School of Computer Science

UNIVERSIDAD DE OVIEDO

Lesson 1: Introduction

Introduction to Programming

Academic year 2023-2024

Concepts (I)

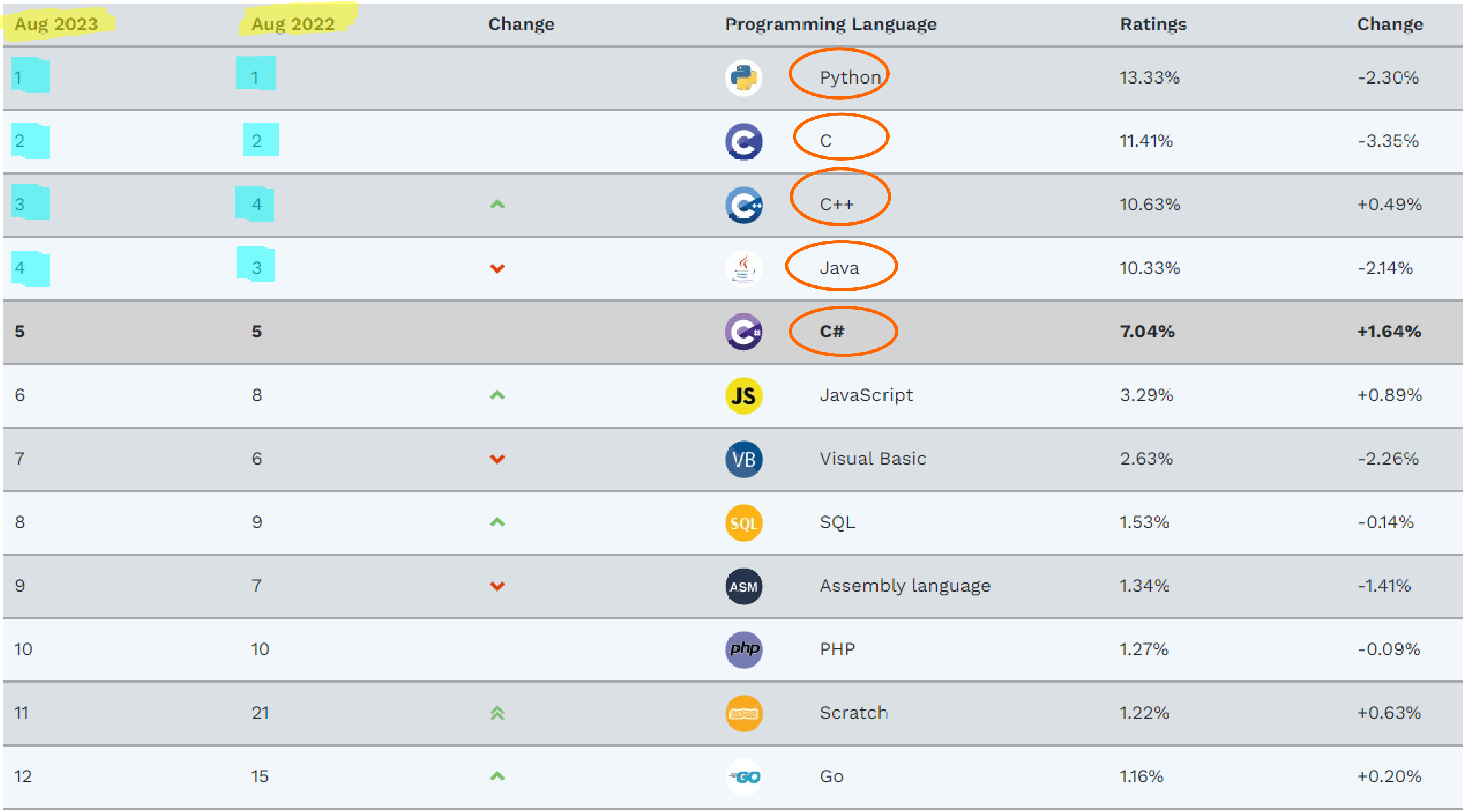
- What is a **language**?
  - *A set of **signs** and **rules** that allows us to communicate with a computer.*
- What is **programming**?
  - ***Programming** consists in writing programs to solve problems by means of computers*.
- What is **software**?
  - The set of **programs**, **instructions** and **rules** to execute certain tasks in a computer.

Types of languages

- **High level**
  - **Language** which allows us to communicate with a computer using signs close to those in a natural language.
- **Assembly**
  - **Language** very similar to machine code, including subtle mnemotechnic modifications which ease its use. It is the immediate upper level to machine code.
- **Machine code**
  - Set of **coded instructions** which a computer can interpret and run straightforward.

