Programming Paradigm

Its a style of programming, a way of thinking about software construction.

A programming paradigm does not refer to a specific language but rather to a way to program, a methodology. Some languages make it easy to write in some paradigms but not others.

**Functional programming (FP)** is a programming paradigm — a style of building the structure and elements of computer programs — that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

Functional programming (FP) is about passing data from function to function to function to get a result.

In FP, functions are treated as data, meaning you can use them as parameters, return them,

build functions from other functions, and build custom functions. Functions in FP have to be pure functions, they should avoid shared state, and side effects and data should be immutable.

A pure function is a function that given the same type of input will always return the same output, it is not dependent on a local or global state. A shared state is a state that is shared between more than one function or more than one data-structure. So with shared state, in order to understand the effects of a function,you need to know all the details of every shared variable. It adds a lot of complexity and permits less modularity

**Object-oriented programming (OOP)** is a programming paradigm based on the concept of “objects”, which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.Object-oriented programming (OOP) is about encapsulating data and behavior into objects. An OOP application will use a collection of objects which knows how to perform certain actions and how to interact with other elements of the application. For example an object could be a person. That person would have a name (that would be a property of the object), and would know how to walk (that would be a method).

A method in OOP can be considered as a procedure in PP, but here it belongs to a specific object. Another important aspect of OOP are classes. A class can be considered as a blueprint for an object.

**Procedural programming** is a programming paradigm, derived from structured programming, based upon the concept of the procedure call. Procedures, also known as routines, subroutines, or functions, simply contain a series of computational steps to be carried out.

Procedural programming (PP), also known as inline programming takes a top-down approach.

It is about writing a list of instructions to tell the computer what to do step by step.

It relies on procedures or routines.

.