# **COMPUTER PROGRAMMING**

**Computer programming** is the process of designing and building an executable program to accomplish a specific computing result or to perform a specific task.

## **Types of programming languages**

1. Procedural Programming Language.
2. Functional Programming Language.
3. Scripting Programming Language.
4. Logic Programming Language.
5. Object-Oriented Programming Language.

### **Procedural programming language**

### A **procedural language** is a computer [programming language](https://www.computerhope.com/jargon/p/programming-language.htm) that follows, in order, a set of commands. Examples of computer procedural languages are [BASIC](https://www.computerhope.com/jargon/b/basic.htm), [C](https://www.computerhope.com/jargon/c/c.htm), [FORTRAN](https://www.computerhope.com/jargon/f/fortran.htm), [Java](https://www.computerhope.com/jargon/j/java.htm), and [Pascal](https://www.computerhope.com/jargon/p/pascal.htm).

### Procedural languages are some of the common types of programming languages used by script and software programmers. They make use of [functions](https://www.computerhope.com/jargon/f/function.htm), [conditional statements](https://www.computerhope.com/jargon/c/contstat.htm), and [variables](https://www.computerhope.com/jargon/v/variable.htm) to create programs that allow a computer to calculate and display a desired output.

|  |  |  |
| --- | --- | --- |
| **Procedural language** | **Year it was founded** | **Founder** |
| Basic | 1964 | John George Kemeny and Thomas E. Kurtz |
| C | 1972 | Dennis Ritchie and Brian Kernighan |
| Fortran | 1959 | John Backus |
| Java | 1991 | James gosling |
| Pascal | 1970 | Niklaus Wirth |

### **Functional programming language**

Functional programming languages are specially designed to handle symbolic computation and list processing applications. Functional programming is based on mathematical functions. Some of the popular functional programming languages include**:** LISP, Python, Erlang, Haskell, Clojure, etc.

* **Pure Functional Languages** − These types of functional languages support only the functional paradigms. For example − Haskell.
* **Impure Functional Languages** − These types of functional languages support the functional paradigms and imperative style programming. For example – LISP

.

|  |  |  |
| --- | --- | --- |
| **Functional language** | **Year it was founded** | **Founder** |
| Lisp | 1960 | John McCthy |
| Python | 1991 | Guido van Rossum |
| Erlang | 1986 | Agner Krarup Erlang |
| Haskell | 1990 |  |
|  |  |  |
|  |  |  |