Programming Community Index for August 2023

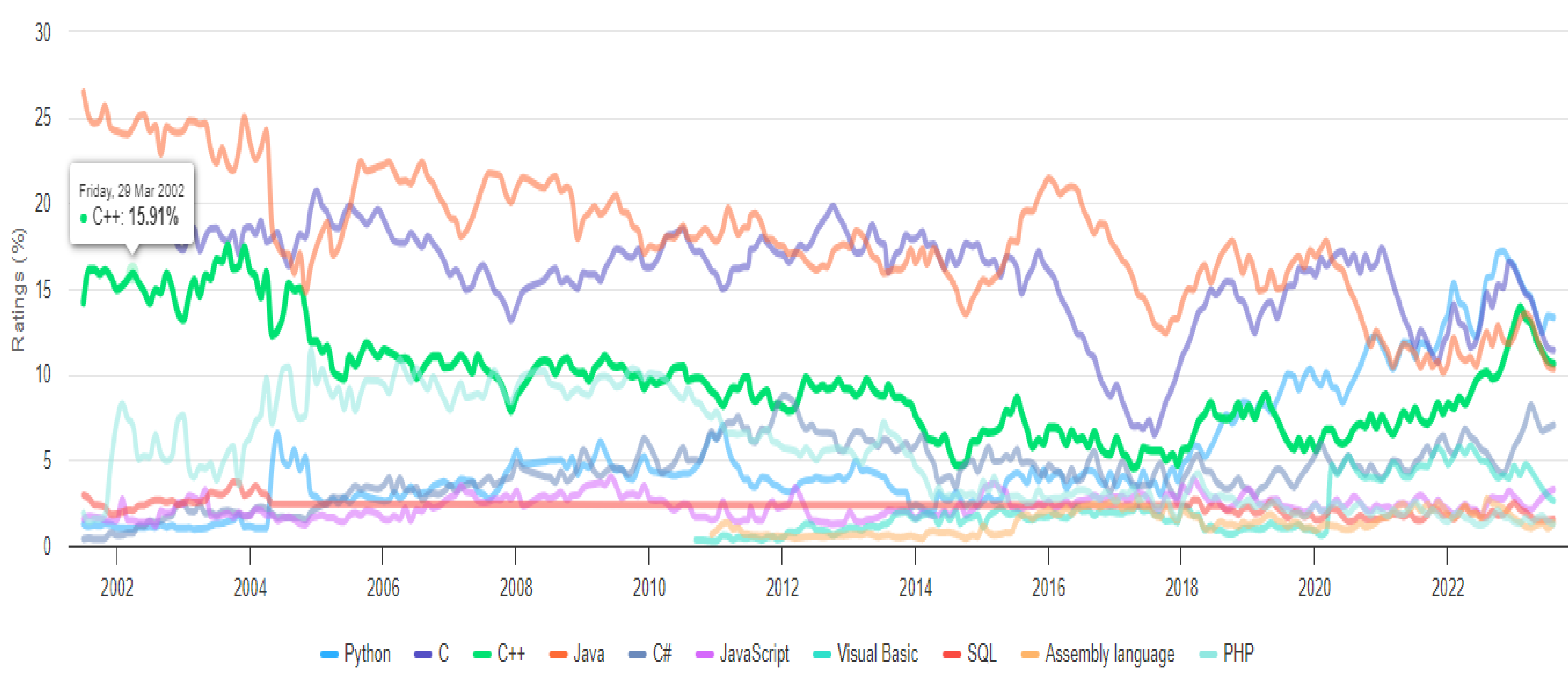
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html



Aug 2023	Aug 2022	Change	Programming Language	Ratings	Change
1	1		Python	13.33%	-2.30%
2	2		C	11.41%	-3.35%
3	4	↑	C++	10.63%	+0.49%
4	3	↓	Java	10.33%	-2.14%
5	**5**		**C#**	**7.04%**	**+1.64%**
6	8	↑	JavaScript	3.29%	+0.89%
7	6	↓	Visual Basic	2.63%	-2.26%
8	9	↑	SQL	1.53%	-0.14%
9	7	↓	Assembly language	1.34%	-1.41%
10	10		PHP	1.27%	-0.09%
11	21	⇈	Scratch	1.22%	+0.63%
12	15	↑	Go	1.16%	+0.20%

Programming Community Index for August 2023

http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html



History of OO languages (I)

- SIMULA (Dahl y Nygaard, 1964):
  - Start of OOP
  - It was conceived as a simulation language
  - Highly influential on the development of other OO languages.
- SMALLTALK (Kay, Goldberg, Ingalls, 1972)
  - “Pure” OO programming language.
- Objective-C (Brad Cox, 1980)
  - Superset of C with an object model similar to that of Smalltalk.
  - Main programming language for Apple’s OS X.

History of OO languages (II)

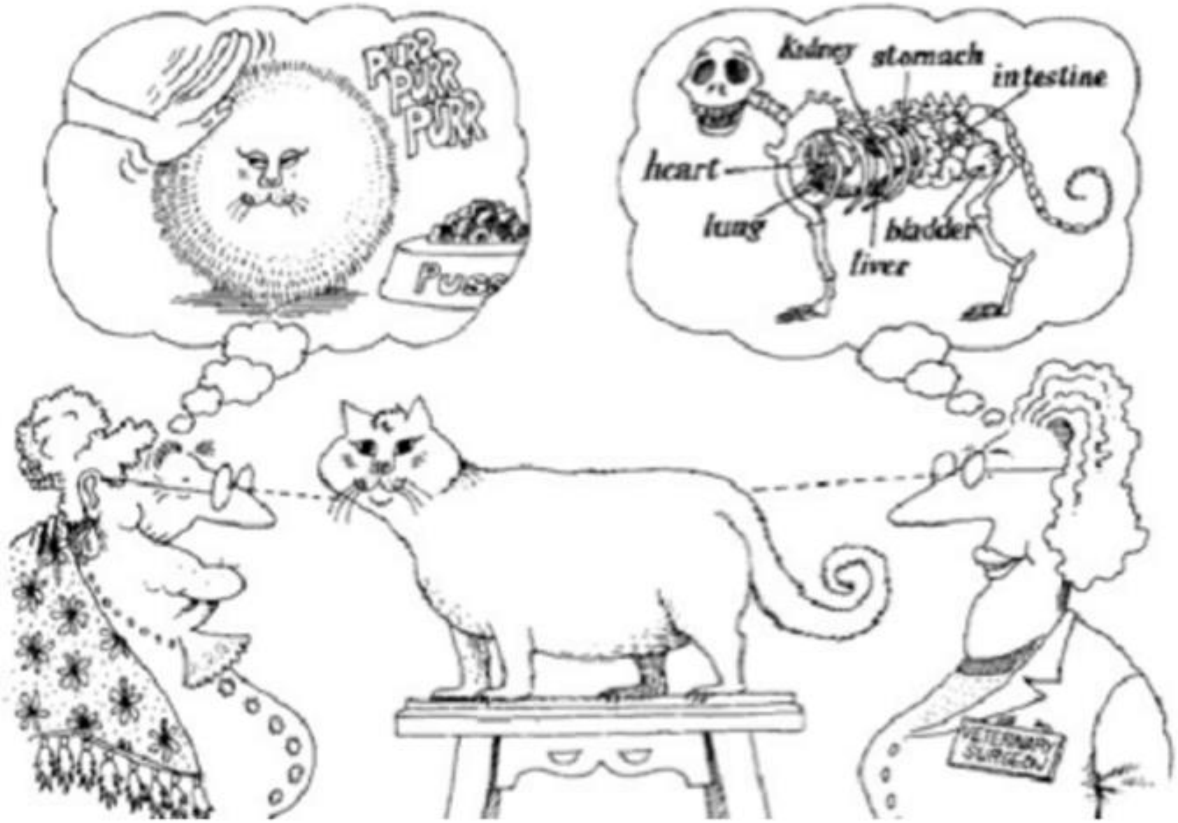
- C++ (Stroustrup, 1985)
  - “Hybrid” OO programming language.
  - Enhancement of C with OO features.
  - Very popular. It helped to spread OOP.
- Eiffel (B. Meyer, 1985)
  - Another “pure” OO programming language.
- Java (SUN, 1995)
  - Similar syntax to that of C++
- C# (Microsoft, 2000)
  - An evolution of C and C++, with important improvements and innovations

