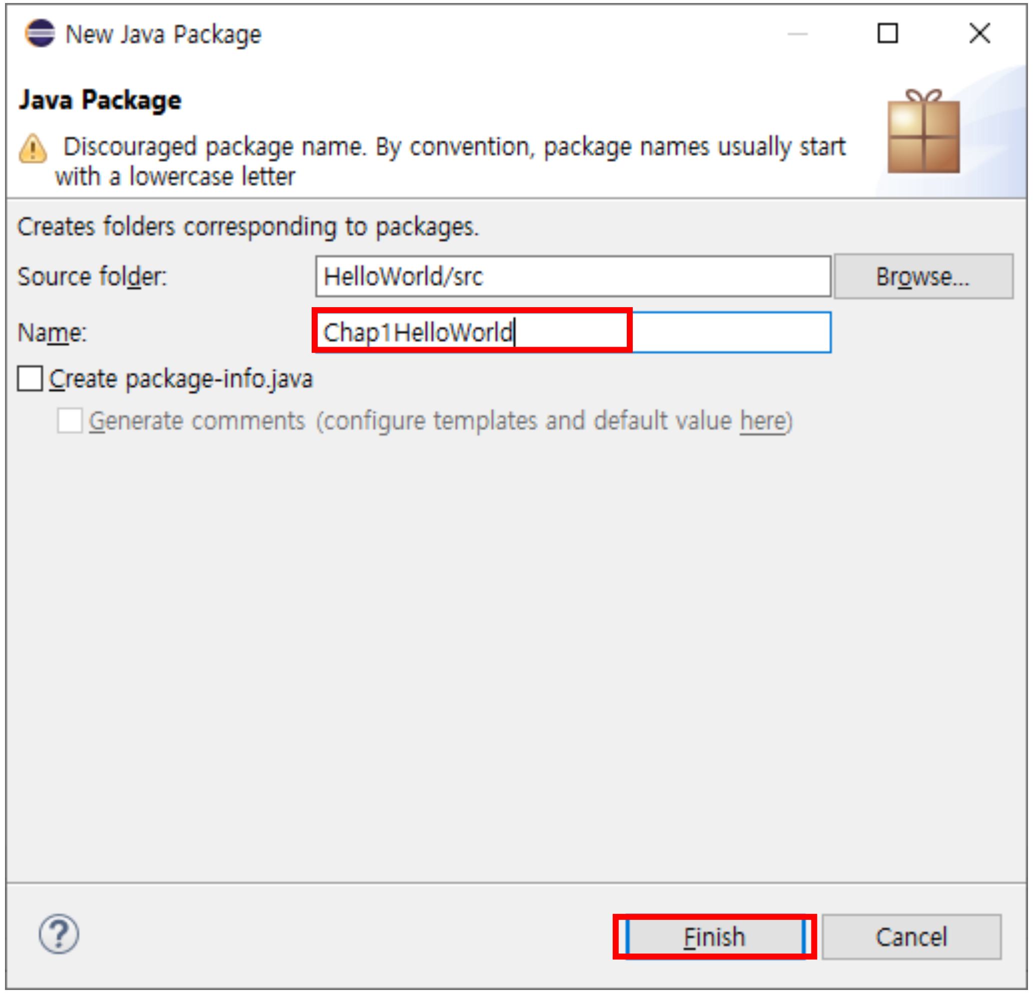
☑ Generate comments

module name will be "HelloWorld" (if no module is specified, then project name will be used as module name)

Step 3: "Hello World!" test

- Create Java Project ➔ Create Package





New Java Package

Java Package

Discouraged package name. By convention, package names usually start with a lowercase letter

Creates folders corresponding to packages.

Source folder: HelloWorld/src

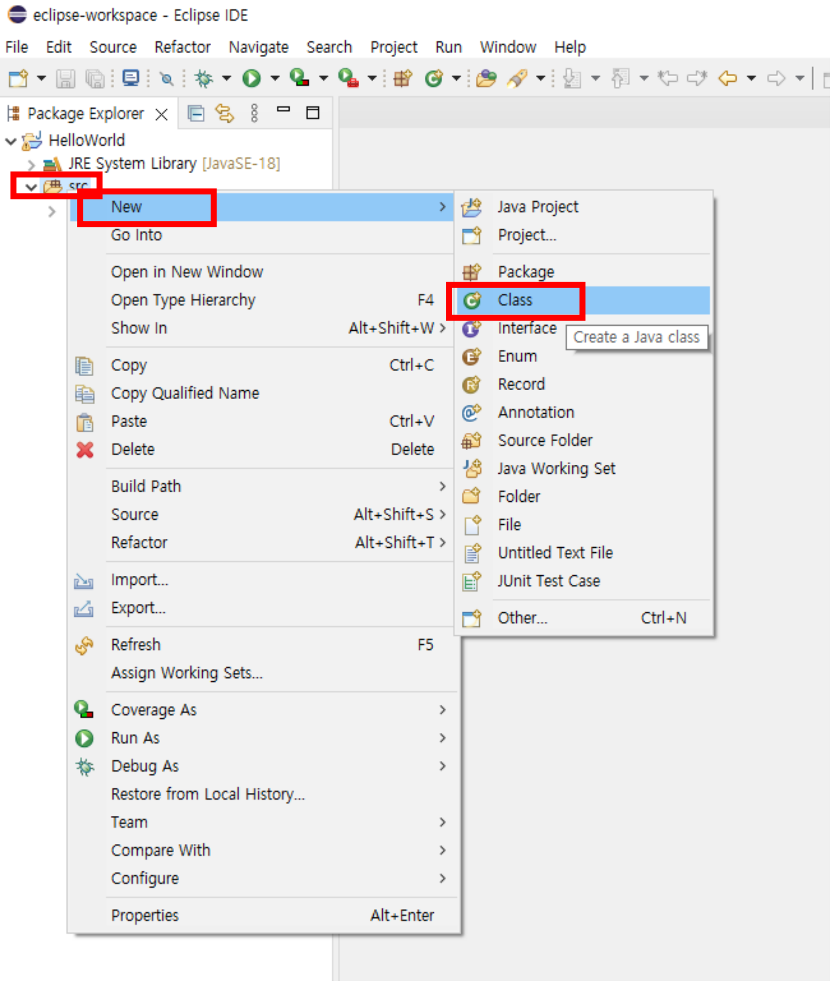
Name: Chap1HelloWorld

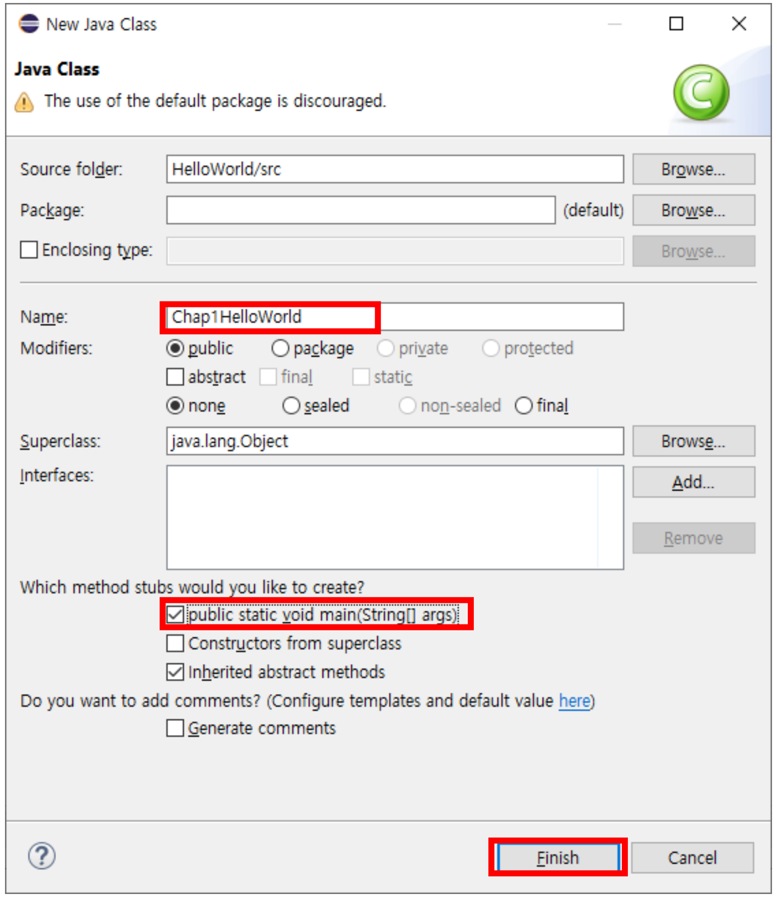
□ Create package-info.java

□ Generate comments (configure templates and default value here)

Step 3: “Hello World!” test

- Create Java Project ➔ Create Package ➔ Create Class




New Java Class

Java Class

The use of the default package is discouraged.

