Functional Programming - Introduction

Functional programming languages are specially designed to handle symbolic computation and list processing applications. Functional programming is based on mathematical functions. Some of the popular functional programming languages include: Lisp, Python, Erlang, Haskell, Clojure, etc.

Functional programming languages are categorized into two groups, i.e. ?

Pure Functional Languages ?
These types of functional languages support only the functional paradigms. For example ?
Haskell.

Impure Functional Languages ?
These types of functional languages support the functional paradigms and imperative style programming.
For example ? LISP.

Functional Programming – Characteristics

The most prominent characteristics of functional programming are as follows ?

Functional programming languages are designed on the concept of mathematical functions that use conditional expressions and recursion to perform computation.

Functional programming supports higher-order functions and lazy evaluation features.

Functional programming languages don’t support flow Controls like loop statements and conditional statements like If-Else and Switch Statements. They directly use the functions and functional calls.

Like OOP, functional programming languages support popular concepts such as Abstraction, Encapsulation, Inheritance, and Polymorphism.

Efficiency of a Program Code

The efficiency of a programming code is directly proportional to the algorithmic efficiency and the execution speed. Good efficiency ensures higher performance.

The factors that affect the efficiency of a program includes ?

The speed of the machine
Compiler speed
Operating system
Choosing right Programming language
The way of data in a program is organized
Algorithm used to solve the problem

The efficiency of a programming language can be improved by performing the following tasks ?

By removing unnecessary code or the code that goes to redundant processing.

By making use of optimal memory and nonvolatile storage

By making the use of reusable components
wherever applicable.

By making the use of error & exception handling at all layers of
program.

By creating programming code that ensures data integrity and consistency.

By
developing the program code that's compliant with the design logic and flow.

An efficient
programming code can reduce resource consumption and completion time as much as possible with
minimum risk to the operating environment.