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

code4sp

coding for social promotion

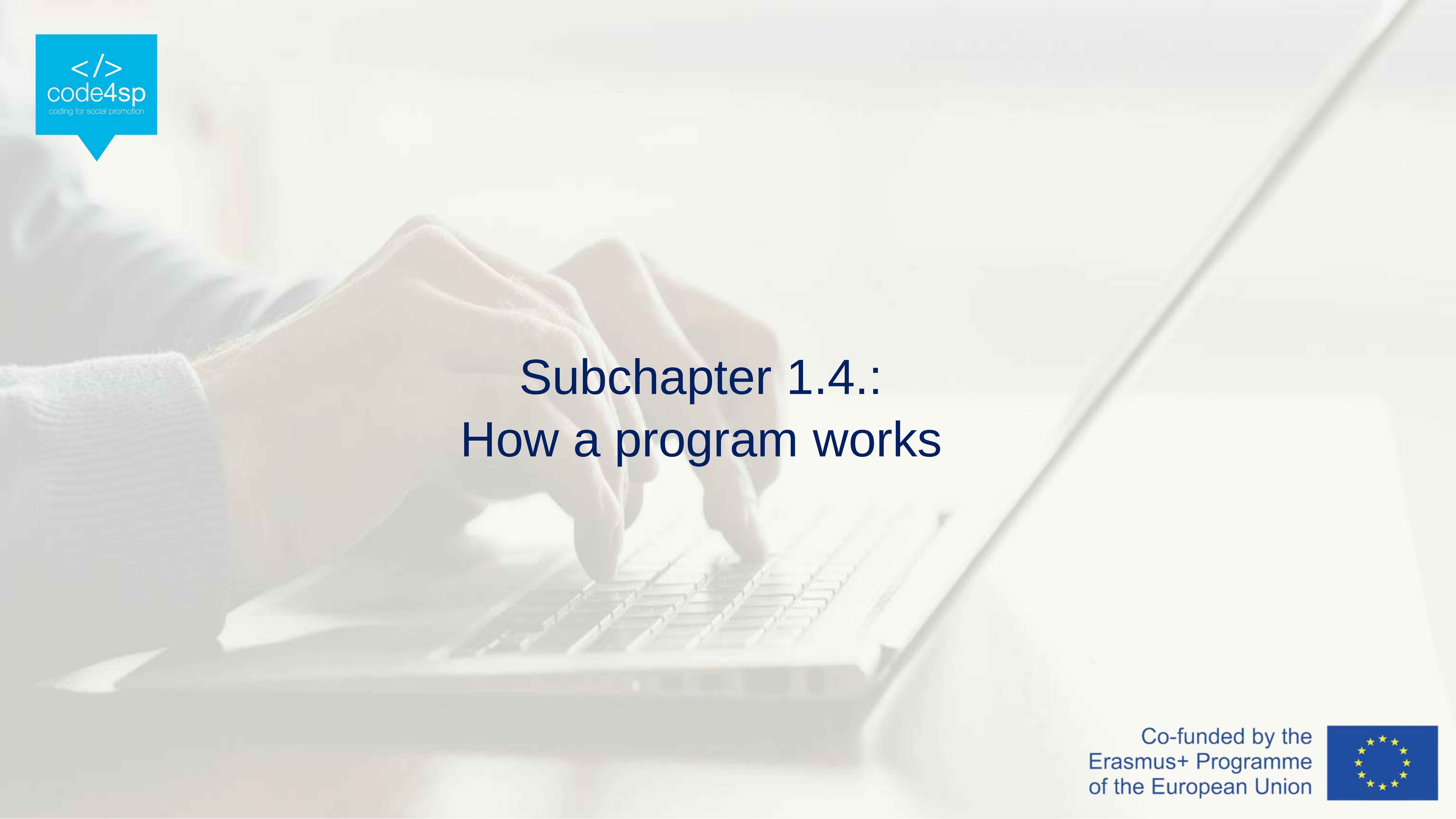
Computer Programming and its basic concepts Trainer Materials
Subchapter 1.4. – How a program works