Object Oriented Programming

- What is **Object Oriented Development**?
  - An approach to software based on **abstractions** from the real world.

Abstraction

[Booch'96] *Booch, G. Object-Oriented Analysis and Design with Applications. Addison-Wesley. 1996*



"Abstraction focuses upon the essential characteristic of some object, ***relative to the perspective of the viewer****."*

Object Oriented Programming

- What does **Object Orientation** mean?
  - Software is organized as a *collection* of **objects** containing **properties** (*attributes*) and **behaviour** (*methods*).

Object

Properties

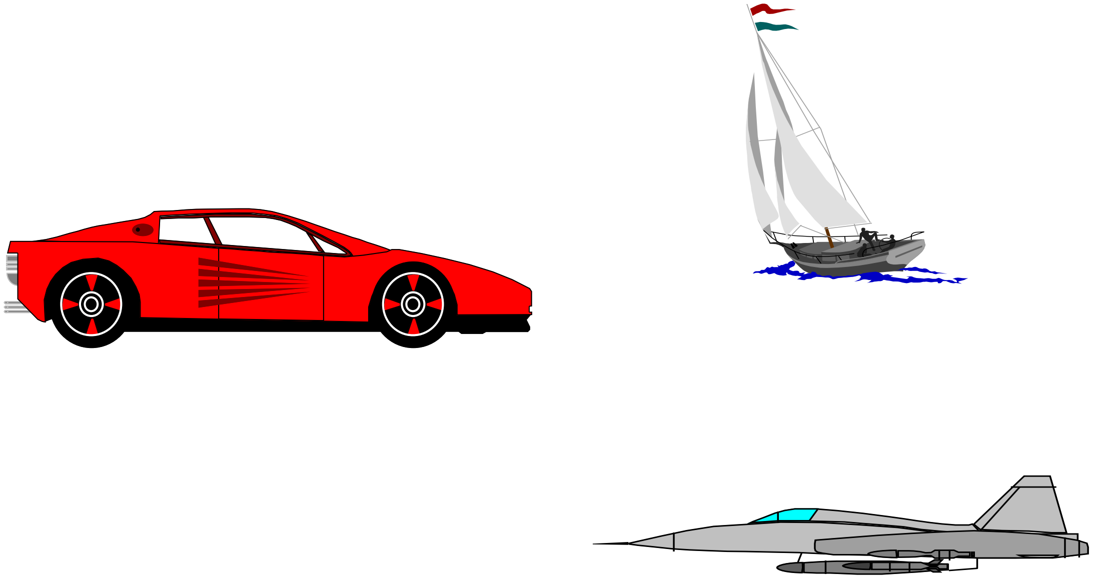
- Color
- Speed
- Size
- Fuel
- …

Behavior

- Stop
- Turn right
- Turn left
- Start
- …

Objects

Objects with similar **properties** and the same **behaviour** are organized into **classes.**



Objects from the **class** **vehicle**

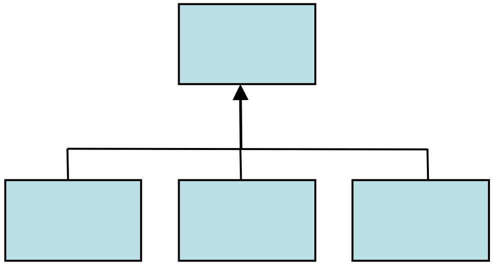
Concepts (I)

- **Domain**
  - Real or imagined context of a given activity.
- **Problem domain**
  - Set of elements relevant to a given problem.
- **Modelling**
  - Representing something real or imagined.
- **Model**
  - A representation for the problem domain. The elements in the model are called **objects**.

Concepts (II)

Real world

Software



PROBLEM DOMAIN

Abstraction

MODEL

In short

- When writing a **program** in an OOPL we are **creating a model** from the real world.
- The elements in the model are the **objects** belonging to the problem domain.
- The **objects** can be organized into **categories**.
- A **class** describes, in an abstract form, all of the objects of a given type.

Example

Modelling a traffic simulation

- One entity type in this problem is car
- What’s car in such a context? **A class or an object?**
- Questions:
  - What colour is it?
  - What speed can it reach?
  - Which is its license plate number?
  - …

Solution

- You cannot answer those questions without referring to a given particular car.

- In this context we are talking about cars in general, not any specific car. Thus, car is referring to a "car class".

- If I say "*I have an old car in my garage*", I could answer the questions:
  - It is red.
  - Top speed is 100 km/h
  - Its license plate is 717-066

Instance

- When we are referring to a **given object** we are talking about an **instance.**
- **Instance** and **object** are near synonyms.
- We tend to refer to objects as instances when we want to emphasize they belong to a given class.

“this object (Porsche Carrera GT)
is an instance of the
car class”

Attributes

- **Attributes** are features of an object (those we want to model) which can adopt different values.
  - Which features would have a car if we wanted to develop a game for 2 to 3 year-old children?
  - Which features would have a car if we wanted to develop a game for teenagers?
- Attributes change depending on the model we want to create.

