C# Member Overloading

If we create two or more members having same name but different in number or type of parameter, it is known as member overloading. In C#, we can overload:

- methods,
- constructors, and
- indexed properties

It is because these members have parameters only.

C# Method Overloading

Having two or more methods with same name but different in parameters, is known as method overloading in C#.

The **advantage** of method overloading is that it increases the readability of the program because you don't need to use different names for same action.

You can perform method overloading in C# by two ways:

1. By changing number of arguments
2. By changing data type of the arguments

C# Method Overloading Example: By changing no. of arguments

Let's see the simple example of method overloading where we are changing number of arguments of add() method.

```
1. using System;
2. public class Cal{
3.     public static int add(int a,int b){
4.         return a + b;
5.     }
6.     public static int add(int a, int b, int c)
7.     {
8.         return a + b + c;
9.     }
10. }
11. public class TestMemberOverloading
12. {
13.     public static void Main()
14.     {
15.         Console.WriteLine(Cal.add(12, 23));
16.         Console.WriteLine(Cal.add(12, 23, 25));
17.     }
18. }
```

Output:

```
35
60
```

C# Member Overloading Example: By changing data type of arguments

Let's see the another example of method overloading where we are changing data type of arguments.

```
1. using System;
2. public class Cal{
3.     public static int add(int a, int b){
4.         return a + b;
5.     }
6.     public static float add(float a, float b)
7.     {
8.         return a + b;
9.     }
10. }
11. public class TestMemberOverloading
12. {
13.     public static void Main()
14.     {
15.         Console.WriteLine(Cal.add(12, 23));
16.         Console.WriteLine(Cal.add(12.4f,21.3f));
17.     }
18. }
```

Output:

```
35
33.7
```