WP3: Code4SP Training Materials

Prepared by: CD DR

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein. (Project Number: 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN)

</>
code4sp
coding for social promotion



Subchapter 1.4.:
How a program works

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

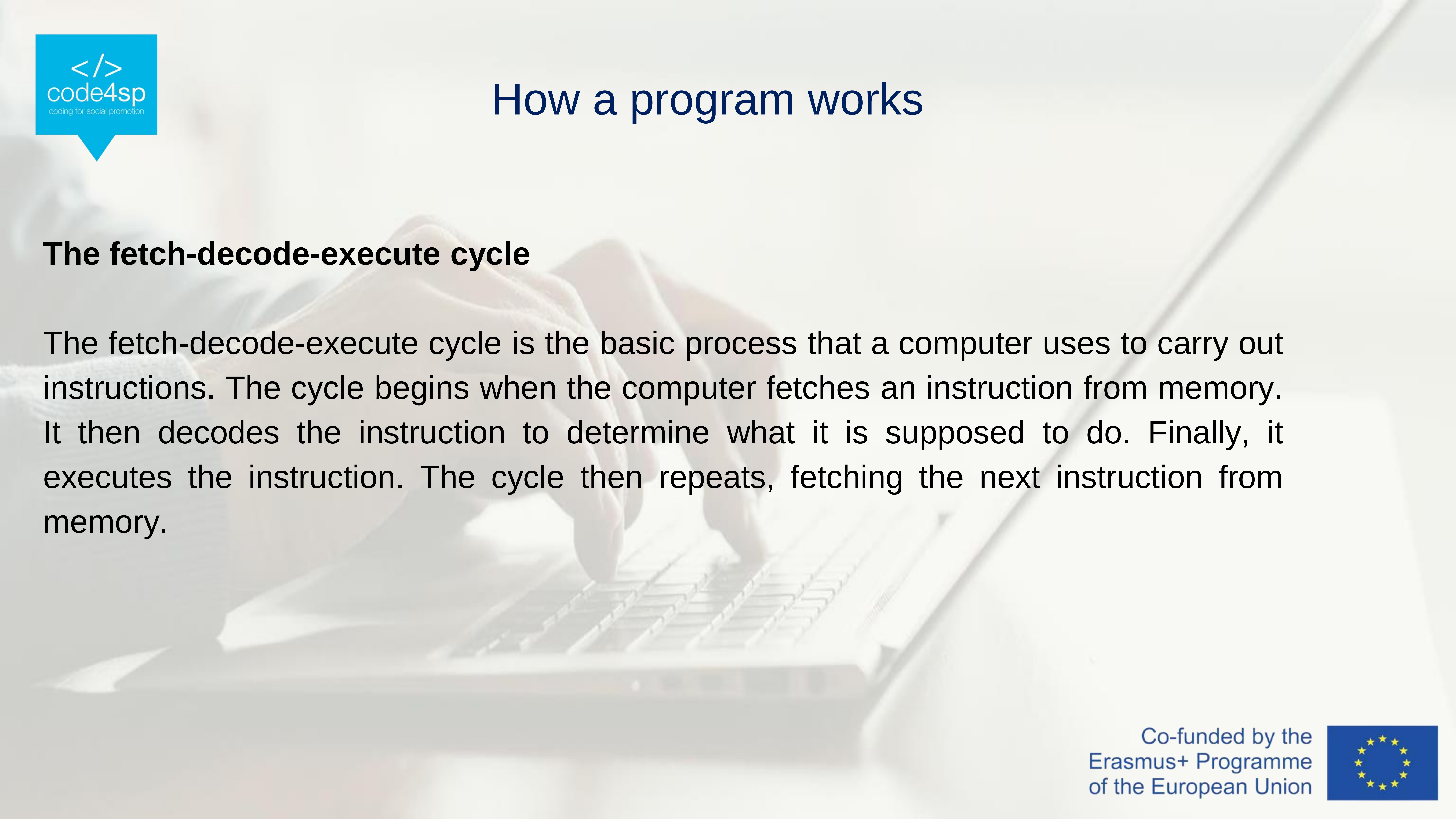
How a program works

How a program works

There are many different types of computer programs, but they all have the same basic components: a user interface, a processor, and memory. The user interface allows the user to input information and instructions into the program, the processor carries out the instructions, and the memory stores the program and the data it processes. Most computer programs are written in a high-level programming language, which is a language that is designed to be easy for humans to read and write. However, the processor can only understand machine code, which is a series of ones and zeroes. So, before a program can be run, it must be converted into machine code. This is done by a program called a compiler. The compiler reads the program and converts it into machine code. It then stores the machine code in a file called an executable. When the user runs the program, the executable is loaded into memory and the processor carries out the instructions.

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

How a program works



The fetch-decode-execute cycle

The fetch-decode-execute cycle is the basic process that a computer uses to carry out instructions. The cycle begins when the computer fetches an instruction from memory. It then decodes the instruction to determine what it is supposed to do. Finally, it executes the instruction. The cycle then repeats, fetching the next instruction from memory.

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

How a program works

From machine language to assembly language

As programming in machine language, which consist only of binary code is too complicated for a human being, assembly language was created. Assembly language is a low-level programming language for a computer, microprocessor, or other programmable device, in which the programmer uses assembly language instructions to control the operation of the device. Assembly language is specific to a certain microprocessor or family of microprocessors. It consists of a series of mnemonic codes, symbolic names for the operations that the microprocessor can perform, and the operands (data) upon which these operations are to be performed. Assembly language is converted into machine code, a form of binary code that is specific to a particular type of computer and can be understood by the computer's processor.

Even assembly language programming is easier than machine language programming it was not handy enough to produce fast and easy to read source code. Therefore high-level programming languages (such as C# or python) where created.

