Translation processes

Camille Duquesne
1ère IB

Any questions/difficulties you would like to share about last session’s content ?

A1 Computer fundamentals

A1.4.1 Evaluate the translation processes of interpreters and compilers.

• The mechanics and use-cases of each translation approach

• The difference in error detection, translation time, portability and applicability for different translation processes, including just-in-time compilation (JIT) and bytecode interpreters

• Example scenarios where the translation method should be considered must include rapid development and testing, performance-critical applications and cross-platform development.

What are fundamental operations of a computer ?

- Add data
- Compare data
- Retrieve data (Load)
- Store data

```
LOAD register 34AB39
ADD 29
STORE result
COMPARE result to
register 4
```

=> These are operations that do not require the processor to go through a large number of sub operations to reach a result.

What are compound operations ?

Compound operations are composed of several fundamental operations together:

- Find the largest number in a collection
- Check if a number is prime
- Rotate an image 90°
- Any method/sub procedure

=> We'll learn more about fundamental operations and compound operations in *A1.2 Data representation and computer logic*

What are programming Languages ?

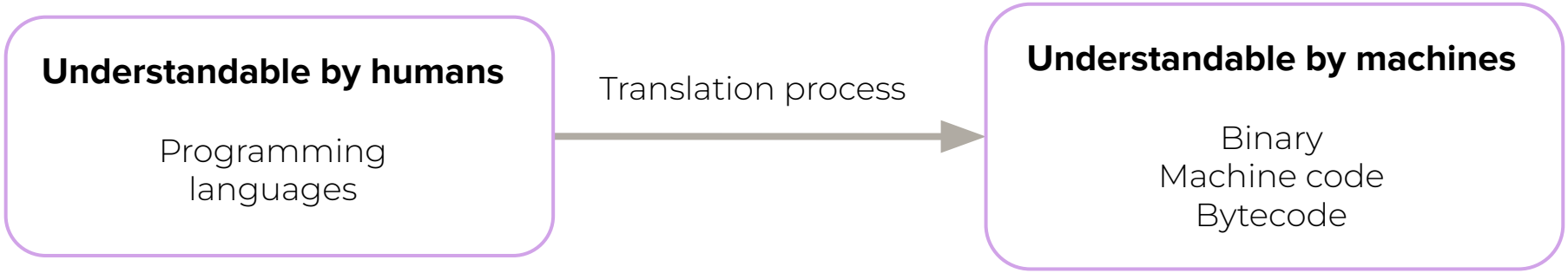
How does a computer language differ from a natural language?

- What programming languages do you know ?
- What do these programming languages have in common ?
- What are differences between these programming languages ?
- Is a programming language directly understood by the computer ?

What was the first computer language invented ?

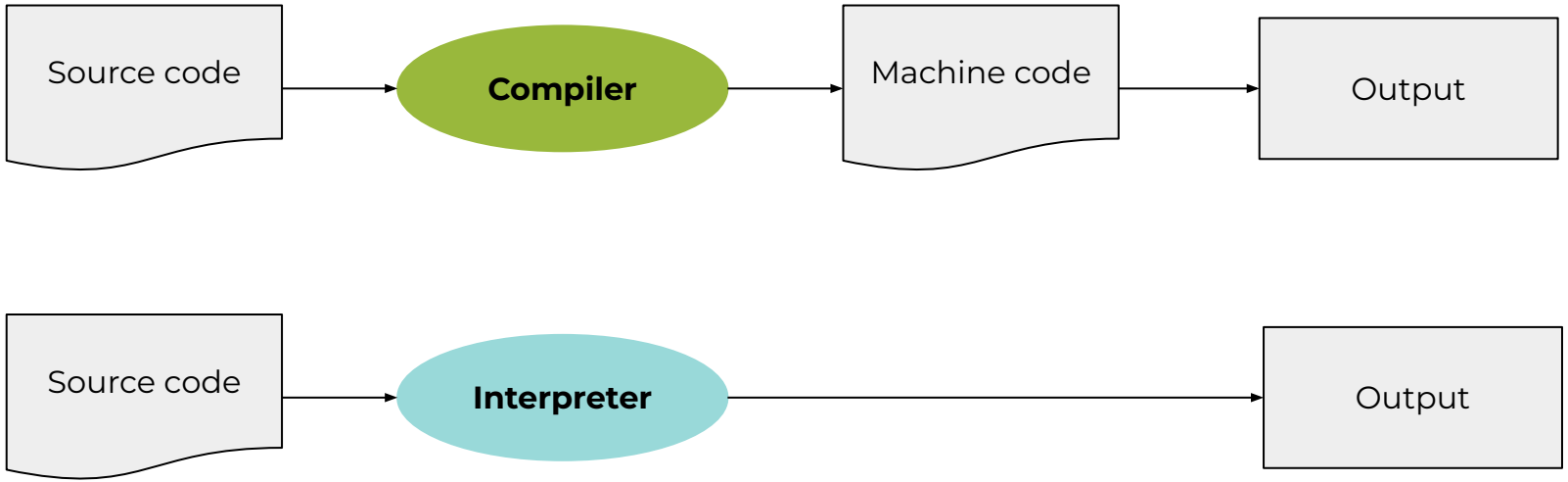
Did java always exist ?
Why are some programming languages more popular than others ?
Is there a “best” programming language ?

