C# - PASSING PARAMETERS BY VALUE

https://www.tutorialspoint.com/csharp/csharp_value_parameters.htm

This is the default mechanism for passing parameters to a method. In this mechanism, when a method is called, a new storage location is created for each value parameter.

The values of the actual parameters are copied into them. Hence, the changes made to the parameter inside the method have no effect on the argument. The following example demonstrates the concept –

Live Demo

```
using System;

namespace CalculatorApplication {
   class NumberManipulator {
      public void swap(int x, int y) {
         int temp;
         
         temp = x; /* save the value of x */
         x = y;    /* put y into x */
         y = temp; /* put temp into y */
      }
      static void Main(string[] args) {
         NumberManipulator n = new NumberManipulator();
         
         /* local variable definition */
         int a = 100;
         int b = 200;
         
         Console.WriteLine("Before swap, value of a : {0}", a);
         Console.WriteLine("Before swap, value of b : {0}", b);
         
         /* calling a function to swap the values */
         n.swap(a, b);
         
         Console.WriteLine("After swap, value of a : {0}", a);
         Console.WriteLine("After swap, value of b : {0}", b);
         
         Console.ReadLine();
      }
   }
}
```

When the above code is compiled and executed, it produces the following result –

```
Before swap, value of a :100
Before swap, value of b :200
After swap, value of a :100
After swap, value of b :200
```

It shows that there is no change in the values though they had changed inside the function.