Polymorphism

- Polymorphism is a core concept in object-oriented programming (OOP) and refers to the ability of a function, method, or operator to behave in multiple ways depending on the context. In C++, polymorphism is achieved through inheritance and dynamic dispatch, enabling methods in derived classes to override base class methods and allow objects of different types to be treated uniformly.

Types of Polymorphism in C++

In C++, polymorphism can be categorized into two types:

1. Compile-time Polymorphism (also known as Static Polymorphism)
2. Runtime Polymorphism (also known as Dynamic Polymorphism)

---

1. **Compile-time Polymorphism (Static Binding)**

> Compile-time polymorphism is resolved during compilation. In C++, the most common examples of compile-time polymorphism are:

- Function Overloading
- Operator Overloading

Function Overloading

> Function overloading allows multiple functions with the same name but different parameter lists (either different number of parameters or different types). The function to be invoked is determined at compile time based on the arguments passed to it.

Operator Overloading

> Operator overloading allows you to define or modify the behavior of operators for user-defined data types (like classes).