Source folder: HelloWorld/src [Browse...]

Package: (default) [Browse...]

□ Enclosing type: [Browse...]

Name: Chap1HelloWorld

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

● none ○ sealed ○ non-sealed ○ final

Superclass: java.lang.Object [Browse...]

Interfaces: [Add...] [Remove]

Which method stubs would you like to create?

☑ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

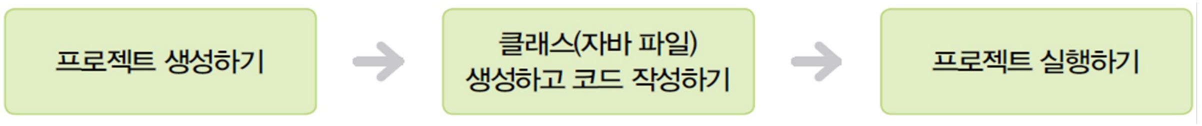
□ Generate comments

[Finish] [Cancel]

Java project structure

- Java Project ➔ Package ➔ Class

- Project
  - 1개의 거대한 프로젝트
  - Package들의 집합



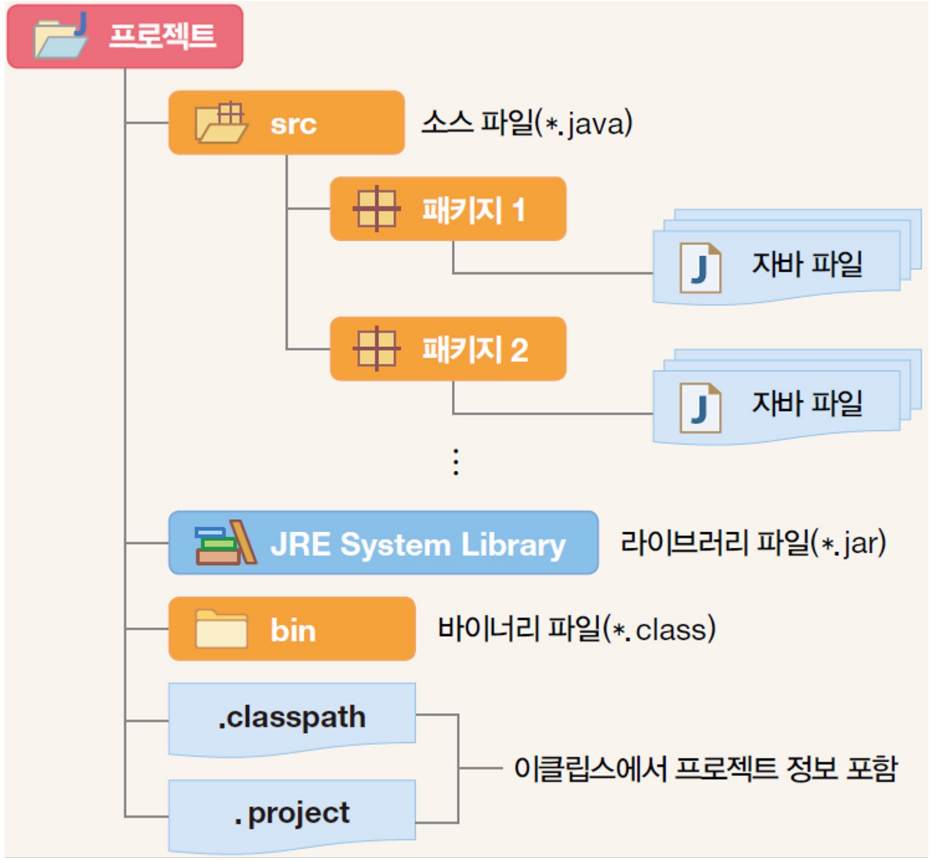
프로젝트 생성하기 → 클래스(자바 파일) 생성하고 코드 작성하기 → 프로젝트 실행하기

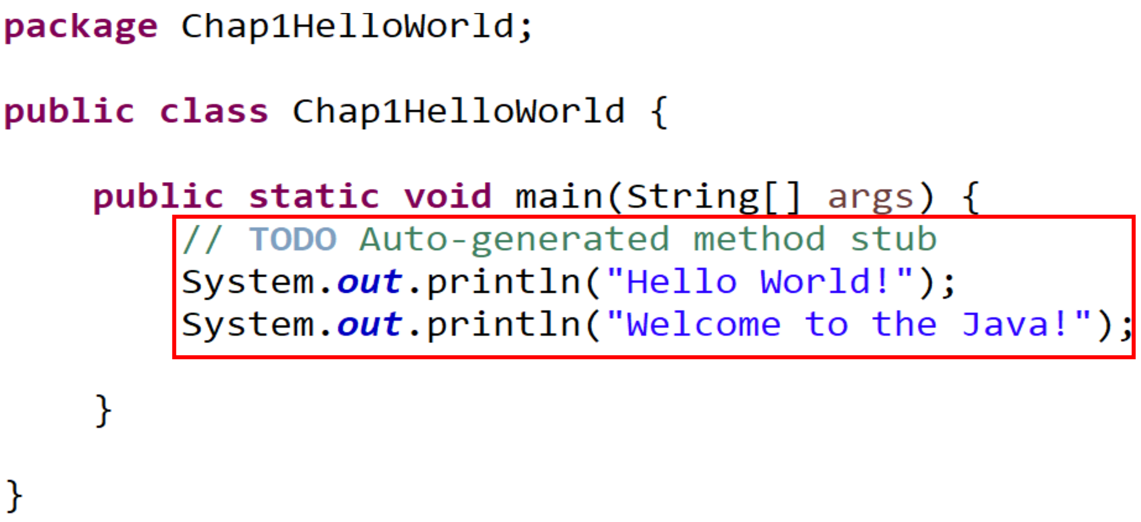
- Package
  - Class, Interface, Enum 등 Instance(인스턴스)의 집합

