C++ as Object Oriented Language

In this lesson, we'll learn about the evolution of object-oriented programming in C++.

WE'LL COVER THE FOLLOWING

- Origin of OOP in C++
- Concepts in OOP
- Preference over other Languages
- Author’s Words

Origin of OOP in C++

`C` is a procedural language and it was used and trusted by programmers at large. When the **object-oriented** paradigm was being introduced, *Bjarne Stroustrup* incorporated the object orientation features in `C` and hence introduced `C++`. It was initially introduced as **C with classes** and later renamed to C++.

Concepts in OOP

Soon `C++` became widely acceptable object-oriented language. To introduce an object-oriented approach in `C` *Stroustrup* introduced concepts of *Classes*, *objects*, *inheritance*, *dynamic binding*, and *polymorphism*. We will be discussing each one of them later in our chapters in detail.

Preference over other Languages

`C++` was successfully used in many applications including embedded systems, system programming, as well as for higher level programming tasks. Some people though argue that `C++` is not a pure object-oriented Language. Reason being the use of `main()` function in `C++` that can exist without any Class. Since making a Class is not a primary requirement in `C++` hence it is considered a partial object-oriented language. However, the choice and use of language depend on the programmer and his nature of the problem. To-date many

programmers still use `C++` as object-oriented language regardless of other Pure object-oriented languages available.

Author’s Words

If you are interested in reading more about `C++` as an **object-oriented** language, a highly recommended text is Why C++ is not just an Object Oriented Language by *Bjarne Stroustrup*. A little old text but it gives you the creator’s own perspective of explaining `C++` as a language.

In the next lesson, we’ll learn about the difference between three popular languages which use the object-oriented approach.