Is a programming language directly understood by the computer ?



How do we execute our program ?

How are Higher Level Languages understood by the computer ?



Interpreter vs Compiler

Why not use both ?

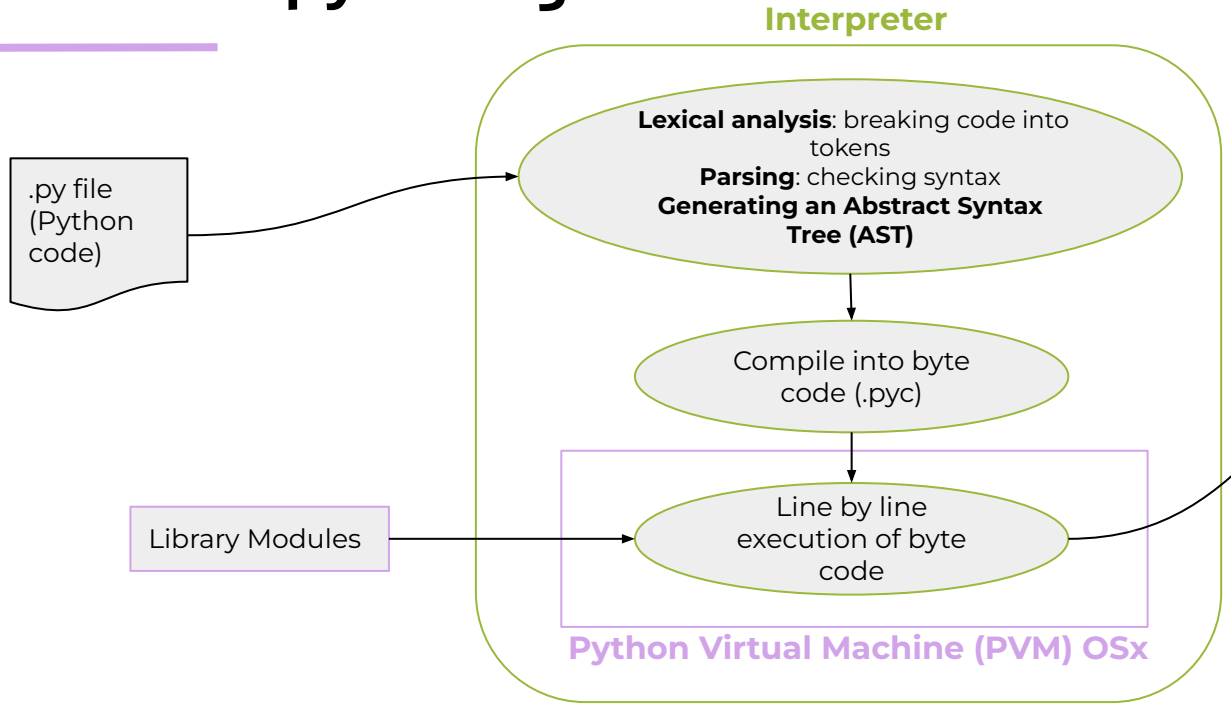
Interpreters:

- Line by line execution
- Slower
- Immediate feedback, useful for quick iteration and debugging
- Platform independent
- Useful for web development and education

Compilers:

- All code is compiled to machine code before being executed
- Compilation step is slower than interpretation
- But faster execution once code is compiled
- Feedback is slower as all errors need to be addressed before execution
- Platform dependent
- Useful for system software, applications, ...
- A little more secure than interpreters

How does python get translated ?



