

# C# Generics

Generic is a concept that allows us to define classes and methods with placeholder. C# compiler replaces these placeholders with specified type at compile time. The concept of generics is used to create general purpose classes and methods.

o define generic class, we must use angle <> brackets. The angle brackets are used to declare a class or method as generic type. In the following example, we are creating generic class that can be used to deal with any type of data.

## C# Generic class example

```
1. using System;
2. namespace CSharpProgram
3. {
4.     class GenericClass<T>
5.     {
6.         public GenericClass(T msg)
7.         {
8.             Console.WriteLine(msg);
9.         }
10.    }
11.    class Program
12.    {
13.        static void Main(string[] args)
14.        {
15.            GenericClass<string> gen  = new GenericClass<string> ("This is generic class");
16.            GenericClass<int>   genI  = new GenericClass<int>(101);
17.            GenericClass<char>  getCh = new GenericClass<char>('I');
18.        }
19.    }
20. }
```

Output:

```
This is generic class
101
I
```

C# allows us to create generic methods also. In the following example, we are creating generic method that can be called by passing any type of argument.

## Generic Method Example

```
1. using System;
2. namespace CSharpProgram
3. {
4.     class GenericClass
5.     {
```

```
6.     public void Show<T>(T msg)
7.     {
8.         Console.WriteLine(msg);
9.     }
10. }
11. class Program
12. {
13.     static void Main(string[] args)
14.     {
15.         GenericClass genC = new GenericClass();
16.         genC.Show("This is generic method");
17.         genC.Show(101);
18.         genC.Show('I');
19.     }
20. }
21. }
```

Output:

```
This is generic method
101
I
```