CAll BY VALUE AND CALL BY REFERENCE

Functions can be invoked in two ways: **Call by Value** or **Call by Reference**. These two ways are generally differentiated by the type of values passed to them as parameters.

The parameters passed to function are called *actual parameters* whereas the parameters received by function are called *formal parameters*.

Call By Value:

In this parameter passing method, values of actual parameters are copied to function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of the caller.

Example

```
using System;
namespace Func_Example
{
    class Example
    {
        public void Display(int a)
        {
             a += a;
            Console.WriteLine("Value inside the function: "+ a);
        }
        static void Main(string[] args)
        {
            int a = 100;
            Example xyz = new Example();
            Console.WriteLine("Value before calling: "+ a);
            xyz.Display(a);
            Console.WriteLine("Value after calling: " + a);
        }
    }
}
```

Output:

```
Value before calling: 100
Value inside the function: 200
Value after calling: 100
```

Call by Reference:

Both the actual and formal parameters refer to the same locations, so any changes made inside the function are actually reflected in actual parameters of the caller.

C# provides a **ref** keyword to pass arguments as reference-type. It passes reference of arguments to the function rather than a copy of the original value. The changes in passed values are permanent and **modify** the original variable value.

Example:

```
using System;
namespace CallByReference
{
   class Program
   {
      // User defined function
      public void Show(ref int val)
      {
          val *= val; // Manipulating value
         Console.WriteLine("Value inside the show function "+val);
         // No return statement
      }
      // Main function, execution entry point of the program
      static void Main(string[] args)
      {
         int val = 50;
         Program program = new Program(); // Creating Object
         Console.WriteLine("Value before calling the function "+val);
         program.Show(ref val); // Calling Function by passing reference
         Console.WriteLine("Value after calling the function " + val);
      }
   }
}
```

Output:

```
Value before calling the function 50
Value inside the show function 2500
Value after calling the function 2500
```