Bytecode is a intermediary machine language (more abstract than machine code). It is run on virtual machines and allows languages to be platform independent

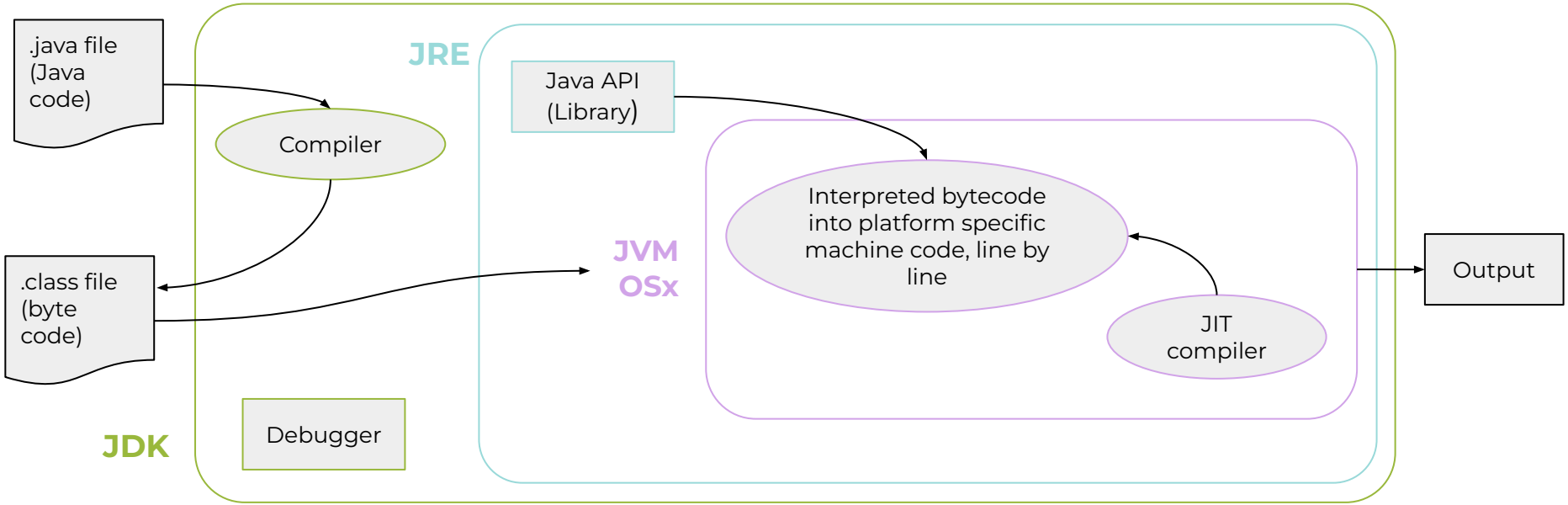
A virtual machine is a software that emulates a specific hardware.

