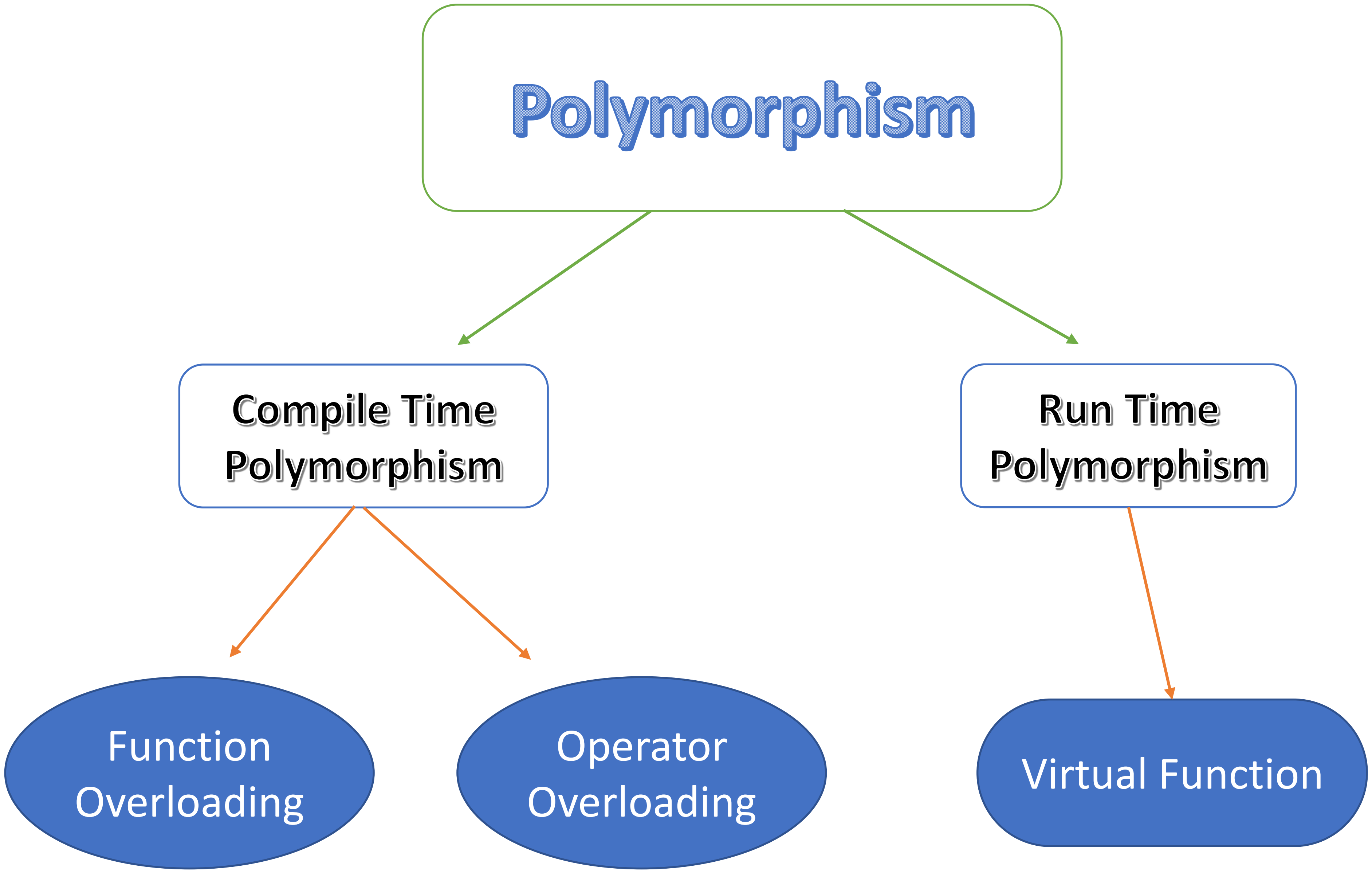
Polymorphism
Compile Time Polymorphism
Run Time Polymorphism
Function Overloading
Operator Overloading
Virtual Function

➢ Polymorphism

- One name and multiple forms
- Eg.- Function Overloading , Operator Overloading
- Eg.- Virtual Function

▪ Polymorphism in C++ can be of two types:

1.Compile Time Polymorphism

Compile time polymorphism in C++ is achieved using :

1.1 – Function Overloading

1.2 – Operator Overloading

2.Run Time Polymorphism

2.1 – Virtual Function

Polymorphism in C++
“Poly” means several and “morphism” means form. So we can say that 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:
1.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. If function overloading is done in the program and function calls are made the compiler already knows that which functions to execute.
2.Operator Overloading
This is a feature that lets us define operators working for some specific tasks. For example, we can overload the operator “+” and define its functionality to add two strings. Operator loading is also an example of compile-time polymorphism because the compiler already knows at the compile time which operator has to perform the task.
Run Time Polymorphism
In the run-time polymorphism, the compiler doesn’t know already what will happen at run time. 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 the virtual function.
3.Virtual Function
A function that is in the parent class but redefined in the child class is called a virtual function. “virtual” keyword is used to declare a virtual function.