



C PROGRAMMING





LEARNING OBJECTIVES

AT THE END OF THIS LESSON,
YOU WILL BE ABLE TO:

- EXPLAIN THE DIFFERENT PROGRAMMING TERMINOLOGIES;
- EXPLAIN THE ADVANTAGES OF USING C LANGUAGE;
- EXPLAIN PROGRAM AND THE SIMPLE PROGRAM PROCESS;





“Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn.”

STEPHEN HAWKING





CREATOR OF C LANGUAGE

DENNIS RITCHIE

created the C Programming
Language at Bell Labs from
1969 to 1973.





Search ...



WHAT IS C PROGRAMMING LANGUAGE

- It is defined as a general programming language which is one of the most used languages of all time.
- It had been used initially to design the Unix operating system that had numerous requirements, especially high performance.
- It has influenced a lot of very well-known and used language such as C++, Objective-C, Java, JavaScript, Perl, PHP, and many others.
- C is both imperative and structured.

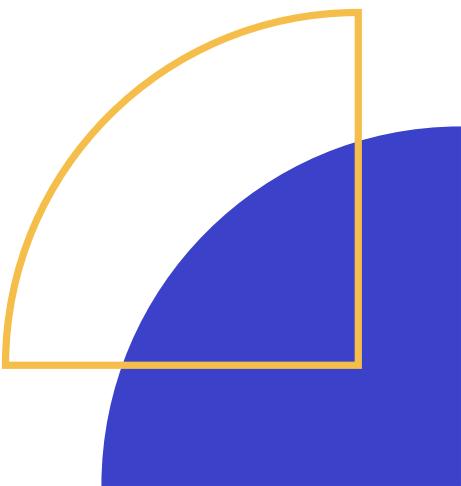




IMPERATIVE



Is a programming paradigm that uses statements that change a program's state.

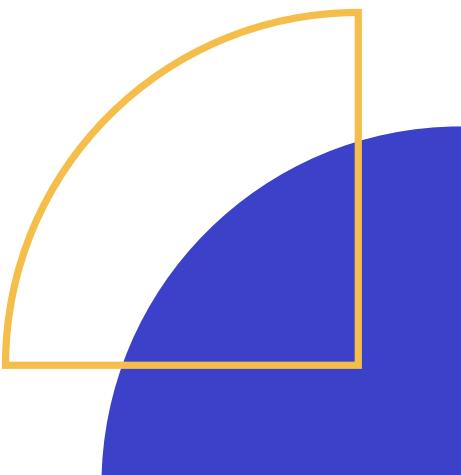




STRUCTURED



Is a programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of the structured control flow constructs of selection and repetition, block structures, and subroutines.

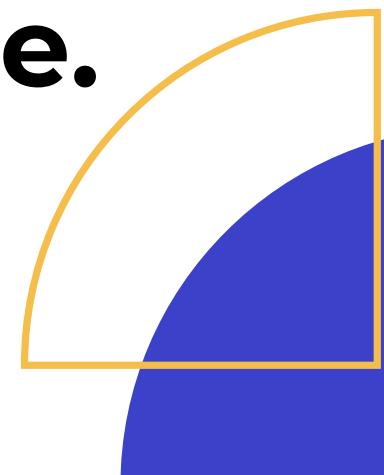


ADVANTAGES OF USING C LANGUAGE

Search ...



- It is small and easy to learn.
- It is processor-independent because compilers exist for almost all processors in the world. This independence provides something very useful to programmers: they can focus on algorithms and the application levels of their job instead of thinking about the hardware level at each row of code.
- It is a very “low-level” high-level language.

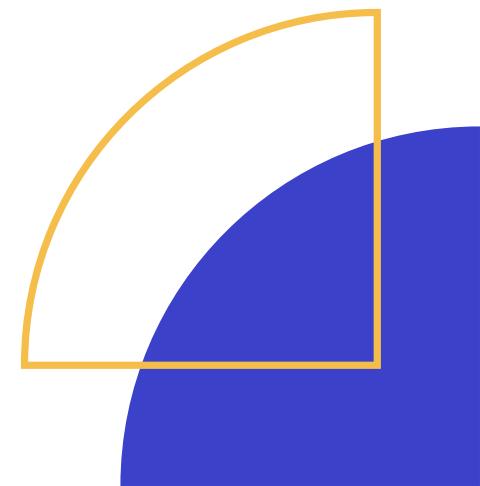


WHAT IS A PROGRAM?

