Procedural programming

When you are brand new to programming, programming paradigms are not of much importance. But as you go up the stairs and start creating complex programs and software, it is vital to understand which programming paradigm is best suited for your project. Before we begin, it is important to know what exactly a paradigm is. According to many cited definitions, a paradigm is ‘a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality for the community that shares them, especially in an intellectual discipline’.

This definition is spot on, as what differentiates the paradigm is the different way of viewing reality for the community. Paradigms matter as they often travel along with a specific culture of writing programs and thinking about them. In this article, we will discuss the major programming paradigms, with a special focus on the Procedural Programming paradigm.

Key Features of Procedural Programming

The key features of procedural programming are given below:

Predefined functions: A predefined function is typically an instruction identified by a name. Usually, the predefined functions are built into higher-level programming languages, but they are derived from the library or the registry, rather than the program. One example of a pre-defined function is ‘charAt()’, which searches for a character position in a string.
Local Variable: A local variable is a variable that is declared in the main structure of a method and is limited to the local scope it is given. The local variable can only be used in the method it is defined in, and if it were to be used outside the defined method, the code will cease to work.
Global Variable: A global variable is a variable which is declared outside every other function defined in the code. Due to this, global variables can be used in all functions, unlike a local variable.
Modularity: Modularity is when two dissimilar systems have two different tasks at hand but are grouped together to conclude a larger task first. Every group of systems then would have its own tasks finished one after the other until all tasks are complete.
Parameter Passing: Parameter Passing is a mechanism used to pass parameters to functions, subroutines or procedures. Parameter Passing can be done through ‘pass by value’, ‘pass by reference’, ‘pass by result’, ‘pass by value-result’ and ‘pass by the name’.

Advantages and Disadvantages of Procedural Programming

Procedural Programming comes with its own set of pros and cons, some of which are mentioned below.

Advantages

Procedural Programming is excellent for general-purpose programming
The coded simplicity along with ease of implementation of compilers and

interpreters
A large variety of books and online course material available on tested algorithms, making it easier to learn along the way
The source code is portable, therefore, it can be used to target a different CPU as well
The code can be reused in different parts of the program, without the need to copy it
Through Procedural Programming technique, the memory requirement also slashes
The program flow can be tracked easily

Disadvantages

The program code is harder to write when Procedural Programming is employed
The Procedural code is often not reusable, which may pose the need to recreate the code if is needed to use in another application
Difficult to relate with real-world objects
The importance is given to the operation rather than the data, which might pose issues in some data-sensitive cases
The data is exposed to the whole program, making it not so much security friendly
There are different types of programming paradigm as we mentioned before, which are nothing but a style of programming.
It is important to understand that the paradigm does not cater to a specific language but to the way the program is written.
Below is a comparison between Procedural Programming and Object-Oriented Programming.