Lectura 2: **Explain OOP Like I'm 5**

Miming pattern that is built around objects or entities, so **it**'s called object-oriented programming.

To better understand the concept, let's have a look at commonly used software programs: A good example to explain **this** would be the use of a printer when you are printing a document.

The first step is initiating the action by clicking on the print command or using keyboard shortcuts. Next you need to select your printer. Afterwards you will wait for a response telling you if the document was printed or not.

Behind what we can't see, the command you clicked interacts with an object (printer) to accomplish the task of printing.

Perhaps you might wonder, how exactly did OOP become so popular?

How OOP Became Popular

The concepts of OOP started to surface back in the 60s with a programming language called Simula. Even though back in the day, developers didn't completely embrace the first advances in OOP languages, the methodologies continued to evolve.

Fast forward to the 80s, and an editorial written by David Robinson was one of the first introductions to OOP, as many developers didn't know **it** existed.

By now languages like C++ and Eiffel had become more popular and mainstream among computer programmers. The recognition continued to grow during the 90s, and with the arrival of Java, OOP attracted a huge following.

In 2002 in conjunction with the release of the .NET Framework, Microsoft introduced a brand new OOP language called C# – **which** is often described as the most powerful programming language.

It's interesting that, generations later, the concept of organizing your code into meaningful objects that model the parts of your problem continues to puzzle programmers.

Many folks who haven't any idea how a computer works find the thought of object-oriented programming quite natural. In contrast, many folks who have experience with computers initially think there's something strange about object oriented systems.