# Lecture 7 Programming Languages Summary

Dr. Ir. Jacopo De Stefani - J.deStefani@tudelft.nl

## Definition

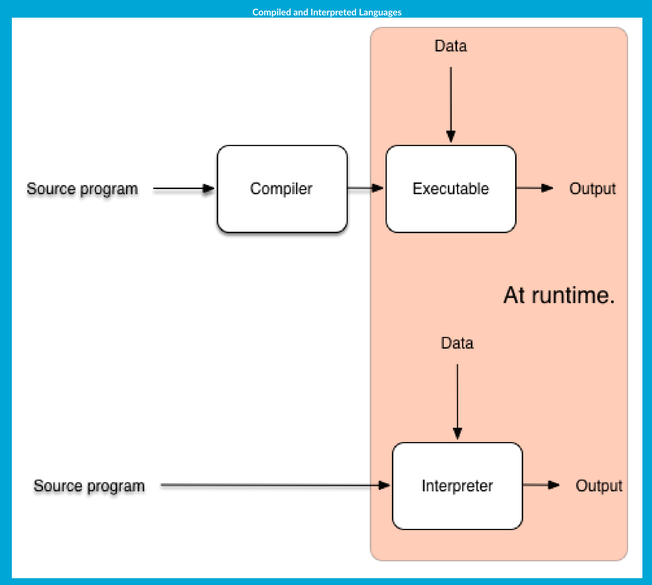
A programming language is any **set of rules** that **converts strings, or graphical program elements** in the case of visual programming languages, **to various kinds of machine code output**.

## Programming categories

- **Machine languages**, that are interpreted directly in hardware
- **Assembly languages**, that are thin wrappers over a corresponding machine language
- **High-level languages**, that are anything machine-independent
- **System languages**, that are designed for writing low-level tasks, like memory and process management
- **Scripting languages**, that are generally extremely high-level and powerful
- **Visual languages:** that are non-text based
- **Esoteric languages:** that are not really intended to be used, but are very interesting, funny, or educational in some way

TU Delft Delft University of Technology

| | C++ | Java | Javascript | Matlab/ Octave | Python | SQL | R |
|---|---|---|---|---|---|---|---|
| **Programming category** | High-level, compiled | High-level, compiled | High-level, compiled | High-level, interpreted | High-level, interpreted | High-level | High-level, interpreted |
| **Programming paradigms** | Object-oriented, Imperative, Functional | Object-oriented, Imperative, Functional | Object-oriented, Imperative, Functional | Object-oriented, Imperative, Functional | Object-oriented, Imperative, Functional | Declarative, Domain-specific Language | Object-oriented, Imperative, Functional |
| **Example of applications** | Operating systems, Simulation software, Games, Libraries, Web Browsers (Firefox) | IntelliJ (IDE), JMars, Minecraft | Paypal, React, Amazon, Google Docs | Simulink (Simulation), Numerical analysis, Signal processing, Data visualization | Dropbox, Youtube, Instagram, Spotify | Spotify, Netflix, LinkedIn | Shiny, OpenCPU, Bioconductor |
| **Domains of application** | Embedded systems, Games, Healthcare, Engineering, Research | Finance, Healthcare, Retail, Manufacturing, Education | Web Applications, IoT | Engineering, Healthcare | Engineering, Research, Machine Learning, Data analytics | Databases, Data analytics | Engineering, Research, Machine Learning, Data analytics |
| **Best practices for code quality** | • Follow the C++ coding standards<br>• Avoid using global variables<br>• Use meaningful variable and function names<br>• Use comments judiciously<br>• Avoid using raw pointers | • Proper naming conventions for variables, functions and classes<br>• Format and indent code for readability and consistency<br>• Avoid duplication in code by extracting functions<br>• Optimize code performance by avoiding nested loops and unnecessary operations | • Consistent formatting (JavaScript Standard Style)<br>• Minimizing code duplication: Using modular and reusable code<br>• Developing functions to re-use code<br>• Using comments (e.g. //) | • Use vectorization instead of loops<br>• Use of functions instead of scripts<br>• Iterate over matrices using a memory efficient indexing | • Using proper indentation (spaces vs tabs)<br>• Using naming conventions and style guide (i.e. PEP8)<br>• Employ type annotations to prevent typing issues<br>• Test code using built-in testing frameworks (e.g. PyUnit) | • Upper Case for the Clauses (SELECT, CREATE)<br>• Indention spacing and multiline commands<br>• Alias (AS)<br>• Intuitive table names<br>• Comments in code | • Consistent naming conventions for variables and functions<br>• Comments to explain the code<br>• Whitespace and indentation<br>• Avoid using global variables |




Except where otherwise noted, this work is licensed by **Jacopo De Stefani** under a Creative Commons **CC-BY-NC-SA** license: https://creativecommons.org/licenses/by-nc-sa/4.0/https://creativecommons.org/licenses/by-nc-sa/4.0/
All images are all rights reserved, solely employed for educational use, and you must request permission from the copyright owner to use this material.

## Programming paradigms

- **Imperative:** in which the programmer instructs the machine how to change its state. Examples are:
  - **Procedural** which groups instructions into procedures,
  - **Object-oriented** which groups instructions with the part of the state they operate on,
- **Declarative:** in which the programmer merely declares properties of the desired result, but not how to compute it. Examples are:
  - **Functional** in which the desired result is declared as the value of a series of function applications,
  - **Logic** in which the desired result is declared as the answer to a question about a system of facts and rules,

## Code hygiene / Code quality

Program hygiene, aka style, entails a program's readability, maintainability and logic structuring. A good quality code:

- Follows a consistent style (i.e. naming conventions, indentation).
- It is easy to understand for different programmers.
- Has been well-documented.
- It is testable.

## Sources


- TB141IC - ICT System Engineering and Rapid Prototyping material (Lecture 7)
- Students contribution (Cohort Q3 2023)