What is object oriented programming?

Object oriented programming is a programming paradigm in which you program using objects to represent things you are programming about (sometimes real world things). These objects could be data structures. The objects hold data about them in attributes. The attributes in the objects are manipulated through methods or functions that are given to the object.

For instance, we might have a Person object that represents all of the data a person would have: weight, height, skin color, hair color, hair length, and so on. Those would be the attributes. Then the person object would also have things that it can do such as: pick box up, put box down, eat, sleep, etc. These would be the functions that play with the data the object stores.

Engineers who program using object oriented design say that it is a style of programming that allows you to model real world scenarios much simpler. This allows for a good transition from requirements to code that works like the customer or user wants it to.

### What is functional programming?

Functional programming is the form of programming that attempts to avoid changing state and mutable data. In a functional program, the output of a function should always be the same, given the same exact inputs to the function.

This is because the outputs of a function in functional programming purely relies on arguments of the function, and there is no magic that is happening behind the scenes. This is called eliminating side effects in your code.

For example, if you call function getSum() it calculates the sum of two inputs and returns the sum. Given the same inputs for x and y, we will always get the same output for sum.

This allows the function of a program to be extremely predictable. Each small function does its part and only its part. It allows for very modular and clean code that all works together in harmony. This is also easier when it comes to unit testing.

Some examples of Functional Programming Languages include Lisp, Clojure, and F#.

### Problems with object oriented programming

There are a few problems with object oriented programing. Firstly, it is known to be not as reusable. Because some of your functions depend on the class that is using them, it is hard to use some functions with another class.

It is also known to be typically less efficient and more complex to deal with. Plenty of times, some object oriented designs are made to model large architectures and can be extremely complicated.

### Problems with functional programming

Functional programming is not without its flaws either. It really takes a different mindset to approach your code from a functional standpoint. It’s easy to think in object oriented terms, because it is similar to how the object being modeled happens in the real world. Functional programming is all about data manipulation. Converting a real world scenario to just data can take some extra thinking.

Due to its difficulty when learning to program this way, there are fewer people that program using this style, which could make it hard to collaborate with someone else or learn from others because there will naturally be less information on the topic.

### A comparison between functional and object oriented programming

Both programming concepts have a goal of wanting to create easily understandable programs that are free of bugs and can be developed fast. Both concepts have different methods for storing the data and how to manipulate the data. In object oriented programming, you store the data in attributes of objects and have functions that work for that object and do the manipulation. In functional programming, we view everything as data transformation. Data is not stored in objects, it is transformed by creating new versions of that data and manipulating it using one of the many functions.