To know more about the python interpreter

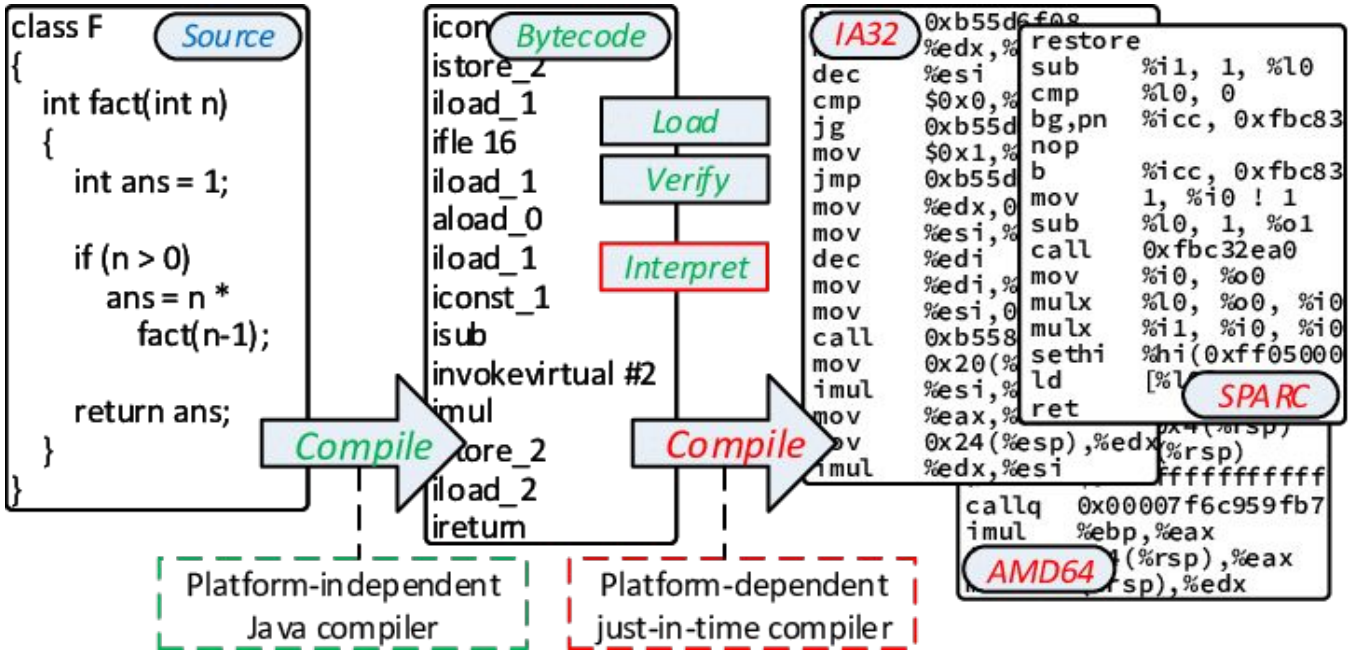
https://www.youtube.com/watch?v=tzYhv61piNY

How does java get translated ?

Just In Time (JIT) compilation the identifies frequently executed sections of the bytecode (called “hot spots”) and compiles them into machine code on the fly. This allows for faster execution as less interpretation is needed.



Java example



https://www.researchgate.net/profile/Michael-Orlov-2/publication/318376879/figure/fig1/AS:544876321751040@1506920046971/Java-source-code-is-first-compiled-to-bytecode-and-subsequently-interpreted-or-executed.png

COBOL the first programming language !

- **COBOL's (Common Business-Oriented Language)** creation is due to **Grace Hopper** who was one of the **first programmers**.
- Grace hopper was the **first one** to create **subroutines** while working on the MARK I computer.
- Grace hopper believed that programming should be simplified with an English-based computer programming language independent of the hardware. **She created the first compiler** that converted English terms into machine code.
- She was met with several **years of resistance to her ideas** but managed to convince business managers and others at the CODASYL consortium where COBOL was born in **1959**.

https://live.staticflickr.com/3118/2782902040_3864311141d_c.jpg

Grace Hopper Next to a UNIVAC computer

TOK

COBOL the first programming language !

- The goal of COBOL was to create a **universal, business-oriented programming language** that could be easily understood by non-programmers.
- It includes features for handling data records, files, and computations commonly found in business operations.
- One of COBOL's distinctive features is its **English-like syntax**, which was intended to make the language more readable and accessible to business professionals.

