

# Programming Paradigm.

Ortiz Vega Angelo

Course Code: CE1103

Name: Algorithms and Data Structures I,  
Academic Area of Computer Engineering.  
Cartago, Costa Rica.

**-Keywords:** Programming Paradigm, Functional Paradigm, Imperative Paradigm, Logical Paradigm, Object-Oriented Paradigm.

**-Content:**

What is a paradigm? A paradigm is a typical example or pattern of something; a model. In a general context, it is a way of seeing the world.

In a computational context, it can be understood as a programming style. It defines in what form the programs are built.

Most recognized paradigms:

- A. Functional.
- B. Imperative
- C. Logical.
- D. Oriented to Objects.

**A. Functional:**

The world is modeled in terms of mathematical functions and function theory. Functional programming has its origins in lambda calculus, a formal system developed in the 1930s to

investigate computability, the Entscheidungsproblem, function definition, function application, and recursion. Many functional programming languages can be viewed as elaborations on the lambda calculus. Another well-known declarative programming paradigm, logic programming, is based on relations. Languages: Scheme, Lisp, F #

**B. Imperative:**

The real world is modeled with procedures, commands and structures. The term is often used in contrast to declarative programming, which focuses on what the program should accomplish without specifying how the program should achieve the result. Procedural programming could be considered a step towards declarative programming. A programmer can often tell, simply by looking at the names, arguments, and return types of procedures (and related comments), what a particular procedure is supposed to do, without necessarily looking at the details of how it achieves its result. At the same time, a complete program is still imperative since it fixes the

statements to be executed and their order of execution to a large extent.

Languages: C, C #, PHP.

### **C. Logical:**

The real world is modeled with logical predicates. Logic programming can be viewed as controlled deduction. An important concept in logic programming is the separation of programs into their logic component and their control component. With pure logic programming languages, the logic component alone determines the solutions produced. The control component can be varied to provide alternative ways of executing a logic program.

Languages: Prolog

### **D. Oriented to Objects.**

The real world is modeled as objects. Object-oriented features have been added to many previously existing languages, including Ada, BASIC, Fortran, Pascal, and COBOL. Adding these features to languages that were not initially designed for them often led to problems with compatibility and maintainability of code.

There are a number of other programming paradigms besides imperative, functional, and object-oriented. Logic programming, for example, defines a program in terms of a set of formal propositions.

Programming paradigm is a loosely defined concept, and many paradigms overlap one another. For example, declarative programming encompasses functional and logic programming, and is defined mostly in contrast to imperative programming. Ultimately, a programming paradigm is simply a big idea in programming, one that completely changes how we think about and write programs. Paradigms that were frequently discussed in the mid-20th century, such as structured programming, are rarely discussed today because their tenets have been taken up by the majority of programmers and programming languages.