

Practical Programming

# Practical Programming

## Introduction

David Bouchet

david.bouchet.epita@gmail.com

<http://www.debug-pro.com/epita/prog/s3/index.html>

# Project

- Groups of students must be in the same class.
- Groups must be composed of four students.
- If the number of students in the class is not divisible by four, groups of three are allowed.
- 21-09-2018: Group submission
- 22-10-2018: First defense
- 03-12-2018: Final defense

# Programming Style

- Indent your code.
- Stay coherent.
- Stay clear.
- Identifiers should be explicit and short.
- 80 columns are enough.

# Optimization

- *“Make it right before you make it fast. Make it clear before you make it faster. Keep it right when you make it faster.”*  
P.J. Plauger – The Elements of Programming Style
- *“We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.”* Donald E. Knuth – Structured Programming with Goto Statements

# Comments

- Even good code needs comments.
- Keep comments in sync with the code.
- Good comments are never a waste of time.

# Main Types of Languages (1)

## Compiled Languages

- The source code is not executed.
- It is used to generate native machine code that will be executed by the microprocessor.
- Examples: C, C++, Go, Rust

## Interpreted Languages

- The source code is executed by an interpreter.
- No machine code is generated.
- Examples: JavaScript, PHP, Python

## Main Types of Languages (2)

**Be careful! These definitions are purely theoretical.**

In practice, some interpreted languages can be compiled and vice versa.

There are also *bytecode-compiled* languages that are compiled in an intermediate bytecode language, which is not the native machine code of the microprocessor. This intermediate language is then interpreted or just-in-time compiled (Java, C#).

# Main Types of Languages (3)

## Low-Level Languages

- Closer to hardware.
- Little abstraction from memory management.
- Less safe.
- Development process is slower.
- Execution is faster.
- Usually compiled languages.



# Main Types of Languages (3)

## High-Level Languages

- Strong abstraction from hardware.
- Strong abstraction from memory management.
- Safer.
- Development process is faster.
- Execution is slower.
- Usually interpreted languages (but not only).

# Main Types of Languages (4)

*Low-Level*

**Less safe**  
**More Control**  
**Faster**

Assembly C++

C

*Usually Compiled*

*High-Level*

**Safer**  
**Less control**  
**Slower**

Go Java

C#

Python Ruby

PHP JavaScript

*Usually Interpreted*