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID.     HELLOWORLD.
000300
000400*
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000700 SOURCE-COMPUTER. RM-COBOL.
000800 OBJECT-COMPUTER. RM-COBOL.
000900
001000 DATA DIVISION.
001100 FILE SECTION.
001200
100000 PROCEDURE DIVISION.
100100
100200 MAIN-LOGIC SECTION.
100300 BEGIN.
100400     DISPLAY " " LINE 1 POSITION 1 ERASE EOS.
100500     DISPLAY "Hello world!" LINE 15 POSITION 10.
100600     STOP RUN.
100700 MAIN-LOGIC-EXIT.
100800     EXIT.
```

Example of COBOL code

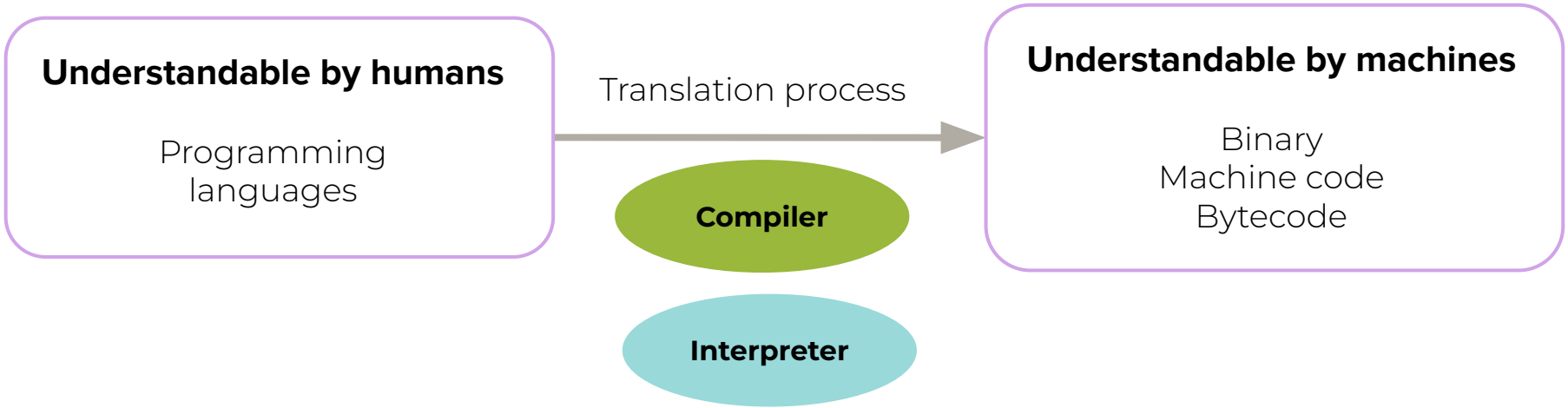
TOK

Let’s fill out the worksheet !

	Compiler	Interpreter	Bytecode interpreter	JIT compiler
Error detection				
Translation time				
Portability				
Applicability				

Characteristics of programming languages

Compiler or Interpreter: The program has to be able to be processed and translated to machine code/byte code in order to be executed



Characteristics of programming languages

Fixed vocabulary: a programming language has a predefined set of keywords, operators, and symbols with well-defined meanings and usages

```
long  return  public  class  for
```

Characteristics of programming languages

Unambiguous meaning: refers to the property of a programming language where every construct and statement has a clear and precise interpretation or meaning. There should be no ambiguity or multiple interpretations of the same code.

```java
public static void main(String[] args) {
    int result = add( a: 3,  b: 5);
    System.out.println("The result is: " + result);
}
1 usage
public static int add(int a, int b) {
    return a + b;
}
```

Characteristics of programming languages

Consistent Syntax: defines how code should be structured and written. Syntax includes rules for declaring variables, defining functions, organizing code blocks, ...

```java
public static void main(String[] args) {
    int result = add( a: 3,  b: 5);
    System.out.println("The result is: " + result);
}
1 usage
public static int add(int a, int b) {
    return a + b;
}
```

Characteristics of programming languages

Consistent grammar: refers to the adherence to a set of rules and conventions within a programming language that govern the structure, syntax, and organization of code.

```java
public static void main(String[] args) {
    int result = add( a: 3,  b: 5);
    System.out.println("The result is: " + result);
}
1 usage
public static int add(int a, int b) {
    return a + b;
}
```

Characteristics of programming languages

Variables: Programming languages allow developers to declare variables to store and manipulate data. These variables are stored in computer memory.

```java
public static void main(String[] args) {
    int result = add( a: 3,  b: 5);
    System.out.println("The result is: " + result);
}
1 usage
public static int add(int a, int b) {
    return a + b;
}
```

Characteristics of programming languages

Data Types: Languages support various data types, such as integers, floats, strings, and booleans, to represent different kinds of information and allow basic arithmetic and logical operations operations on those types

```java
public static void main(String[] args) {
    int result = add( a: 3,  b: 5);
    System.out.println("The result is: " + result);
}
1 usage
public static int add(int a, int b) {
    return a + b;
}
```

Characteristics of programming languages

Control Structures: Most programming languages include control structures like loops (for, while) and conditionals (if, else) to control the flow of the program.

```java
public static boolean isOdd(int a) {
    if (a % 2 == 1) {
        return true;
    } else {
        return false;
    }
}
```

Characteristics of programming languages

Functions/Methods: Languages typically allow you to define functions or methods to encapsulate blocks of code that can be reused.

