**C++ Polymorphism**

**Polymorphism in C++**

Poly means several and morphism means form. Polymorphism is something that has several forms or we can say it as one name and multiple forms. There are two types of polymorphism:

* Compile-time polymorphism
* Run time polymorphism

**Compile Time Polymorphism**

In compile-time polymorphism, it is already known which function will run. Compile-time polymorphism is also called early binding, which means that you are already bound to the function call and you know that this function is going to run.

There are two types of compile-time polymorphism:

* **Function Overloading**

This is a feature that lets us create more than one function and the functions have the same names but their parameters need to be different. When function overloading is implemented in a program and function calls are made, the compiler knows which functions to execute.

* **Operator Overloading**

This is a feature that lets us define operators working for some specific tasks. Using the operator +, we can add two strings by concatenating and add two numerical values arithmetically at the same time. This is operator overloading.

**Run Time Polymorphism**

With run-time polymorphism, the compiler has no idea what will happen when the code is executed. Run time polymorphism is also called late binding. The run-time polymorphism is considered slow because function calls are decided at run time. Run time polymorphism can be achieved from virtual functions.

**Virtual Functions in C++**

A member function in the base class that is declared using a virtual keyword is called a virtual function. They can be redefined in the derived class. A function that is in the parent class but redefined in the child class is called a virtual function.

For a function to become virtual, it should not be static.