

## UNIT 1. Introduction to Language Processors

In this course we are interested in introducing the main design practices and principles for the development of programming languages. The course covers the analysis and synthesis phases of a language processor: lexical analysis, syntax analysis (top-down and bottom-up techniques), semantic analysis, runtime environments, error handling, intermediate code, code optimization, and final code generation.

To start the course, Unit 1 introduces the main aims of the course and describes important concepts related to compilers, translators, interpreters, and related programs. It also presents the main components in the architecture of a compiler, their evolution, programming paradigms and language design.

The main objective of the course is to understand compilers and programming language by means of:

- Understanding the code structure.
- Understanding the language semantics.
- Understanding the relation between source code and generated machine code.
- Becoming a better programmer.

At the end of the course you should know how to:

- Develop a scanner for a given language.
- Write a grammar for a given language.
- Verify and modify a grammar fulfills some properties.
- Write a parser for a given language using diverse kinds of techniques.
- Verify the semantics of a given language.
- Implement different techniques for error detection and management.
- Generate intermediate/final code.