Methods

- **Actions** which can be demanded from an object from a given class (i.e. "stuff" they can do).
  - They define a ***behaviour***.
- **Example.** We want to draw different figures in a window.
- For a circle object…

Methods

Move up
Move down
Move right
Move left
…

Attributes

Diameter
Colour
X position
Y position
…

Origins of Java

- 1991. James Gosling starts a project to write platform independent code. He tries with C++ but he is not happy with it
- 1993. A new language, Oak, is developed. It is similar to C++ but portable across different platforms
- 1994. The first graphical Web browsers are developed. The interest on Internet grows
- 1995. Oaks changes into Java
- 1997. The first commercial version is produced: JDK1.1
- 1998. Sun releases the platform Java 2
- 2000. Sun releases version 1.3
- 2002. Sun releases version 1.4
- 2004. Sun releases version 1.5 also known as Java 5.0
- 2006. Sun releases version 1.6 also known as Java 6.0
- …
- 2021. Java SE 16 (Oracle)
- 2022. Java SE 18 (Oracle)

Java technology

- Programming language
  - Object oriented.
  - Syntax similar to C/C++.
- Platform
  - A platform is the base defining a **hardware** on top of which a **software** can be developed and run.
- **The Java platform** is based on the concept of an **abstract machine**
  - A computer designed without the purpose to be hardware-implemented.
- A **virtual machine** is the implementation of an abstract machine.

- Java Virtual Machine (JVM).
- Libraries or API (Application Programming Interface).

JRE

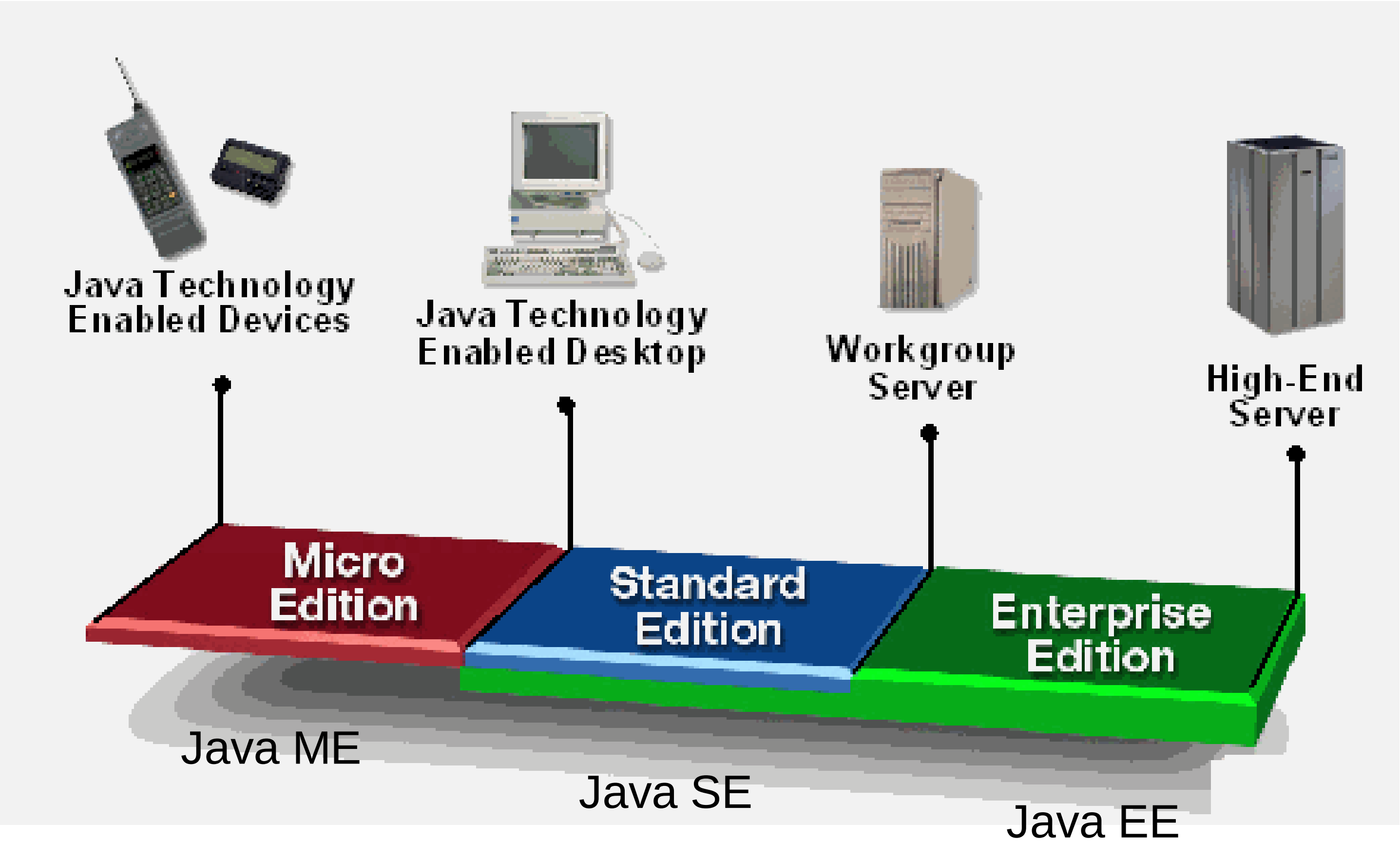
Java technology (I)

- A platform is the Programming language
  - Object oriented.
  - Syntax similar to C/C++.
- Platform
  - A platform is the base defining a **hardware** on top of which a **software** can be developed and run.
- **The Java platform** is based on the concept of an **abstract machine**
  - A computer designed without the purpose to be hardware-implemented.
- A **virtual machine** is the implementation of an abstract machine.

- Java Virtual Machine (JVM).
- Libraries or API (Application Programming Interface).

