**Functional Programming**

Functional programming is a declarative programming paradigm that writes in **pure functions**, meaning that these functions do not modify variables but instead generate new ones as an output. In other words, the output of a pure function only depends on the input parameters; thus, there is no external impact, which avoids side effects. Moreover, writing pure functions also helps developers avert mutable data and shared state.

Therefore, functional programming has many benefits and is used in a lot of programming languages and frameworks. It is a popular programming paradigm due to its ability to **create maintainable and clean software** by using functions, which are vital to code organization.

## Object-Oriented Programming

Object-oriented programming is a programming paradigm that organizes data and the software structure based on the concept of **classes and objects**.

**Classes** are a set of instructions (or blueprints) that establish a data structure for a specific object, determining what the object will contain (the types of variables that can exist in an object) and how it will behave (the methods or member functions that define how to operate on the variables). Thus, **objects** are instances of classes since classes work as "templates" to create objects. Plus, objects can contain data in the form of fields (also known as attributes) and code in the form of procedures (also named methods).

|  |  |
| --- | --- |
| **Functional Programming** | **OOP** |
|  |  |
| A function is the primary unit. | Objects are the main unit. |
| Pure functions do not have side-effects. | The methods may have side effects. |
| Follows a more declarative programming model. | Mainly follows an imperative programming approach. |
| In pure functional programming languages it is impossible to create mutable objects. Thus, objects are typically immutable. | In OOP languages, the answer is not that straightforward since it depends more on the specifications of each OOP language. Hence, OOP can support both mutable and immutable objects. |
| Functional programming writes pure functions. Pure functions only produce outputs with identical inputs. Consequently, functional programming is extremely operational, practical, and, as the name indicates, functional. | OOP is not as operational as Functional programming. In fact, OOP stores data in objects, and the data is prioritized over the operations. |