

GRACE HOPPER



Grace Hopper (1906 – 1992) was a pioneering computer scientist who helped develop the first compiler, a tool that translates programming languages into machine code. She also played a key role in the development of the COBOL programming language. Hopper was a strong advocate for greater inclusion of women in computer science and received numerous awards for her contributions to the field. She was known for her intelligence, innovation, and determination, and her legacy continues to inspire future generations of computer scientists. Hopper passed away in 1992, but her impact on the field of computer science is still felt today.

“A ship in port is safe, but that is not what ships are built for.”

The Queen of Code

Grace Hopper's work on the first compiler and the COBOL programming language revolutionized computer science. She created the first compiler, which translated programming languages into machine code, making programming more accessible and efficient. Hopper's work on the first compiler paved the way for the development of new programming languages, including COBOL, which she helped to create. COBOL became widely used in business and government, making it possible to perform complex operations and data processing tasks with greater ease and speed. Hopper's contributions to the development of the compiler and COBOL laid the foundation for modern programming and had a significant impact on the growth of the computer industry.

More information

[Documentary: "She Taught Computers to Talk"](#)

[Lecture by Hopper](#)

Experiment: Explore Code Compilers

Did you know that computers can only execute low-level programming languages such as machine code or assembly code? Machine code is just a stream of binary data (a sequence of 0's and 1's) and assembly language also has little semantics or formal specification, being only a mapping of human-readable symbols, including symbolic addresses, to opcodes, addresses, numeric constants, strings and so on. Before the compiler and high-level programming languages were invented, programmers had to write codes in machine code or assembly code. These low-level programming languages are quite different from human languages, making coding difficult.

Every programming language you write in needs to be translated so that a computer can understand it. This is done using a special program called a compiler. The process of compiling involves parsing the source code, analyzing it for errors, optimizing it for efficiency, and generating object code or executable files. The resulting machine code can be executed directly by the computer's CPU. The use of a compiler allows programmers to write code in a higher-level language, making programming more accessible and efficient. In this experiment, you will explore code compilers and how they work.

What you need

A computer, an internet connection, and access to an online code editor and compiler.

Procedure

1. Choose a programming language that you are interested in learning more about, such as Python or Java.
2. Open an online code editor and compiler. A good one would be Compiler Explorer (<https://godbolt.org/>).
3. Write a simple program in your chosen programming language. For example, you could write a program that prints "Hello, world!" to the screen.
4. Look at the output of the compiler on the right side of your screen. This will show you the assembly code that was generated from your program.
5. Experiment with making changes to your program, such as adding or removing lines of code, and observe how the compiler generates different assembly code.
6. Research the history of compilers and how they have evolved over time.
7. Optional: If you can code in different languages, try to write the same program in another language and see the compiling results. Are they similar?

Think about the following questions:

What is the role of a compiler in the programming process? How does a compiler translate high-level code into machine code? What are the benefits and drawbacks of using a compiler versus an interpreter? How have compilers evolved over time, and what impact have they had on the development of programming languages?