JRE

Java platform



Java ME

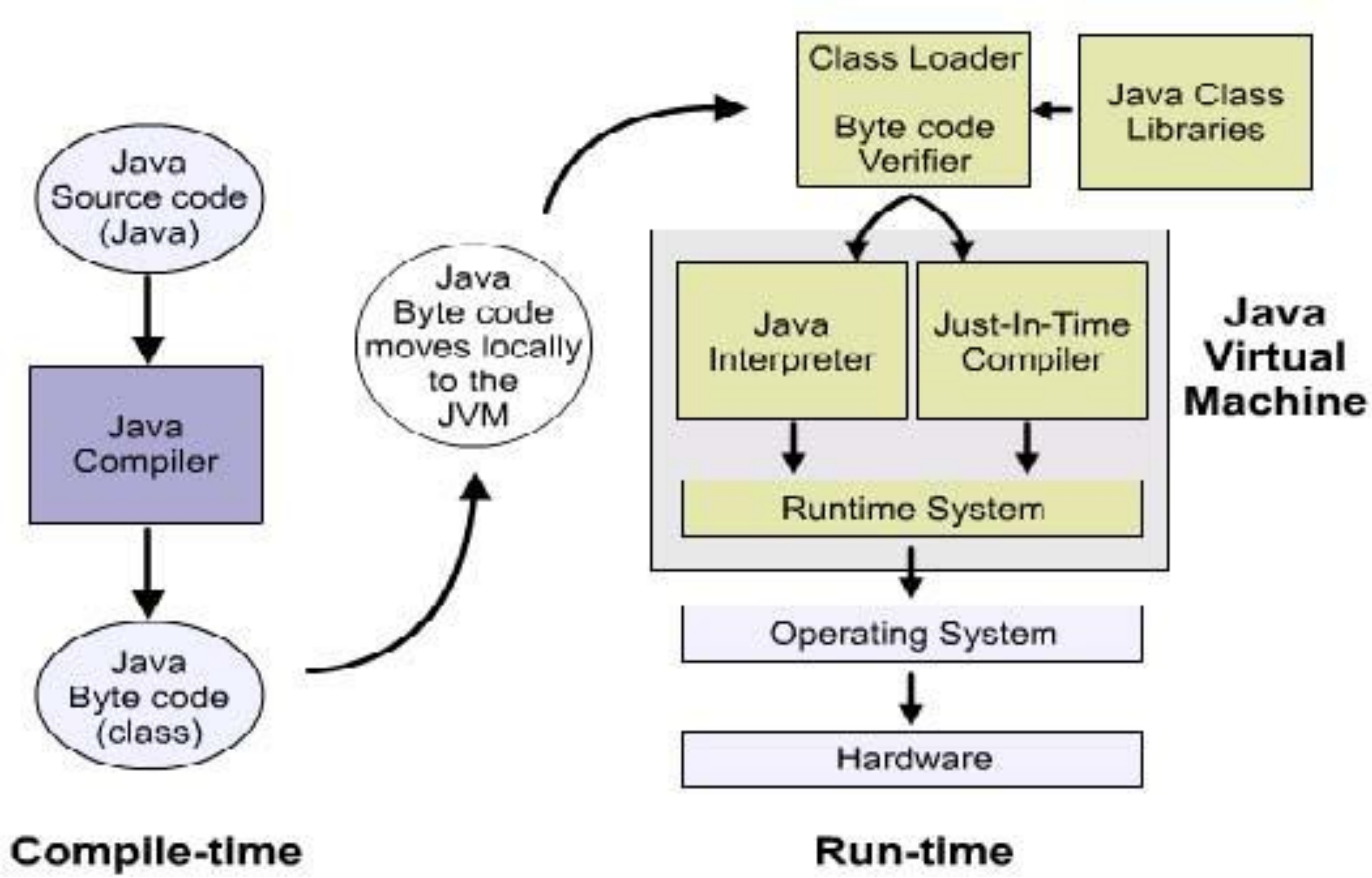
Java SE

Java EE

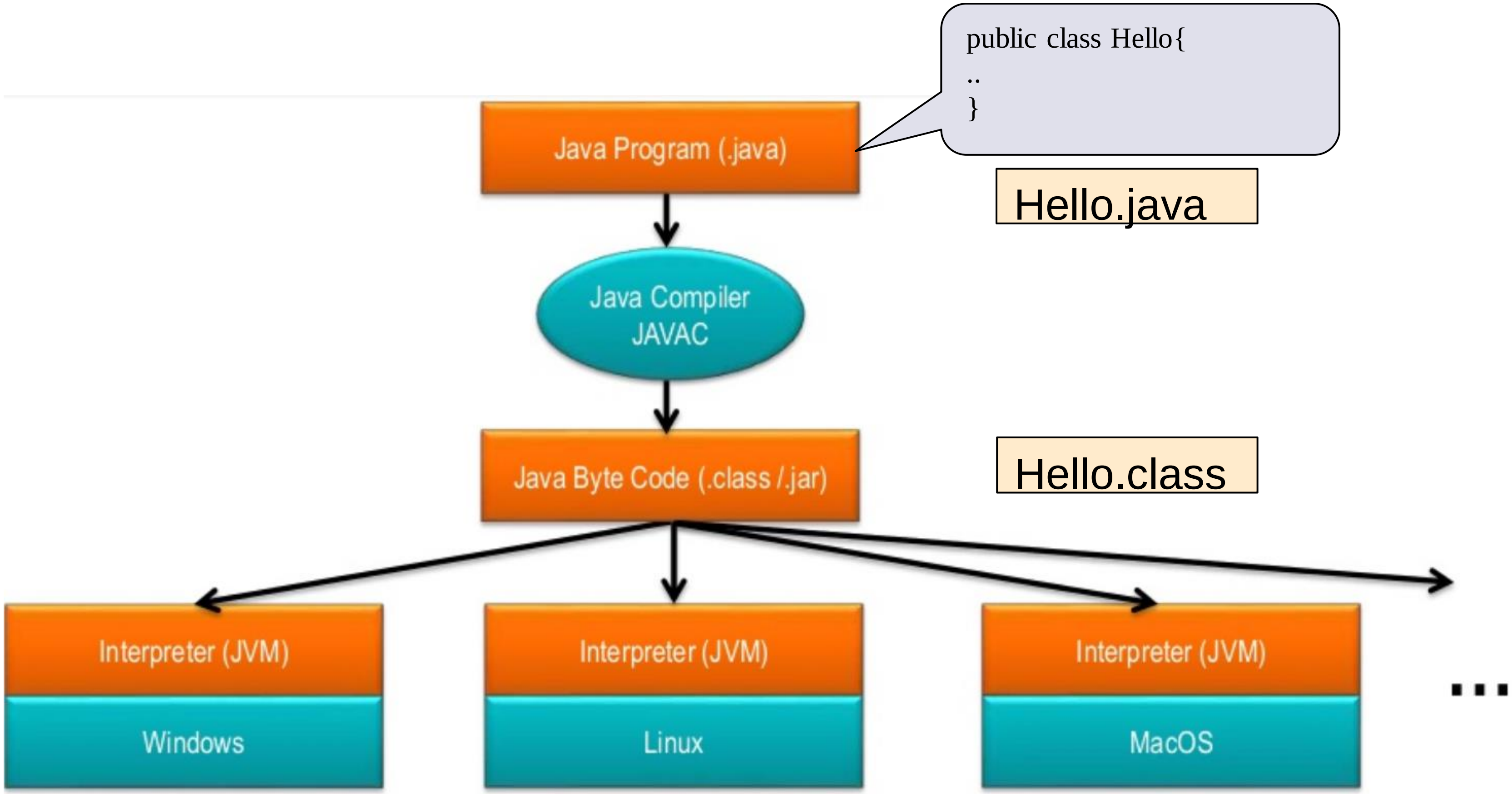
Platform versions

- The Java platform has three standard versions:
  - **JME** (*Java Micro Edition*): Platform for limited resources devices (PDAs, cell phones, GPS navigation devices, etc.)
  - **JSE** (*Java Standard Edition*): Platform to develop desktop applications and applets
  - **JEE** (*Java Enterprise Edition*): Platform to develop enterprise/industrial applications (usually on servers)