Search ...



A program is text that you write using a programming language that contains behaviors that you need a processor to acquire. It basically creates a way of handling inputs, processing the inputs, and finally producing outputs according to these behaviors





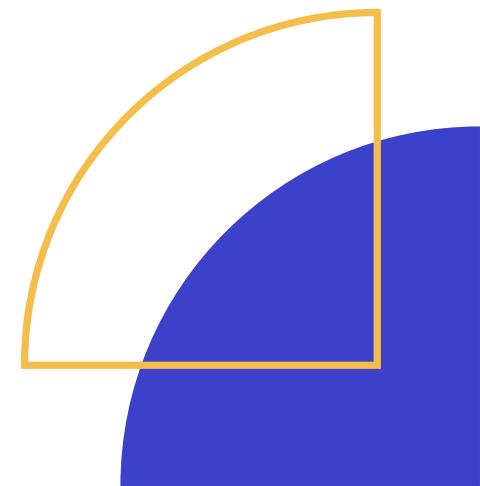
WHAT IS A PROGRAM?





MACHINE LANGUAGE

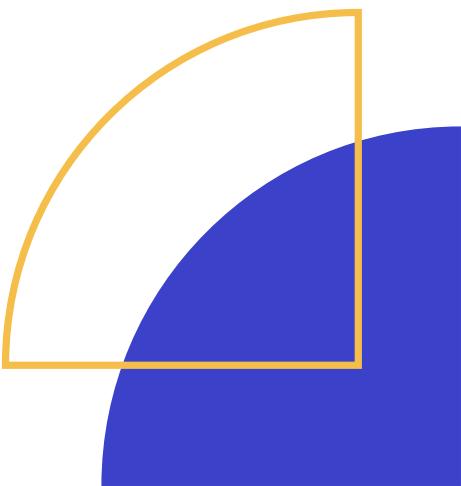
- Sometimes referred to as machine code or object code.
 - It is a low-level language.
 - It is a collection of binary digits or bits that the computer reads and interprets.
 - It is the only language a computer is capable of understanding.
- Each computer has its own machine language and the computer can execute instructions written in that language only.





ASSEMBLY LANGUAGE

- An assembly language is a type of low-level programming language that is intended to communicate directly with a computer's hardware.
- Unlike machine language, which consists of binary and hexadecimal characters, assembly languages are designed to be readable by humans.
- Uses Mnemonic instruction codes instead of pure binary digits.





Search ...



ASSEMBLER

A PROGRAM THAT IS USED TO
CONVERT AN ASSEMBLY
LANGUAGE PROGRAM INTO
MACHINE LANGUAGE.





LOW-LEVEL LANGUAGE

- The machine-level language comes at the lowest level in the hierarchy, so it has zero abstraction level from the hardware.
- It cannot be easily understood by humans.
- The machine-level language is written in binary digits, i.e., 0 and 1.
- It does not require any translator as the machine code is directly executed by the computer.
- It is a first-generation programming language.

ASSEMBLY LANGUAGE

- The assembly language comes above the machine language means that it has less abstraction level from the hardware.
- It is easy to read, write, and maintain.
- The assembly language is written in simple English language, so it is easily understandable by the users.
- In assembly language, the assembler is used to convert the assembly code into machine code.
- It is a second-generation programming language.





Search ...



DRAWBACKS OF ASSEMBLY LANGUAGE

- IT WAS VERY TEDIOUS AND ERROR PRONE.
- IT FORCED THE PROGRAMMER TO THINK IN TERMS OF THE MACHINE RATHER THAN IN TERMS OF HIS PROBLEM
- +A PROGRAM WRITTEN IN THE MACHINE LANGUAGE OF ONE COMPUTER COULD NOT BE RUN ON A COMPUTER WITH A DIFFERENT MACHINE LANGUAGE. CHANGING YOUR COMPUTER COULD MEAN HAVING TO REWRITE ALL YOU PROGRAMS.