High-level programming languages are easier to use than low-level programming languages. They allow you to focus on the task at hand, rather than on the details of the computer. This makes them ideal for creating applications and programs. High-level programming languages also tend to be more forgiving than low-level programming languages. If you make a mistake when writing code in a high-level language, the compiler will usually be able to correct it for you. This can save you a lot of time and frustration when coding.

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

How a program works

Key Words, Operators, and Syntax: an overview

There are many high-level programming languages available today. Each has its own unique set of keywords, operators, and syntax. In order to be effective with a high-level programming language, it is important to be familiar with the specific keywords, operators, and syntax used by that language. Some of the most common keywords used in high-level programming languages include: if, then, else, while, for, do, break, continue. These keywords are used to control the flow of program execution. Operators are symbols that represent operations that can be performed on values. The most common operators include: + (addition), - (subtraction), * (multiplication), / (division), and % (modulus). These operators can be used to calculate the results of expressions. The syntax of a programming language is the set of rules that govern how code must be written in order to be interpreted by the compiler or interpreter. The syntax of a high-level programming language is typically more forgiving than the syntax of a lower-level language. This can make it easier for beginners to learn how to program.

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

How a program works

Compilers and Interpreters

Computer compilers and interpreters are important tools for software developers. A compiler takes code written in one language and converts it into code that can be run on a different machine. An interpreter takes code written in one language and runs it as it is, without compiling it first. Compilers are typically used for languages that have a lot of structure, like C or Java. Interpreters are typically used for languages that are more flexible, like Python or Ruby. Compilers usually produce faster code than interpreters. However, interpreters are typically more portable, meaning they can run on more types of machines. Which tool to use depends on the situation. If speed is important, a compiler is a better choice. If portability is important, an interpreter is a better choice.

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

</> code4sp
coding for social promotion

THANK YOU!

NEXT CHAPTER: Programs and Program Languages

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