



Computer Programming and its basic concepts Trainer Materials

Subchapter 1.5. – Programs and Program Languages

WP3: Code4SP Training Materials

Prepared by:



CITIZENS
IN POWER



Center for Social
Innovation



ZAUG
gGmbH





Subchapter 1.5.: Programs and Program Languages





Programs and Program Languages

Programming languages are used to create programs, which are used to control the behavior of a machine, typically a computer. A programming language provides a structure for the programmer to give instructions to the machine, and a way to communicate those instructions to other programmers. There are many programming languages in use today. The most popular ones are C, Java, Python, and JavaScript.





Programs and Program Languages

Types of languages

There are dozens of programming languages in use today, but they can be broadly classified into five categories:

- **Low-level programming languages:** These programming languages are very close to the hardware and are used to program microprocessors and other low-level devices. They are not easy to learn and are not popular for general-purpose programming. Examples: Assembly language, C programming language, and Low-level assembly language.
- **High-level programming languages:** These programming languages are designed to be easy to learn and use. They are popular for general-purpose programming. Examples: Java, C++, and Python.
- **Scripting languages:** Scripting languages are designed to be easy to use and are popular for scripting purposes. Examples: Python, Ruby, and JavaScript.
- **Domain-specific languages:** Domain-specific languages are designed for a specific task or industry. They are not easy to learn and are not popular for general-purpose programming. Examples: MATLAB, SQL, and FORTRAN.
- **Object-oriented programming languages:** These programming languages are based on the object-oriented programming paradigm. Examples: Java, C++, and Python.





Programs and Program Languages

From a high-level program to an executable file

When a computer program is written in a high-level language, it is first translated into a lower-level language, which is more easily understood by machines. The lower-level language is then compiled into an executable file, which can be run on a computer.





Programs and Program Languages

IDEs (Integrated Development Environments)

An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE typically consists of a source code editor, build automation tools, and a debugger. The source code editor allows the programmer to write code, while the build automation tools automate the process of compiling that code into a form the computer can run. The debugger allows the programmer to step through the code, examining the state of the program at each point in its execution. IDEs are often used in conjunction with a version control system, which allows different programmers working on the same project to share and merge their changes seamlessly.





Programs and Program Languages

The common elements in programming languages

Computer programming languages share a number of common elements, despite their differences. All programming languages have a way of representing instructions to the computer in a form that the computer can understand. This is usually called code, or source code. Programmers use code to create software programs and applications. All programming languages also have a way of organizing instructions so that they can be reused, modified, or shared with other programmers. This is usually called a library or module. Libraries and modules allow programmers to create complex programs by building on the work of other programmers. Finally, all programming languages have a way of conveying information to the user about what the program is doing and how it is performing. This is usually called output or debug information. Output and debug information helps programmers understand and fix problems with their programs.





Programs and Program Languages

Procedural and object-oriented programming

There are two main types of programming: procedural and object-oriented. Procedural programming involves a step-by-step process, while object-oriented programming involves creating objects that interact with one another. Procedural programming is often seen as simpler than object-oriented programming. It is easy to learn the steps required to complete a task, and it is easy to change the order of those steps without affecting the outcome. However, procedural programming can be less efficient because it can be difficult to reuse code that has been written for a specific task. Object-oriented programming is more complex than procedural programming, but it allows for more flexibility and reuse of code. Objects can be created for specific tasks and then reused as needed. In addition, object-oriented code is often easier to read and understand than procedural code. However, object-oriented programming can be more difficult to learn and may be less efficient than procedural programming.





THANK YOU!

Co-funded by the
Erasmus+ Programme
of the European Union

