What is OOP?

In this lesson, we'll learn about the historical background of OOP and also key features of object-oriented programming.

WE'LL COVER THE FOLLOWING

Historical Background

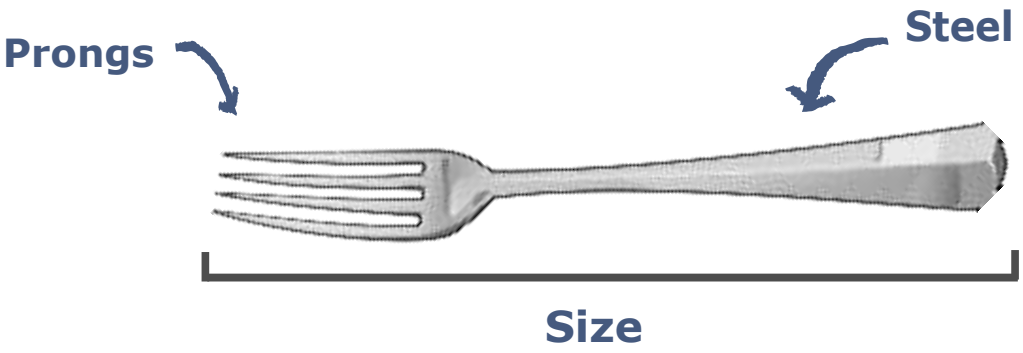
It was in the `60's` and early `70's` when the idea of **object-oriented** programming started occupying the minds of programmers. Keeping in view the benefits acquired through object orientation in SIMULA, the scientists encouraged languages using the same approach for programming. **Object-oriented** programming was a complete paradigm shift from the popular structural programming.

Object: A Fundamental Entity

Object-oriented programming is based on the idea of an `object`. An `object` is an entity with some *data* and *operations*. Data is also referred to as *properties* of the object whereas operations include accessing and modifying those properties along with other functions that depict the behavior of the object.

Example

A **fork** is an object with properties including a number of *prongs*, its *size*, and *material* (made of plastic or metal), etc. Behavior and functions of a fork include shredding, squashing, making design or may be simply eating.



Fork

Explanation

Programmers realized that when we represent entities in the program as objects, having their behaviors and properties, it becomes easy to deal with the increasing code complexity as well as the code becomes more reusable. So, a *fork* object can be part of a dinner set, and a similar object may also be sold separately. Once, we know its properties and behavior, we only need to reuse the same piece of code whenever a *fork* is needed.

In the next lesson, we'll learn about how C++ is an object-oriented programing.