- Class
  - Function(함수), Method(메소드) 등 으로 구성된 로직

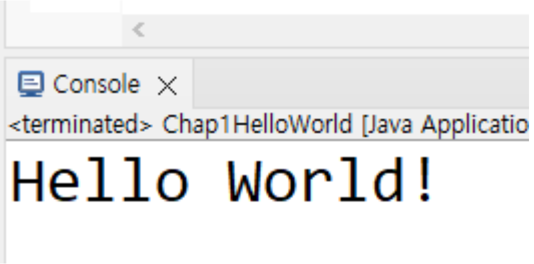
Java project structure

- Java project의 기본 구조





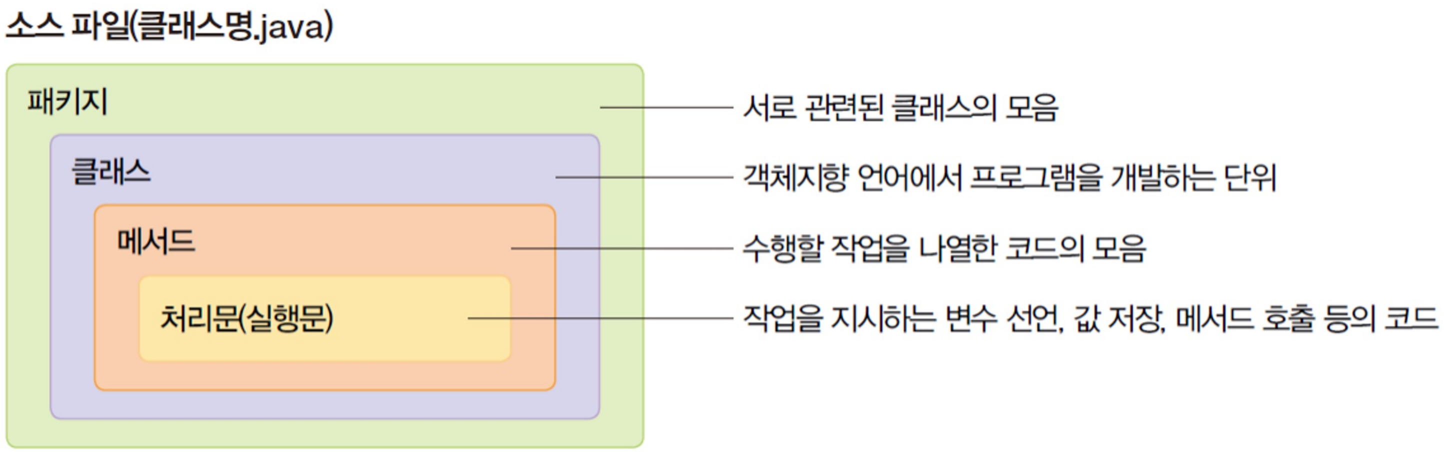
```
package Chap1HelloWorld;

public class Chap1HelloWorld {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World!");
        System.out.println("Welcome to the Java!");

    }

}
```



```
<terminated> Chap1HelloWorld [Java Applicatio
Hello World!
```

Java project structure

- Java project의 기본 구조
  - Java 프로그램은 하나 이상의 class로 구성됨
  - 각 class의 프로그램 코드를 별도의 source file에 저장하고
    각 source file명을 source file에 정의된 class명과 동일하게 지정해야 함
    - 다르면 오류 발생
    - 모든 Java source file의 확장자: .java



소스 파일(클래스명.java)

패키지 — 서로 관련된 클래스의 모음

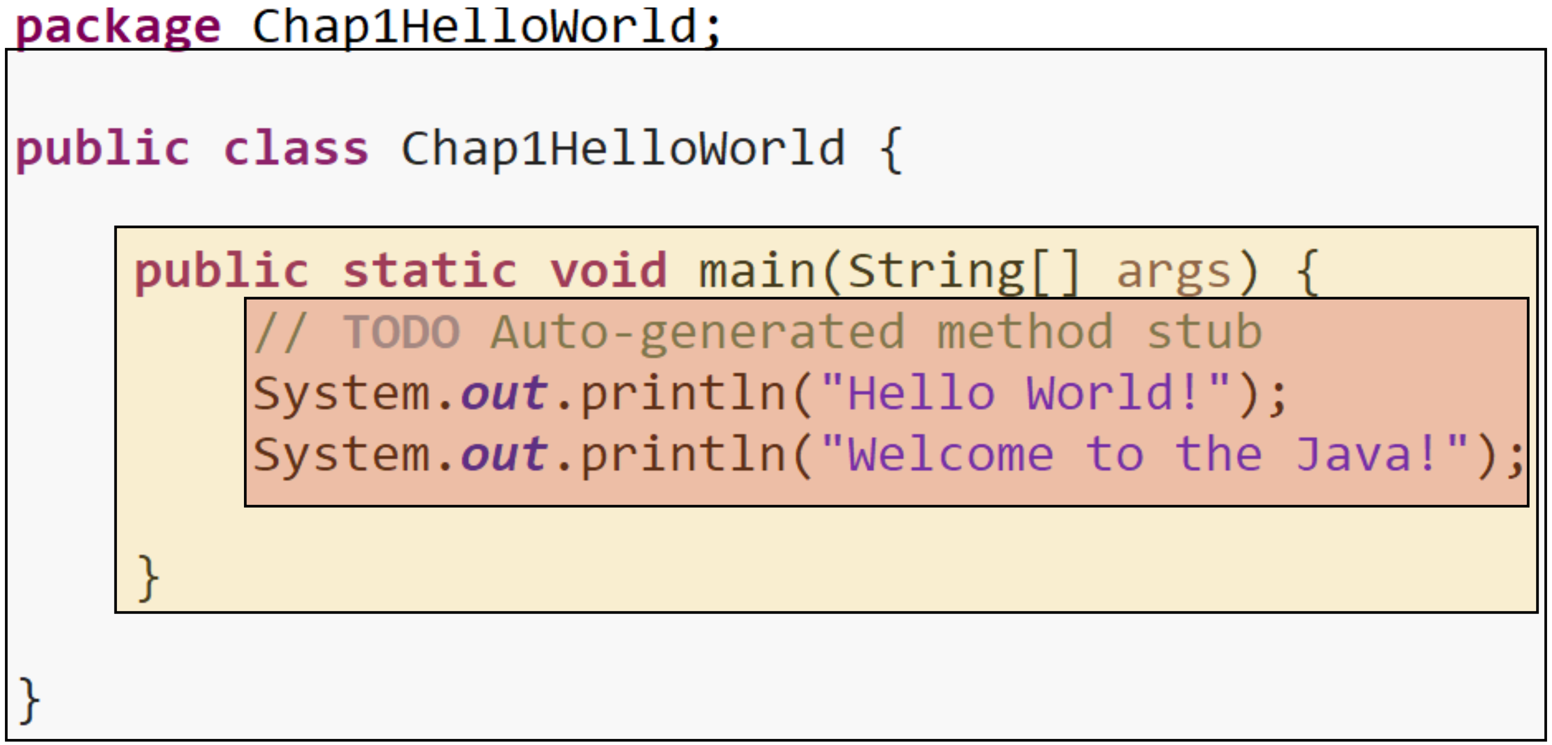
클래스 — 객체지향 언어에서 프로그램을 개발하는 단위

메서드 — 수행할 작업을 나열한 코드의 모음

처리문(실행문) — 작업을 지시하는 변수 선언, 값 저장, 메서드 호출 등의 코드

Java project structure

- Chap1HeeloWorld.java 파일의 기본 구조



```java
package Chap1HelloWorld;

public class Chap1HelloWorld {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World!");
        System.out.println("Welcome to the Java!");

    }

}
```

Java project structure

- Package (패키지)
  - 패키지는 기능을 기반으로 class를 구성하는 데 사용됨
  - Package문: class가 저장되는 namespace
  - Package문을 생략하면 이름 없는 기본 package에 class명만 선언됨

```
package 패키지명;
```

Java project structure

- Class (클래스)
  - class 키워드를 사용하여 class를 선언함
  - class명은 대부분 첫 글자가 대문자로 시작
  - 전체 클래스의 내용은 중괄호 ({ })안에 포함되어야 함
  - public 키워드를 사용하여 패키지 외부에서 클래스의 접근 가능성을 지정함

```
public class 클래스명 {

}
```

  - 반드시 .java 파일명과 동일해야함

Java project structure

- main() method
  - 모든 java 프로그램의 시작점이자 진입점
    - 즉, Java application이 시작될 때마다 가장 먼저 호출되는 method
  - method명은 대부분 소문자로 시작
  - 하나의 Java 프로그램에는 main() method를 가진 class가 반드시 존재해야 함