- Each one is distributed with a JVM implementation and a set of classes providing an API.

Development process in Java



Platform independence

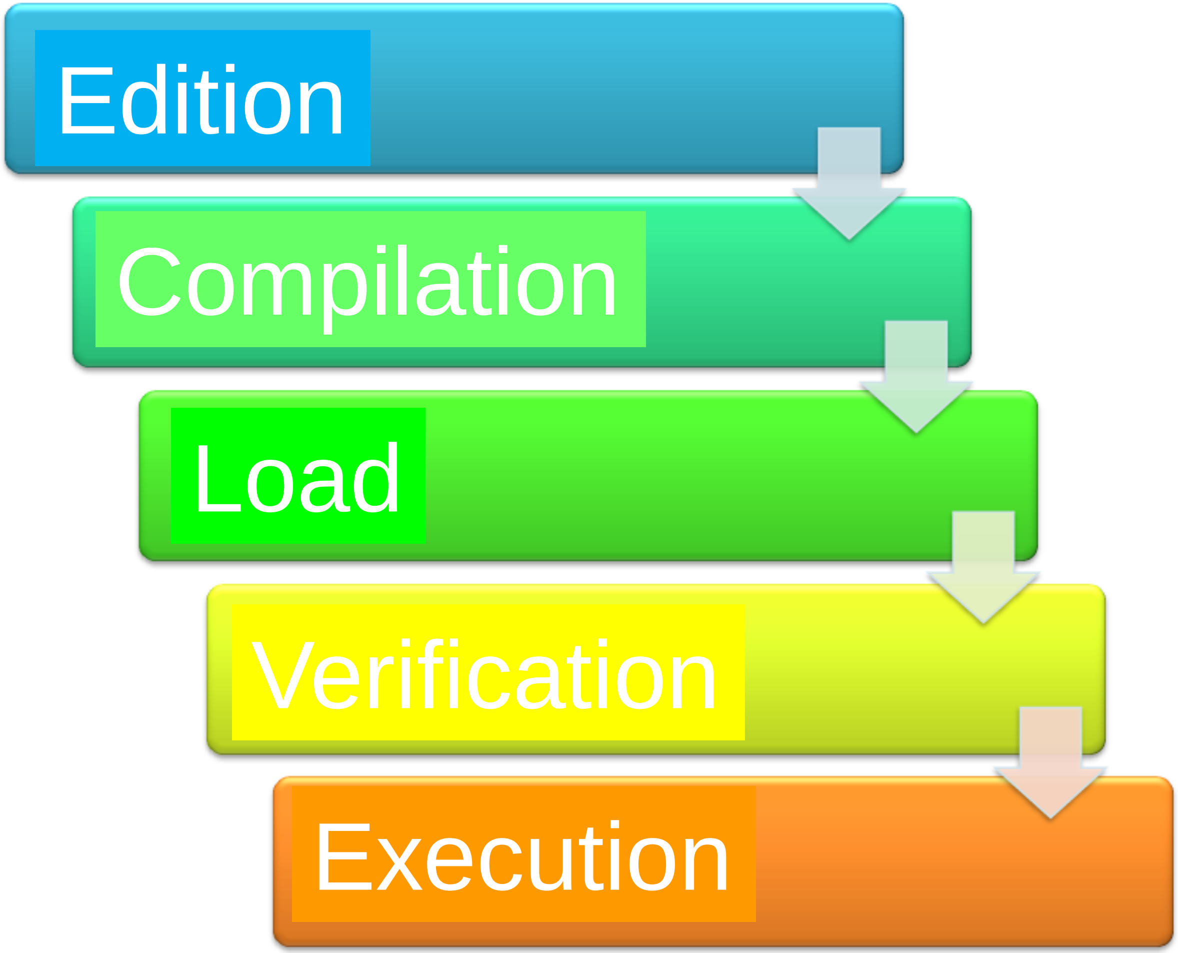


```
public class Hello{
..
}
```

Hello.java

Hello.class

Development phases



Edition

Compilation

Load

Verification

Execution

Development phases

Phase 1: Edition

The program is written with an **editor** and saved to disk into a **.java** file.

