



Designing Programming Language

Introduction

MESSI NGUELE Thomas (PhD)

Yaoundé, Mars 13th 2025

Chapter Goal

- ❶ It looks at the different stages involved in the development of software.
- ❷ It shows that programming language are designed for the construction of reliable software.
- ❸ It also provides the distinction between a language and its development environment.
- ❹ It presents the low-level building blocks used in the construction of a language :
 - ❶ Character set.
 - ❷ Rules for identifiers and special symbols.
 - ❸ The handling of comments, blanks and layout.

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design
- 4 The lexical elements
- 5 Exercises

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design
- 4 The lexical elements
- 5 Exercises

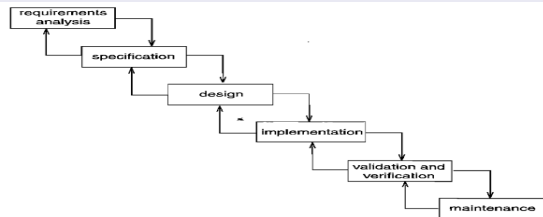
The Diversity of Languages

- ❶ There are many different programming languages designed by various research groups, international committees, and computers companies.
 - Used **also outside** the group designed them.
 - Used **only inside** the group designed them.
 - ❷ A large number of languages remain in current use and new languages continue to emerge.
 - Can appear very confusing for student who have mastered one language
 - They may ask : "**Does a lifetime of learning new language await me ?**"
 - ❸ The situation is not as bleak as it appears because :
 - Languages often have many more similarities than differences.
 - Individual languages are not usually built on separate principles ;
 - Languages differences are often due to quite minor variations in the same principle.
- Finally, a new faced language provides the same facilities disguised in a different syntax to the one of the language we already knew.

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design
- 4 The lexical elements
- 5 Exercises

Software Development Process



- Waterfall model
- Spiral model (incremental and iterative approach)
- * OO development methods are influenced by OO languages

- 1 **Requirement analysis** : understand and clarify the user requirements.
- 2 **Specification** : a document that defines the problem to be solved.
- 3 **Design** : Imagine the solution (artifacts, components, objects, architecture).
- 4 **Implementation** : Translate the solution in a **programming language**.
- 5 **Validation and verification** : Does the solution satisfy the specification?
- 6 **Maintenance** : Correction of errors and add new user requirements.

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design**
- 4 The lexical elements
- 5 Exercises

Language Design

- **Declarative programming** : a paradigm that expresses the logic of a computation without describing its control flow. (**What** is the problem to be solved ?). Logic programming, functional programming.
 - **Imperative programming** : a paradigm describing the operations in sequences of instructions executed by the computer to modify the state of the program. (**How** to solve the problem ?)
-
- ① **Expressive power** : allows the programmer to be concentrated on problem solving rather than to think about the target computer.
 - ② **Simplicity** : implies that a language allows programs to be expressed concisely in a manner that is easily written, understood and read.
 - ③ **Orthogonality** : allows any combination of the basic language constructs to build other language elements
 - ④ **Implementation** : Compilation (imperative) or interpretation (declarative).
 - ⑤ **Error detection and correction. ; Correctness and standards :**

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design
- 4 The lexical elements**
- 5 Exercises

The lexical elements

- 1 Character set.
- 2 Identifiers and reserved words.
- 3 Comments.
- 4 Spaces and line termination.

Agenda

- 1 The Diversity of Languages
- 2 The Software Development Process
- 3 Language Design
- 4 The lexical elements
- 5 Exercises**

Exercises

- 1 - An orthogonal language is constructed by combining a small set of basic language constructs. It should therefore be simple to learn and use. Discuss whether that is in fact the case.
 - It is the case because the only thing to know is this set of basic language constructs and the rules of combining these basic language constructs in order to get another one.
- 2 - Describe the importance of readability in the creation of reliable software. What are the features of a language that enhance readability and what makes it more difficult ?
 - Readability allows program to be read and understood easily. This kind of program can be developed by many persons. In that way readability contributes to the creation of reliable software.
 - The features that enhance readability are simplicity, orthogonability and expressive power.

Exercises

- 3 - How important is it that there is a single standard definition of a language?
 - It increases the portability (in the case implementers do not omit any feature).
- 4 - What are the advantages and disadvantages of considering a library to be part of a language definition as opposed to it being considered part of a language implementation?
 - Advantage : It improves portability. In fact, every implementation of the language will have this library.
 - Disadvantage : It makes language definition be heavy or complex.
- 5 - According to language design, how can you compare the following programming languages? C, C++, Java, Python, Pascal, Prolog, Mercury, GO.
- 6 - Make history of previous languages.

Bibliography

- WILSON, Leslie B. et CLARK, Robert George. Comparative programming languages. Pearson Education, 2001.
- MACLENNAN, Bruce J. Principles of programming languages : design, evaluation, and implementation. Holt, Rinehart & Winston, 1986.

Thank you for your attention!