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

}
```

Useful Shortcut

- Frequently used shortcut keys

  (1) Ctrl + F11: Compile & Run

  (2) Ctrl + /: Single line comment & Remove comment

  (3) Ctrl + Shift + ₩: Multiple lines comment

  (4) Ctrl + Shift + /: Remove block comment

  (5) Ctrl + I (i): Code sorting

5. Basic data representation in computer

Data representation

- Data
  - symbol that represent people, events, things, and ideas
    - e.g., a name, a number, colors, in a photograph, notes in musical composition, etc.

- How to represent data in computer?
  - a form in which data is stored, processed, and transmitted
  - device (PC or smart phone, etc.) stores data in digital formats that can be handled by electronic circuitry

Primary ways data represented in computers

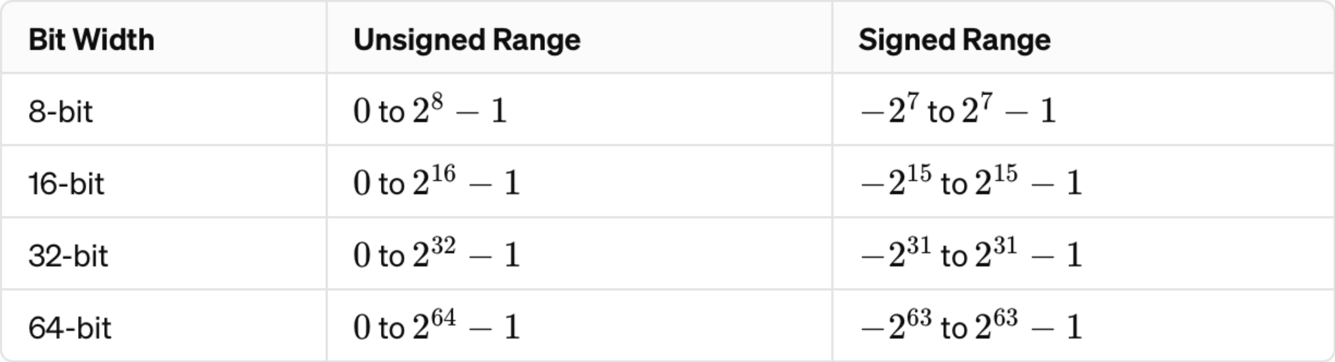
- Binary numbers (or binary codes)
  - the most basic form of data representation
  - all data is represented as sequences of bits (0s or 1s)
  - numbers, characters, and even executable instructions can be encoded in binary

- Hexadecimal numbers
  - a more compact form of binary representation
  - four bits are represented by a single hexadecimal digit (0-9 and A-F)

- ASCII code (The American Standard Code for Information Interchange)
  - a character encoding standard used to represent text in computer

Bit and byte

- Bit
  - the smallest unit of data in computer for a single binary value; either 0 or 1
  - can represent a range of different meanings
    - e.g., 1/0, on/off, true/false, or any other two-state system



Bit Width	Unsigned Range	Signed Range
8-bit	0 to $2^8 - 1$	-2^7 to $2^7 - 1$
16-bit	0 to $2^{16} - 1$	-2^{15} to $2^{15} - 1$
