Difference between Functional Programming and Object Oriented Programming

A programming paradigm is a style, or “way, ” of programming. Programming paradigms differ from one another based on the features and the style they support. There are several features that determine a programming paradigm such as modularity, objects, interrupts or events, control flow etc. Every programming paradigm has its own advantage so, it better to know where to use it before actually using it.

Object-oriented languages are good when you have a fixed set of operations on things, and as your code evolves, you primarily add new things. This can be accomplished by adding new classes which implement existing methods, and the existing classes are left alone.

Functional languages are good when you have a fixed set of things, and as your code evolves, you primarily add new operations on existing things. This can be accomplished by adding new functions which compute with existing data types, and the existing functions are left alone.

It is also possible to use both the programming paradigms according to our own need. As we have got languages like python, java, etc that supports both object oriented concept and are also functional by supporting various inbuilt functions.

**Functional Programming vs Object Oriented Programming**

| **FUNCTIONAL PROGRAMMING** | **OBJECT ORIENTED PROGRAMMING** |
| --- | --- |
| This programming paradigm emphasizes on the use of functions where each function performs a specific task. | This programming paradigm is based on object oriented concept. Classes are used where instance of objects are created |
| Fundamental elements used are variables and functions.The data in the functions are immutable(cannot be changed after creation). | Fundamental elements used are objects and methods and the data used here are mutable data. |
| Importance is not given to data but to functions. | Importance is given to data rather than procedures. |
| It follows declarative programming model. | It follows imperative programming model. |
| It uses recursion for iteration. | It uses loops for iteration. |
| It is parallel programming supported. | It does not support parallel programming. |
| The statements in this programming paradigm does not need to follow a particular order while execution. | The statements in this programming paradigm need to follow a order i.e., bottom up approach while execution. |
| Does not have any access specifier. | Has three access specifiers namely, Public, Private and Protected. |
| To add new data and functions is not so easy. | Provides and easy way to add new data and functions. |
| No data hiding is possible. Hence, Security is not possible. | Provides data hiding. Hense, secured programs are possible. |