```java
public static boolean isOdd(int a) {
    if (a % 2 == 1) {
        return true;
    } else {
        return false;
    }
}
```

How does a computer language differ from a natural language?

- Purpose
- Audience
- Syntax
- Grammar
- Precision
- Ambiguity

=> Natural language processing (NLP) is challenging due to the ambiguity inherent in human language !

TOK

Higher Level Languages vs Lower Level Languages

Usually what we refer to, when talking about programming languages

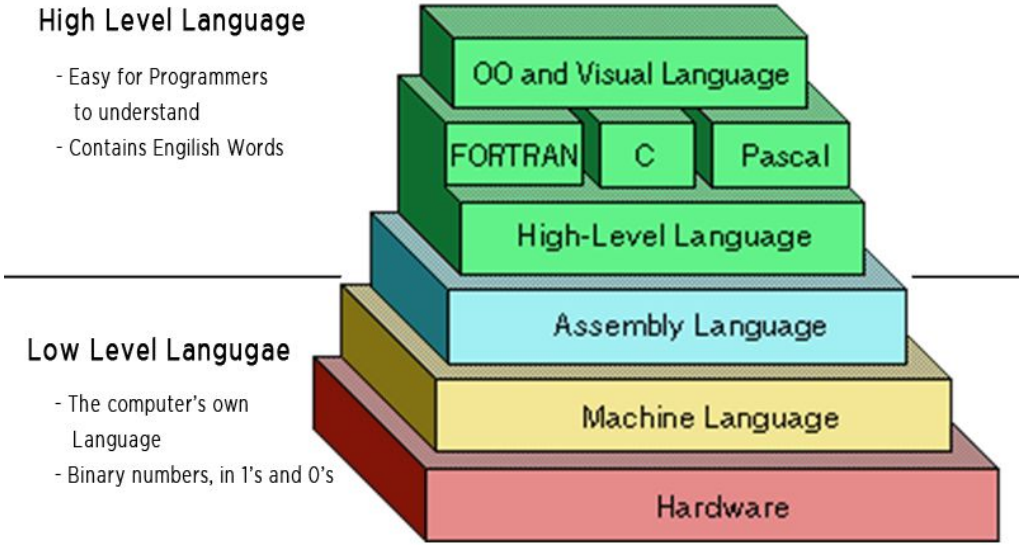
Higher Level Language

- Closer to the human language
- Hardware knowledge not required
- Easy to understand
- Translation process required
- Slower execution
- Lots of abstraction

Lower Level Language

- Closer to machine code
- Hardware knowledge required
- Hard to understand
- Translation process not required
- Faster execution
- Fewer abstraction

Higher Level Languages vs Lower Level Languages



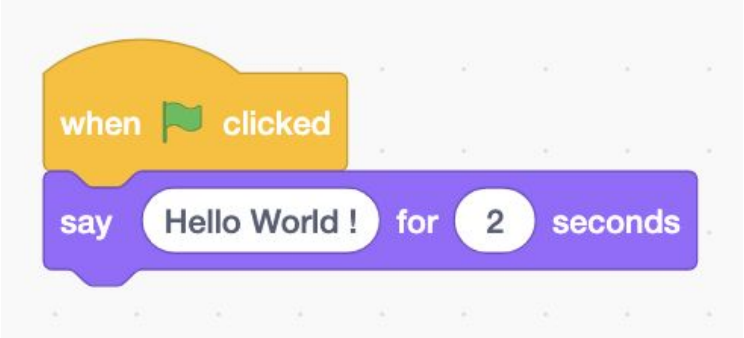
https://justcode.me/wp-content/uploads/2017/06/Low-Level-and-High-Level-Language.png

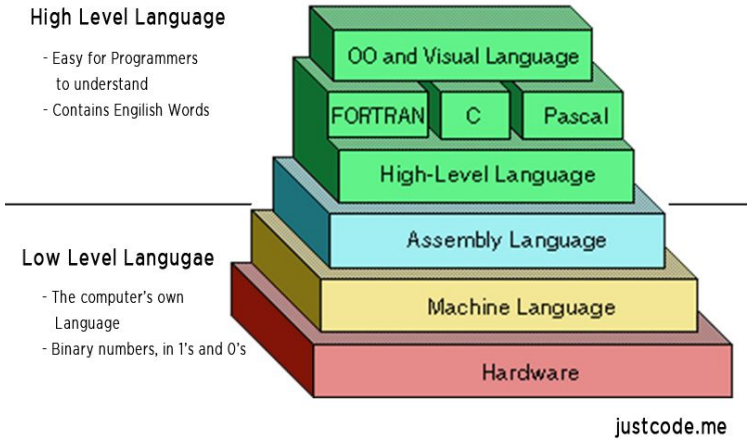
- Where would you place python ?
- Where would you place Java ?

Abstraction in programming languages