32-bit	0 to $2^{32} - 1$	-2^{31} to $2^{31} - 1$
64-bit	0 to $2^{64} - 1$	-2^{63} to $2^{63} - 1$

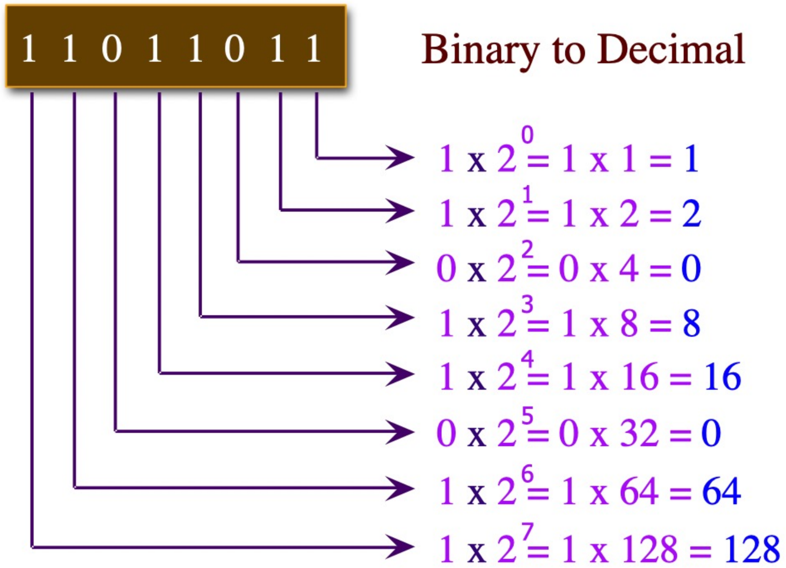
- Byte
  - a unit of digital information that most commonly consists of 8 bits
  - can represent 256 different values (from 0 to 255 in decimal)

Number system conversion

- Binary-decimal conversion

Decimal (BASE 10)	0	1	2	3	4	5	8	10
Binary (BASE 2)	0	1	10	11	100	101	1000	1010

- binary to decimal conversion



Binary to Decimal

1 1 0 1 1 0 1 1

$1 \times 2^0 = 1 \times 1 = 1$

$1 \times 2^1 = 1 \times 2 = 2$

$0 \times 2^2 = 0 \times 4 = 0$

$1 \times 2^3 = 1 \times 8 = 8$

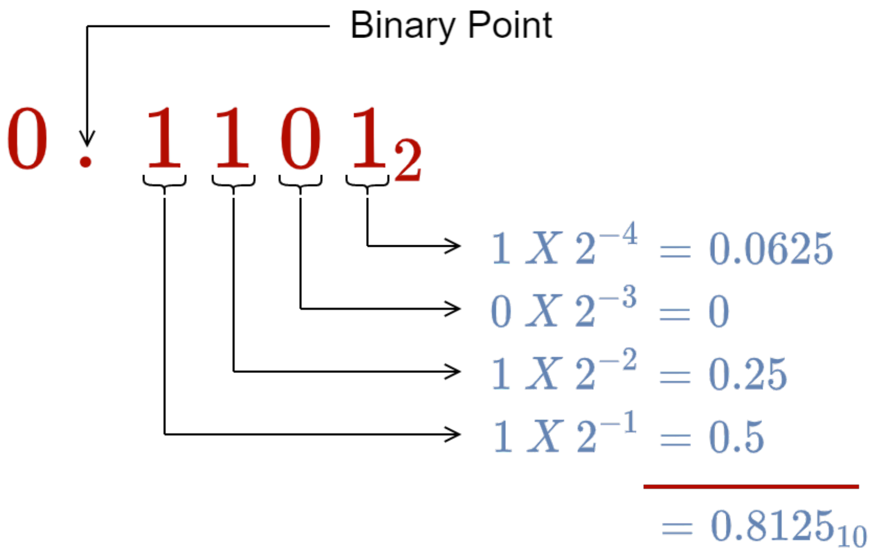
$1 \times 2^4 = 1 \times 16 = 16$

$0 \times 2^5 = 0 \times 32 = 0$

$1 \times 2^6 = 1 \times 64 = 64$

$1 \times 2^7 = 1 \times 128 = 128$

$$1 + 2 + 8 + 16 + 64 + 128 = 219$$



Binary Point

$0 . 1101_2$

$1\ X\ 2^{-4} = 0.0625$

$0\ X\ 2^{-3} = 0$

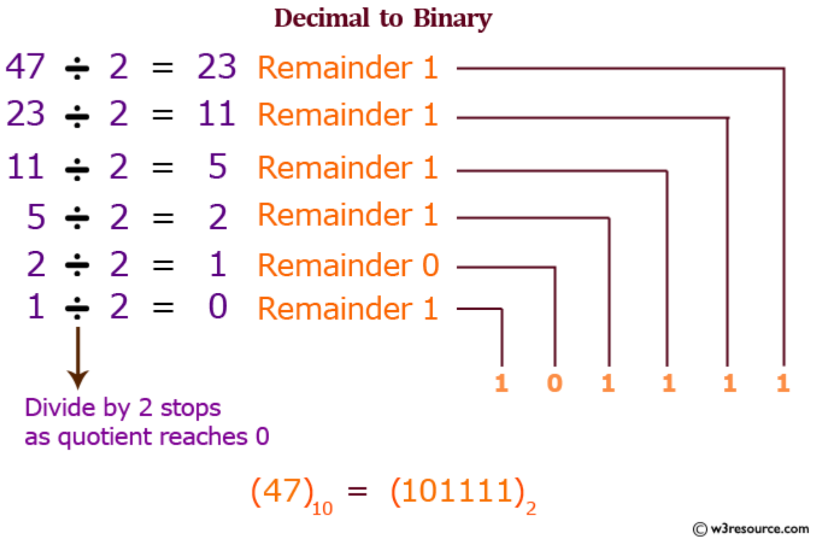
$1\ X\ 2^{-2} = 0.25$

$1\ X\ 2^{-1} = 0.5$

$= 0.8125_{10}$

Number system conversion

- Binary-decimal conversion
  - decimal to binary conversion
    - integer part: divide this number repeatedly by 2 until the quotient becomes 0
    - fractional part: multiply the fractional part repeatedly by 2 until it becomes 0
    - example: 47.375 (decimal) to binary conversion



Decimal to Binary

47 ÷ 2 = 23 Remainder 1

23 ÷ 2 = 11 Remainder 1

11 ÷ 2 = 5 Remainder 1

5 ÷ 2 = 2 Remainder 1

2 ÷ 2 = 1 Remainder 0

1 ÷ 2 = 0 Remainder 1

$(47)_{10} = (101111)_2$

© w3resource.com

- fractional part: 0.375

  0.375 x 2 = 0.750

  0.750 x 2 = 1.500

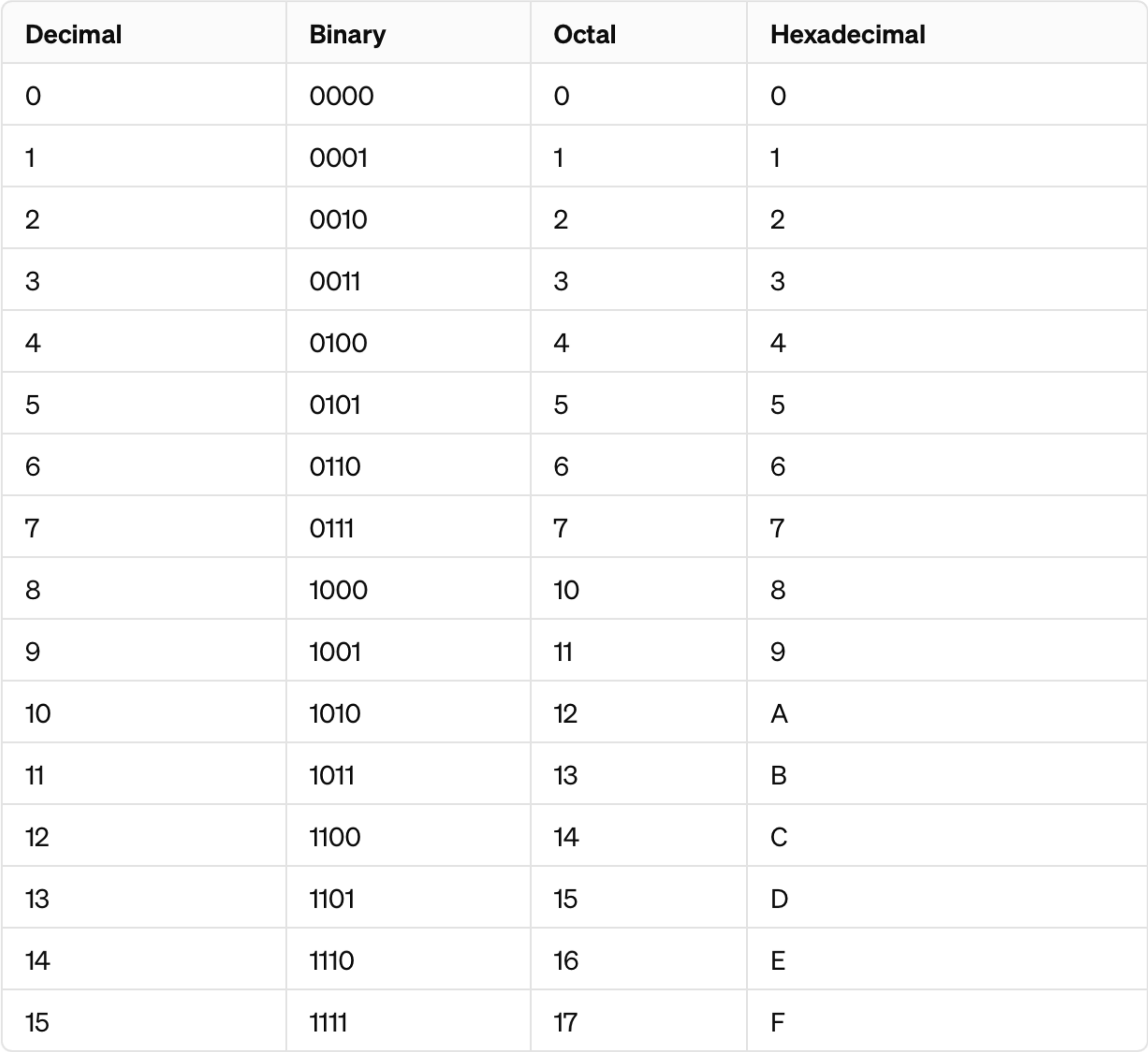
  0.500 x 2 = 1.000