Development phases

Phase 2: Compilation

The **compiler** (**`javac`**) translates the Java source code into **bytecode** and saves it to disk into a **.class** file.

Development phases

Phase 3: Load

To be executed, the code must be loaded into main memory. The **class loader** reads the **bytecode** from the **.class** files into the JVM **memory**.

Development phases

Phase 4: Verification

The **bytecode verifier** checks the bytecode is “safe” before being executed. That is, that the code complies with Java safety policies.

Development phases

Phase 5: Execution

The **JVM** runs the program **bytecode**.

Java features (I)

- Simple
  - The language was designed to simplify the tasks for the programmers and to allow them to use the language efficiently.
- Object oriented
  - Everything can be modelled as an object and Java focuses on the creation and management of objects.

Java features (II)

- Distributed
  - Java provides libraries and tools to allow programs to be distributed, that is, they can run and interact from different machines (e.g. the API includes TCP/IP, HTTP, FTP, etc.)
- Robust
  - Java provides several checkings both in compile and run time (memory management and improperly treated exceptions).
  - Data type checking makes it easy to detect errors.

Java features (III)

- Safe
  - Java allows us to develop virus-free programs.
  - When downloading a Java applet and running it in our PC there is no way it can harm the system because Java provides a number of safety mechanisms to protect the host system.
- Portable
  - By using Java, developers can just write a single version of their program to be run in every platform with a JVM.

Java features (IV)

- High performance
  - Java provides JIT (Just in Time) compilers which allow Java programs (which are platform independent) to run with a performance close to those programs originally written in compiled languages.
- Multithread
  - Java has the ability to run different tasks at the same time by means of so-called threads. Synchronized threads are very useful for distributed and network applications.

Some applications developed in Java

- Web applets
- OpenOffice
- Many IDEs (Eclipse, BlueJ, …)
- Android apps.
- Plenty of web applications (e.g. Liferay, CMS used by UniOvi).
- …

How to create and execute a Java program

First option:

1. Using a text editor to write the Java source code.
2. Using the JDK compiler (javac) to translate the source code into bytecode.
3. Using the runtime (JRE) to execute the program.

How to create and execute a Java program

Second option:

1. Using an Integrated Development Environment (IDE).

- It blends in a single tool the three previous steps.
- In addition to allow you writing the program it provides a way to compile and execute it.
- It usually includes:
  - A text editor.
  - A compiler.
  - An interpreter.
  - A debugger.
  - A version control system (sometimes).
  - Tools to build graphical user interfaces (sometimes).

Available IDEs for Java

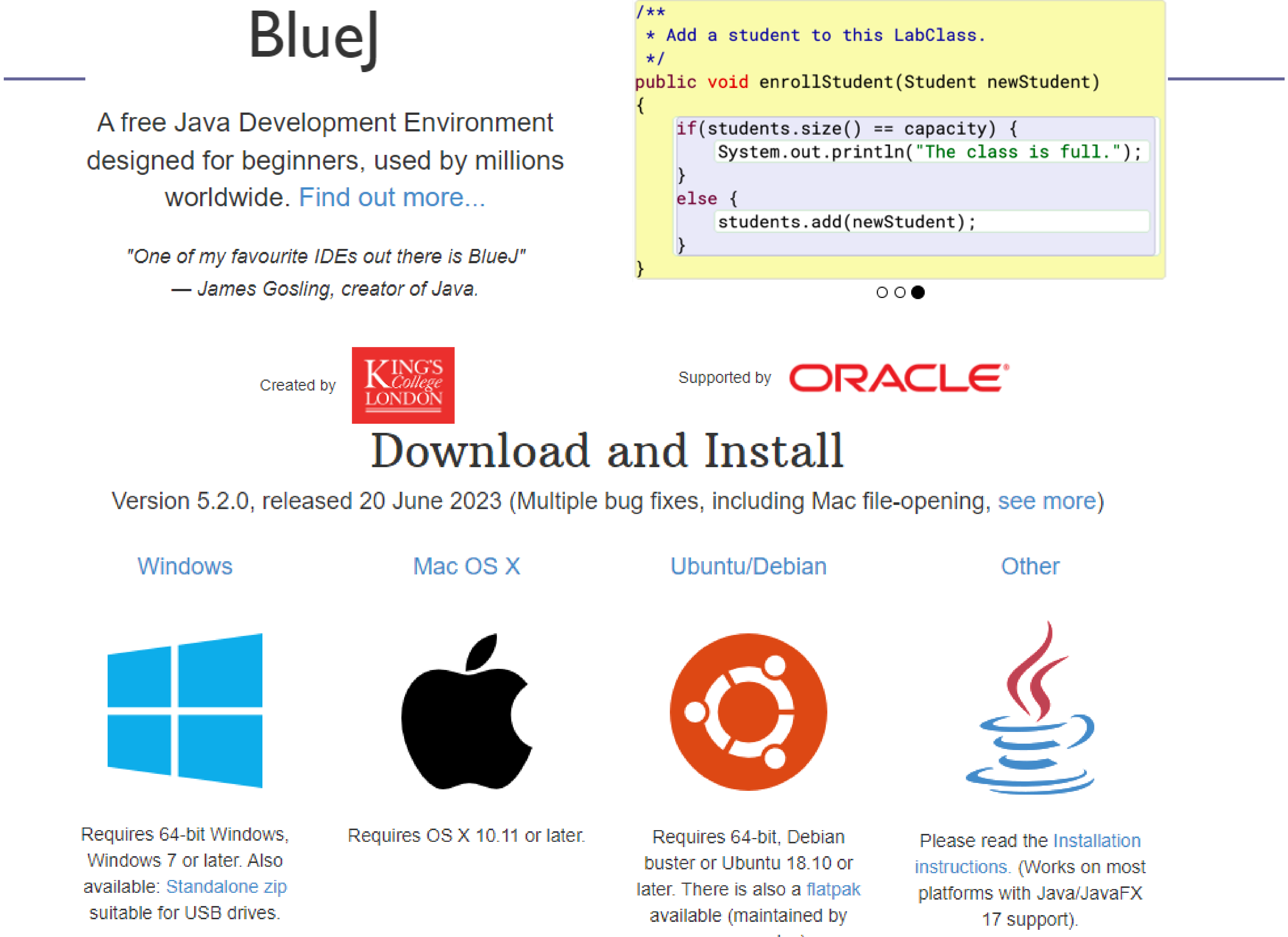
Professional tools: Eclipse, NetBeans, JBuilder, …

Educational tools: **BlueJ (www.bluej.org)**

Others (of course)…

The one
we will use

Downloading BlueJ: http://www.bluej.org/



BlueJ

A free Java Development Environment designed for beginners, used by millions worldwide. Find out more...