```
System.out.println("Hello World");
```

```
public void println(boolean x) {
    if (getClass() == PrintStream.class) {
        writeln(String.valueOf(x));
    } else {
        synchronized (this) {
            print(x);
            newLine();
        }
    }
}
```





Why are some programming languages more popular than others ?

https://www.youtube.com/watch?v=qQXXI5QFUfw

- Technology Advancements
- Paradigm Shifts (OOP, functional, procedural,..i)
- Communities and Ecosystem (support, updates, libraries)
- Industry Demand
- Ease of Use and Learning
- Trends and Hype

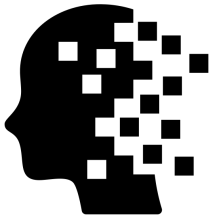
TOK

Why might the question “Is there a ‘best’ programming language ?” be irrelevant ?

- "What factors should one consider when choosing a programming language for a specific project or task?"
- "How does the choice of programming language impact the efficiency and effectiveness of software development?"
- "In what situations might one programming language be more suitable or advantageous than another?"
- "What are the key criteria for evaluating the appropriateness of a programming language for a particular programming problem?"
- "How do personal preferences and development team expertise influence the selection of a programming language in software development projects?"

TOK

Pause & Recall



Created by Victoruler
from Noun Project

Close your eyes and try to recall as many things as possible that were covered during this lesson.

Alternatively, you can keep your eyes open and write down as many things you remember on a piece of paper.

This will help strengthen your memory of key concepts 💪

Exercise 1 *(GenAI 🟠 Orange)*

Define the following terms and explain how they relate to program translation:

a) Compiler

b) Interpreter

c) Bytecode

d) Just-In-Time (JIT) Compilation

Exercise 2 *(GenAI 🟠 Orange)*

Describe one key difference between how a compiler and an interpreter handles:

a) Error detection

b) Translation time

c) Portability

d) Applicability

Exercise 3 *(GenAI 🟠 Orange)*

True or False: A compiler allows a program to run on any platform without modification.

Exercise 4 *(GenAI 🟠 Orange)*

A company is developing software that will run on many different operating systems. They want to maintain a single codebase that can be executed across all platforms.

Explain which translation method would be most suitable.

Exercise 5 *(GenAI 🟠 Orange)*

A startup is rapidly prototyping a new web application and needs to test changes frequently without long build times. They also prioritize fast debugging and quick iteration over execution speed.

Explain which translation method would be most suitable.

Exercise 6 *(GenAI 🟠 Orange)*

You're optimizing a CPU-intensive scientific computing application that must execute as quickly as possible with minimal runtime overhead.

Explain which translation method would be most suitable.

Exercise 7 *(GenAI 🟠 Orange)*

Outline the need for a translation process from high level language to machine code.

Exercise 8 *(GenAI 🟠 Orange)*

Compilers translate source code into object code. Identify two other operations performed by a compiler.

Exercise 9 - A little about Python! *(GenAI 🟠Orange/🟢Green)*

- When was python invented ?
- By whom ?
- By whom is python maintained ?
- What are the advantages of using python as a language ?
- What are the disadvantages of using python ?
- For what kind of usages is python used ?
- Can you cite a famous app written with python ?

TOK

Exercise 10 - A little about Java ! *(GenAI 🟠Orange/🟢Green)*

- When was Java invented ?
- By whom ?
- By whom is Java maintained ?
- What are the advantages of using Java as a language ?
- What are the disadvantages of using Java ?
- For what kind of usages is Java used ?
- Can you cite a famous app written with Java ?

TOK

Exercice 11 *(GenAI 🟠 Orange)*

Outline the need for higher level languages.

Exercice 12 *(GenAI 🟠 Orange)*

Identify two essential features of a computer language.

Homework

Finish all the exercises

Create flashcards for the following terms:
Interpreter
Compiler
Bytecode
Bytecode interpreter
JIT compilation
Parsing
Lexical analysis
error detection
translation time
portability
Applicability
cross-platform development
Machine code
Higher level language
Lower level language
Abstraction
Virtual machine