➔ 101111.011 (binary)

Number system conversion

- Other number systems
  - Octal numbers: 0~7
  - Hexadecimal number: 0~9, A~F

  - How to convert
    - hexadecimal to decimal number
      - e.g., AB1 (hexa) ➔ ? (decimal)
    - octal to binary number
      - e.g., 1071 (octal) ➔ ? (binary)



Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Number system conversion

- e.g., AB1 (hexa) ➔ ? (decimal)

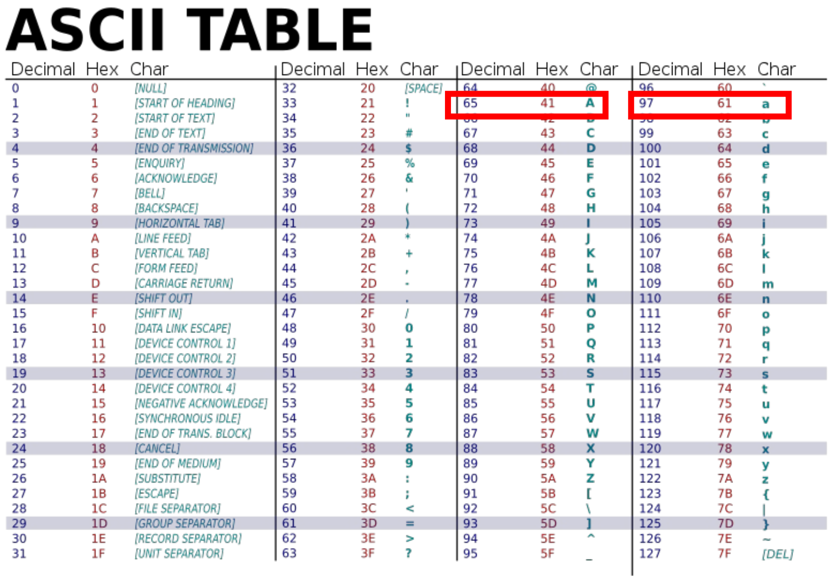
$$A \ * 16^2 = 10 \ * 256 = 2560$$
$$B \ * 16^1 = 11 \ * 16 = 176$$
$$1 \ * 16^0 = 1 \ * 1 = 1$$
$$2560 + 176 + 1 = 2737$$

- e.g., 1071 (octal) ➔ ? (binary)

1 → $(001)_2$
0 → $(000)_2$
7 → $(111)_2$
1 → $(001)_2$
$(001\ 000\ 111\ 001)_2$

Text representation

- ASCII
  - one of the earliest character encoding schemes used by computer
  - uses 7 bits to represent 128 unique characters
- Unicode
  - a comprehensive character encoding standard designed
    to support text written in most of the world's writing systems
  - can represent over 143,000 characters across multiple symbol sets
- UTF-8 (8-bit Unicode Transformation Format)
  - one of several encoding schemes for representing Unicode characters
  - designed to be backward compatible with ASCII and to minimize the file size

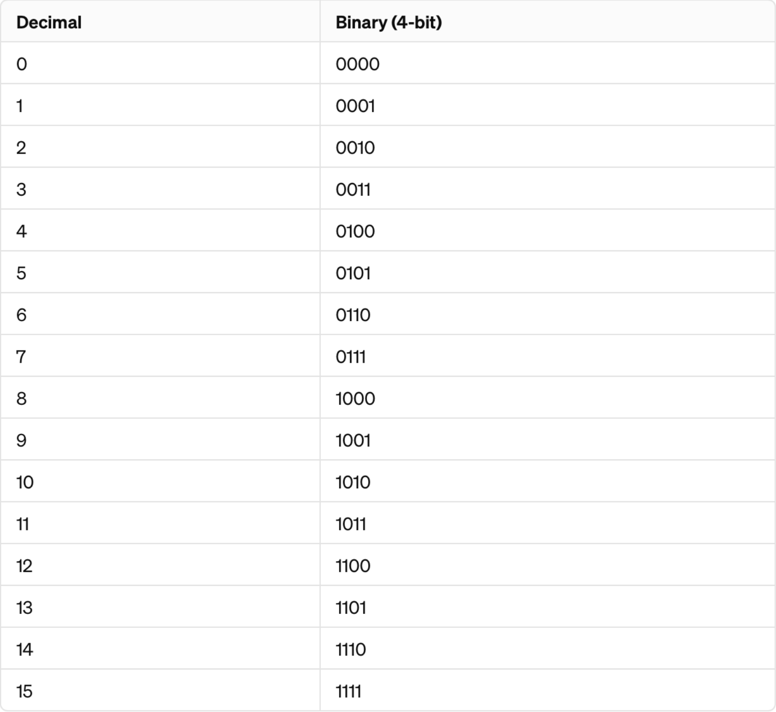


ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Complement (보수)

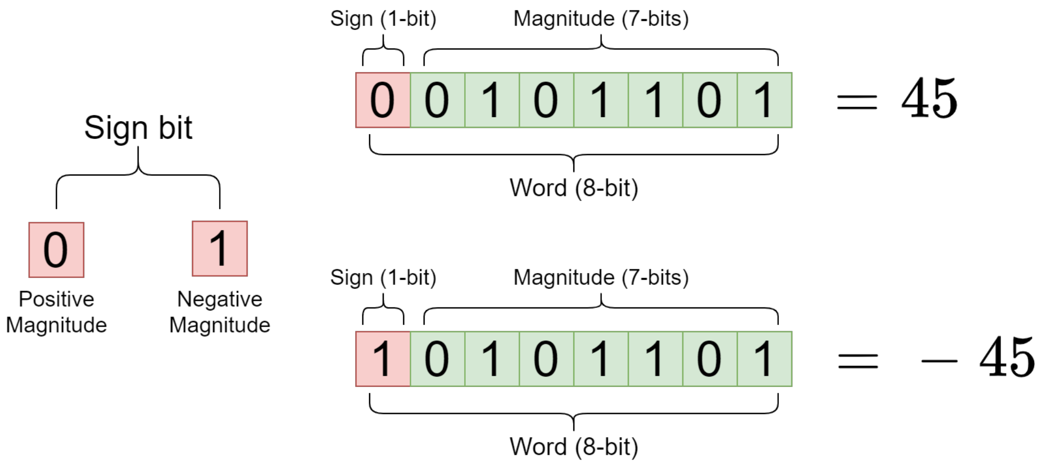
- What is complement?
  - a concept of number representation, especially negative numbers
    - to simplify the operations of subtraction
    - to perform arithmetic operations on negative numbers
    - to facilitate error detection and correction
  - simple thoughts