"One of my favourite IDEs out there is BlueJ"
— James Gosling, creator of Java.

```
/**
 * Add a student to this LabClass.
 */
public void enrollStudent(Student newStudent)
{
    if(students.size() == capacity) {
        System.out.println("The class is full.");
    }
    else {
        students.add(newStudent);
    }
}
```

○ ○ ●

Created by KING'S College LONDON

Supported by ORACLE

Download and Install

Version 5.2.0, released 20 June 2023 (Multiple bug fixes, including Mac file-opening, see more)

Windows

Requires 64-bit Windows, Windows 7 or later. Also available: Standalone zip suitable for USB drives.

Mac OS X

Requires OS X 10.11 or later.

Ubuntu/Debian

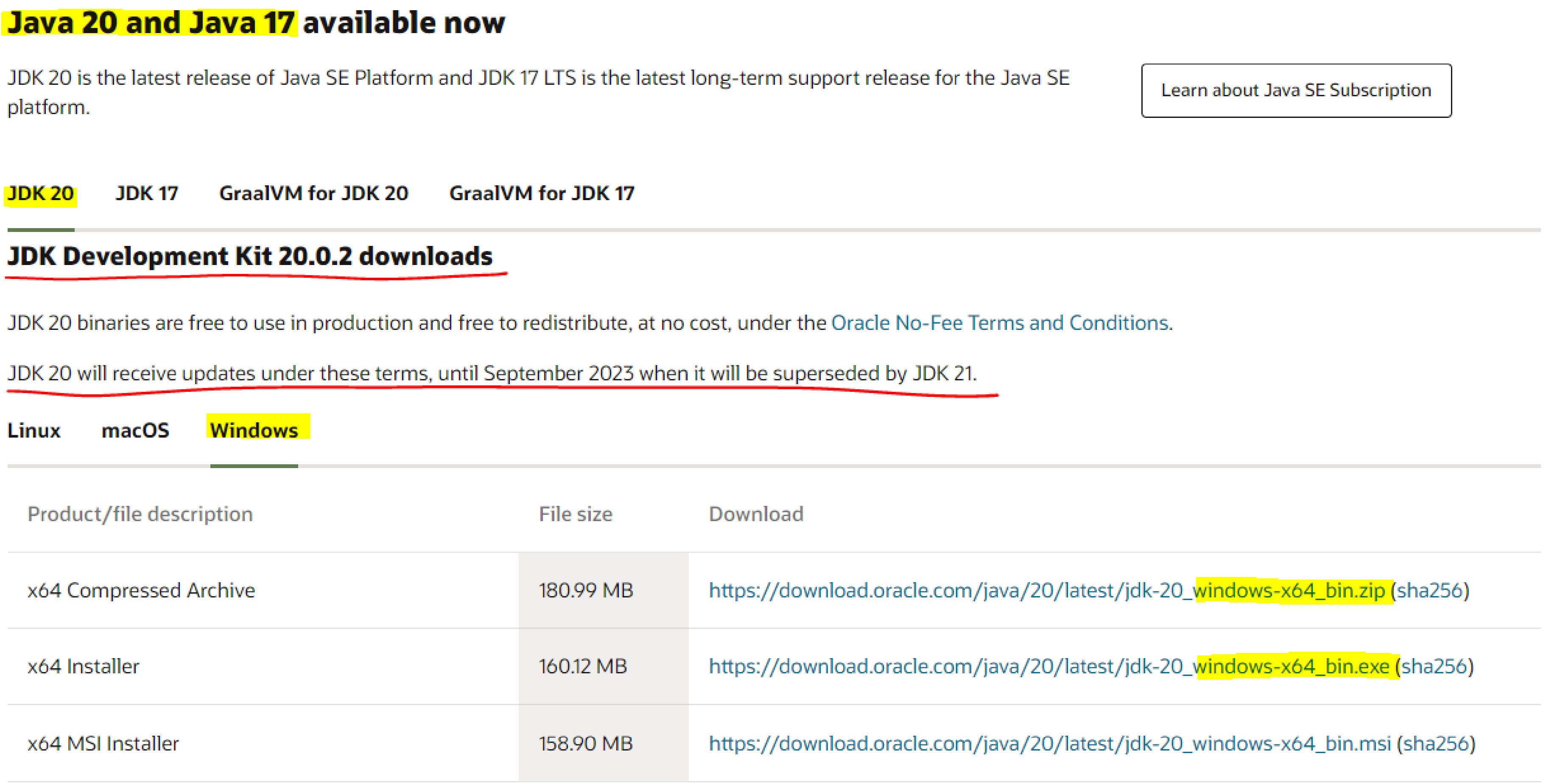
Requires 64-bit, Debian buster or Ubuntu 18.10 or later. There is also a flatpak available (maintained by

Other

Please read the Installation instructions. (Works on most platforms with Java/JavaFX 17 support).

Downloading the JDK (only for linux systems)

https://www.oracle.com/java/technologies/javase-downloads.html



Java 20 and Java 17 available now

JDK 20 is the latest release of Java SE Platform and JDK 17 LTS is the latest long-term support release for the Java SE platform.

JDK Development Kit 20.0.2 downloads

JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions.

JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

Product/file description	File size	Download
x64 Compressed Archive	180.99 MB	https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.zip (sha256)
x64 Installer	160.12 MB	https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe (sha256)
x64 MSI Installer	158.90 MB	https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.msi (sha256)