HIGH-LEVEL OR PROBLEM-ORIENTED LANGUAGES

- These languages were developed in the late 1950s to 1960.
 - The high-level language is a programming language that allows a programmer to write the programs which are independent of a particular type of computer.
- 

HIGH-LEVEL OR PROBLEM-ORIENTED LANGUAGES

- The high-level languages are considered as high-level because they are closer to human languages than machine-level languages.
- When writing a program in a high-level language, then the whole attention needs to be paid to the logic of the problem.

COMPILER

A PROGRAM THAT IS USED
TO TRANSLATE A PROGRAM
WRITTEN IN A HIGH-LEVEL
LANGUAGE TO MACHINE
LANGUAGE.



ADVANTAGES OF HIGH-LEVEL LANGUAGE

- The high-level language is easy to read, write, and maintain as it is written in English like words.
- The high-level languages are designed to overcome the limitation of low-level language, i.e., portability. The high-level language is portable; i.e.. these languages are machine independent.



LOW-LEVEL LANGUAGE

- It is a machine-friendly language, i.e., the computer understands the machine language, which is represented in 0 or 1.
- The low-level language takes more time to execute.
- It requires the assembler to convert the assembly code into machine code.
- The machine code cannot run on all machines, so it is not a portable language.
- It is memory efficient.
- Debugging and maintenance are not easier in a low-level language.

HIGH-LEVEL LANGUAGE

- It is a user-friendly language as this language is written in simple English words, which can be easily understood by humans.
- It executes at a faster pace.
- It requires the compiler to convert the high-level language instructions into machine code.
- The high-level code can run all the platforms, so it is a portable language.
- It is less memory efficient.
- Debugging and maintenance are easier in a high-level language.



WRAP-UP

DENNIS RITCHIE

THE CREATOR OF THE C PROGRAMMING
LANGUAGE AT BELL LABS FROM 1969 TO
1973.

C PROGRAMMING LANGUAGE

IT IS DEFINED AS A GENERAL PROGRAMMING LANGUAGE WHICH IS ONE OF THE MOST USED LANGUAGE OF ALL TIME. IT IS BOTH IMPERATIVE AND STRUCTURED PROGRAMMING LANGUAGE.

ADVANTAGES OF USING C

- IT IS EASY TO LEARN.
- IT IS PROCESSOR-INDEPENDENT.
- IT IS A “LOW-LEVEL” HIGH LEVEL LANGUAGE

PROGRAM

A PROGRAM IS A TEXT THAT YOU WRITE USING A PROGRAMMING LANGUAGE THAT CONTAINS BEHAVIORS THAT YOU NEED A PROCESSOR TO ACQUIRE. BASICALLY CREATES A WAY OF HANDLING INPUTS, PROCESSING THE INPUTS, AND FINALLY PRODUCING OUTPUTS ACCORDING TO THESE BEHAVIORS

MACHINE LANGUAGE

IT IS A COLLECTION OF BINARY DIGITS OR BITS THAT THE COMPUTER READS AND INTERPRETS, IT IS THE ONLY LANGUAGE THE COMPUTER UNDERSTANDS.

ASSEMBLY LANGUAGE

AN ASSEMBLY LANGUAGE IS A TYPE OF LOW-LEVEL PROGRAMMING LANGUAGE THAT IS INTENDED TO COMMUNICATE DIRECTLY WITH A COMPUTER'S HARDWARE. USES MNEMONIC INSTRUCTION CODES.

HIGH-LEVEL LANGUAGE

THE HIGH-LEVEL LANGUAGE IS A PROGRAMMING LANGUAGE THAT ALLOWS A PROGRAMMER TO WRITE THE PROGRAMS WHICH ARE INDEPENDENT OF A PARTICULAR TYPE OF COMPUTER.

COMPILER

**A PROGRAM THAT IS USED TO TRANSLATE A PROGRAM
WRITTEN IN A HIGH-LEVEL LANGUAGE TO MACHINE
LANGUAGE.**

THANK YOU