    - for 4-bits; we can represent the decimal numbers 0 to 16
    - how can we represent the negative numbers?

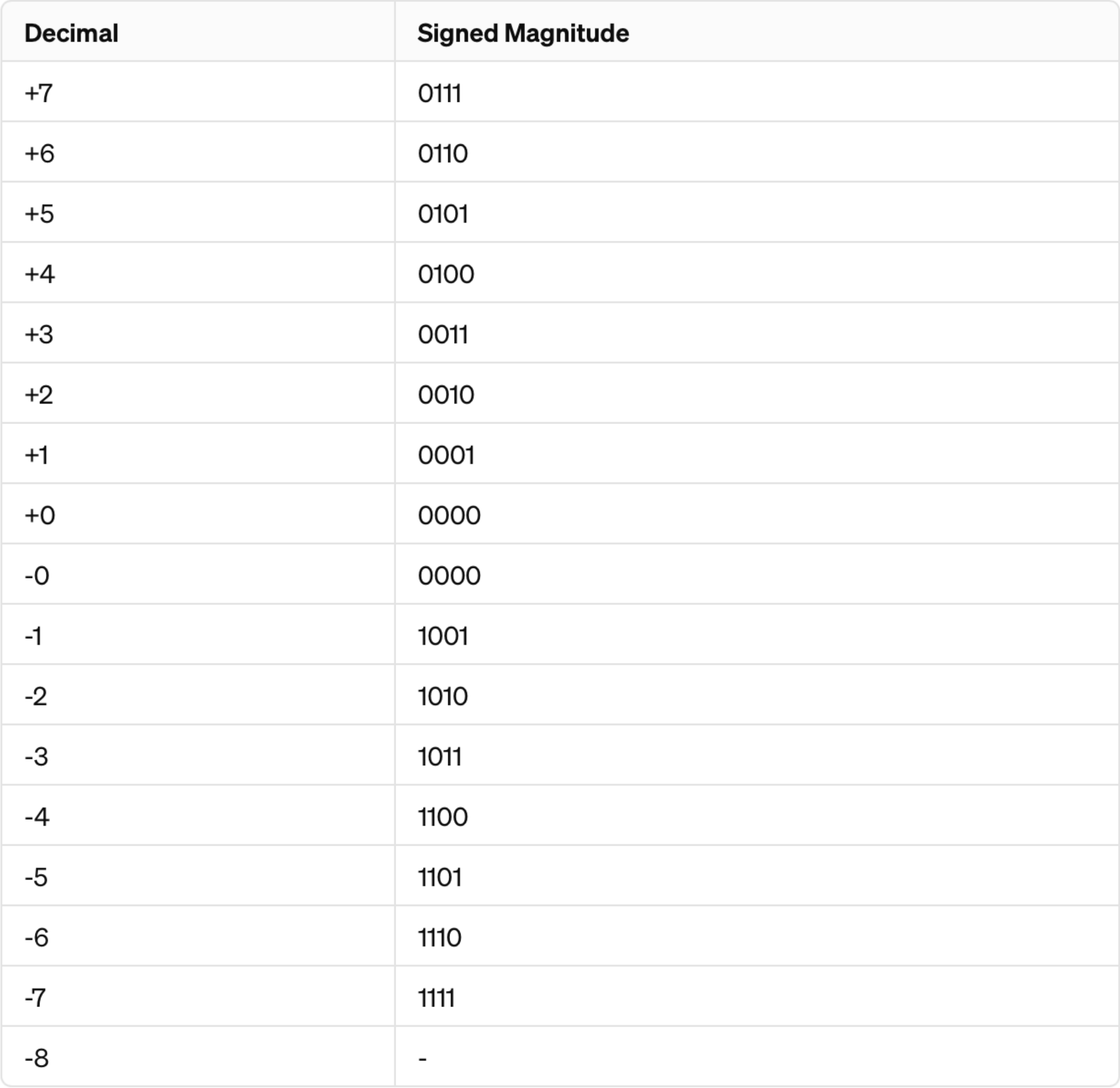


Decimal	Binary (4-bit)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Signed magnitude

- Idea to represent negative number
  - the first bit is reserved for “sign”
    - 0 ➔ positive, 1 ➔ negative
  - possible range +7 to -7

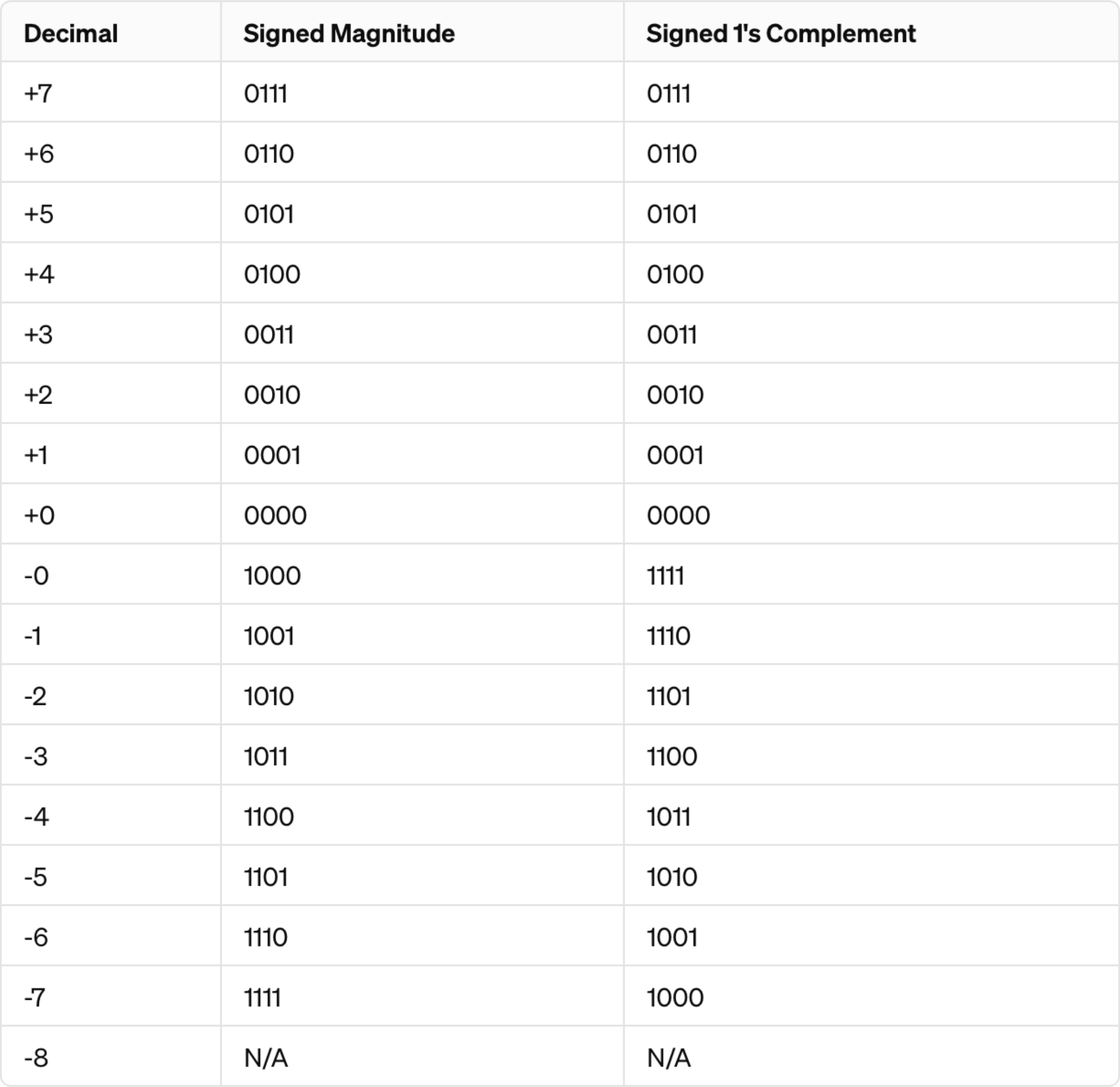




Decimal	Signed Magnitude
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000
-0	0000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111
-8	-

1's complement

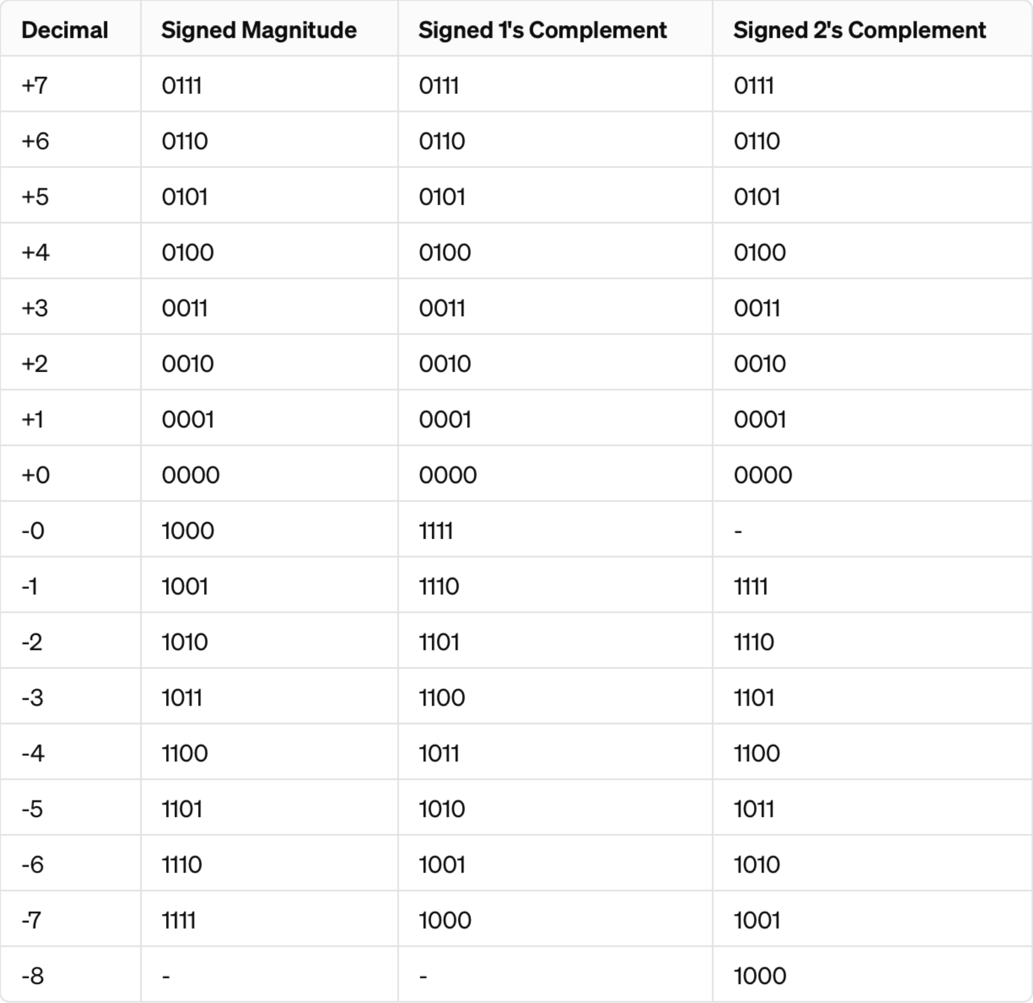
- Idea of 1's complement
  - the first bit is reserved for "sign"
    - 0 ➔ positive, 1 ➔ negative
  - invert all the bits in the number
    - converting all 1s ➔ 0s and all 0s ➔ 1s for negative number representation
    - e.g., 1011001 ➔ 0100110



Decimal	Signed Magnitude	Signed 1's Complement
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
+0	0000	0000
-0	1000	1111
-1	1001	1110
-2	1010	1101
-3	1011	1100
-4	1100	1011
-5	1101	1010
-6	1110	1001
-7	1111	1000
-8	N/A	N/A

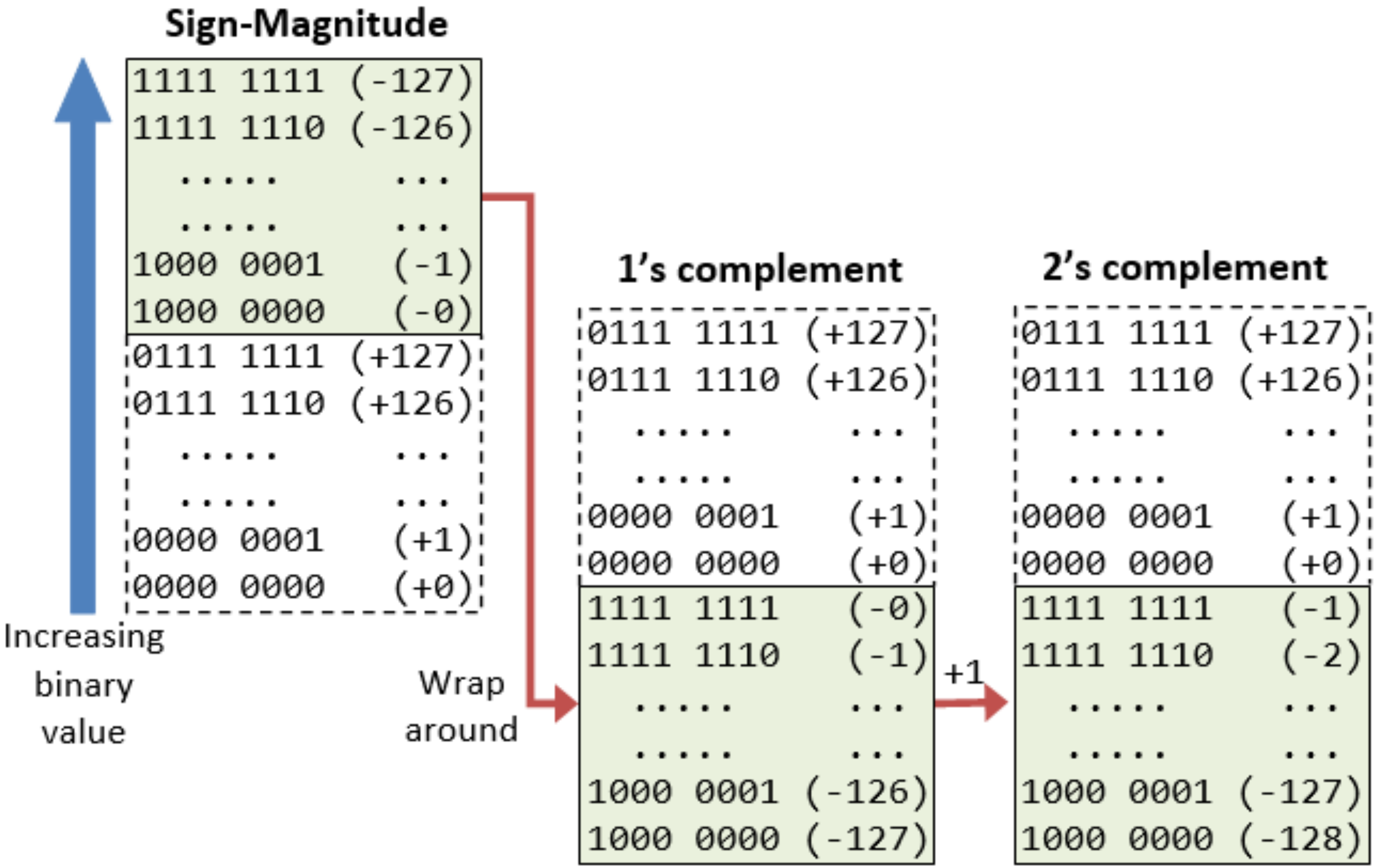
2’s complement

- Idea of 2’s complement
  - the first bit is reserved for “sign”
    - 0 ➔ positive, 1 ➔ negative
  - invert all the bits in the number
    - converting all 1s ➔ 0s and all 0s ➔ 1s for negative number representation
  - add 1 to the 1’s complement of a number
    - e.g., 1011001 ➔ 0100110 ➔ 0100111



Decimal	Signed Magnitude	Signed 1's Complement	Signed 2's Complement
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	1000	1111	-
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	-	-	1000

Summary



Sign-Magnitude

