The Five Paradigms of Programming

Programming paradigms are different ways or styles to write or organize a code. Each programming paradigm has certain structures, features and opinions on how to solve programming problems. The five paradigms of programming are listed and explained below:

1. Imperative Programming: This consists sets of detailed structures that lets the computer how to execute a task step by step. This focuses on how things are done. E.g., providing the computer on the steps in baking a cake.

2. Procedural Programming: This is a type of imperative programming which allows programmers to organize their codes into procedures or functions. Example, using the cake analogy, this allows us to divide the steps in baking the cake into different procedures or function like Preparing the ingredients, Mixing the ingredients, and finally the oven instruction procedures.

3. Functional Programming: Programs designed here are for specific functions which has no changing state. Here, no changing variables. E.g. a math formula where same input always gives same output.

4. Declarative Programming: This consists of programs where you tell the computer what you want but not how to do it. Here, less code to write for complex task and it's easier to maintain and read.

5. Object-Oriented Programming (OOP): This consists of codes based on object (like real-world things) that have properties and behaviors. It allows modelling of real-world systems. Easier to maintain and scale and makes complex programs more understandable which involves separating responsibilities into different entities.