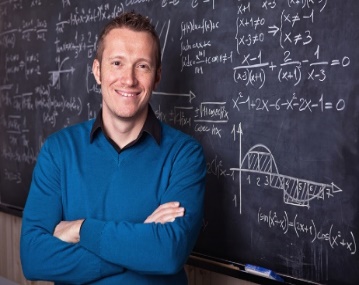
**Polymorphism In C#**

* Polymorphism is one of the four pillars of Object Oriented Programming.
* Polymorphism in C# is a concept by which we can perform a single action by different ways.
* Polymorphism is derived from 2 Greek words: POLY and MORPHS.
* The word "poly" means **many** and "morphs" means **forms**.
* So polymorphism means many forms.

**Real World Example**

FATHER

SON

HUSBAND

TEACHER

**There Are Two Types Of Polymorphism**

1. Static Polymorphism (Compile Time Polymorphism)
2. Dynamic Polymorphism (Run Time Polymorphism)

**Static Polymorphism (Compile Time Polymorphism) In C#**

* The mechanism of linking a function with an object during compile time is called static polymorphism or early binding.
* It is also called static binding.

C# provides two techniques to implement static polymorphism. They are −

* Method Or Function Overloading
* Operator Overloading

**Method Or Function Overloading**

* You can have multiple definitions for the same function name in the same scope.
* The definition of the function must differ from each other by the types and/or the number of arguments in the argument list.
* You cannot overload function declarations that differ only by return type.

**Dynamic Or Runtime Polymorphism In C#**

* Run time polymorphism is achieved by method overriding.
* Method overriding allows us to have virtual and abstract methods in the base using derived classes with the same name and the same parameter.

**C# Method Overriding**

* If derived class defines same method as defined in its base class, it is known as method overriding in C#.
* It is used to achieve runtime polymorphism.
* It enables you to provide specific implementation of the method in child class which is already provided by its base class.
* To perform method overriding in C#, you need to use **virtual** keyword with base class method and **override** keyword with derived class method.
* A method declared using the virtual keyword is referred to as a virtual method.
* In the derived class, you need to declare the inherited virtual method using the override keyword.
* In the derived class, you need to declare the inherited virtual method using the override keyword which is mandatory for any virtual method that is inherited in the derived class.
* The override keyword overrides the base class method in the derived class.