```
1111 1111 (-127)
1111 1110 (-126)
  .....     ...
  .....     ...
1000 0001   (-1)
1000 0000   (-0)
0111 1111 (+127)
0111 1110 (+126)
  .....     ...
  .....     ...
0000 0001   (+1)
0000 0000   (+0)
```

1's complement

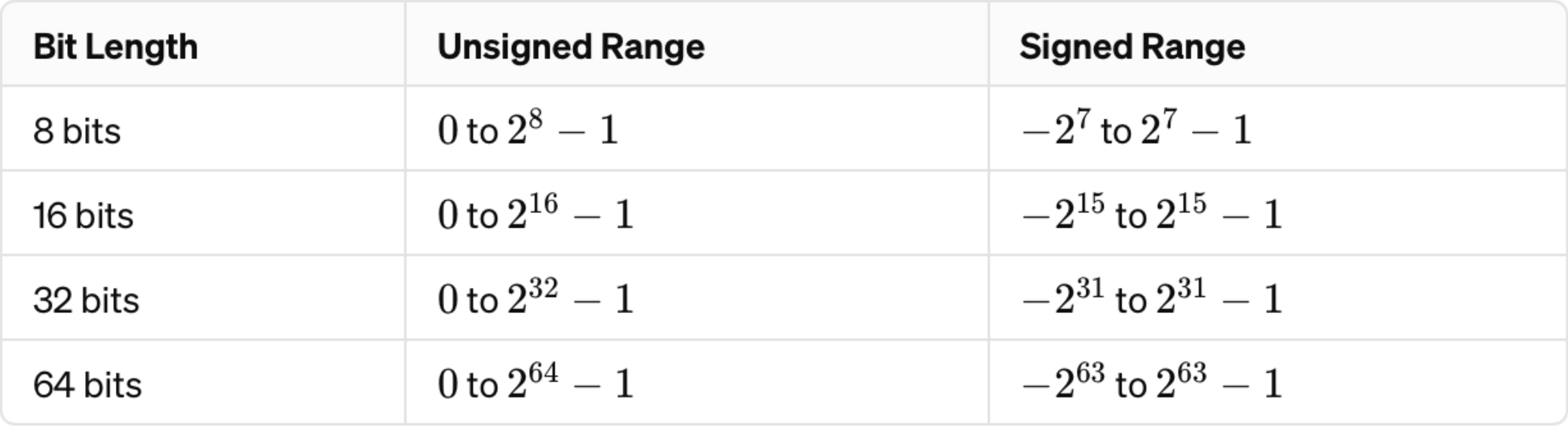
```
0111 1111 (+127)
0111 1110 (+126)
  .....     ...
  .....     ...
0000 0001   (+1)
0000 0000   (+0)
1111 1111   (-0)
1111 1110   (-1)
  .....     ...
  .....     ...
1000 0001 (-126)
1000 0000 (-127)
```

2's complement

```
0111 1111 (+127)
0111 1110 (+126)
  .....     ...
  .....     ...
0000 0001   (+1)
0000 0000   (+0)
1111 1111   (-1)
1111 1110   (-2)
  .....     ...
  .....     ...
1000 0001 (-127)
1000 0000 (-128)
```

Summary

- Range of the numbers by bit length for unsigned and signed



Bit Length	Unsigned Range	Signed Range
8 bits	0 to $2^8 - 1$	-2^7 to $2^7 - 1$
16 bits	0 to $2^{16} - 1$	-2^{15} to $2^{15} - 1$
32 bits	0 to $2^{32} - 1$	-2^{31} to $2^{31} - 1$
64 bits	0 to $2^{64} - 1$	-2^{63} to $2^{63} - 1$

End of slide