

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



## راهنمای برنامه نویسی شی گرا در C#

**\* درس پنجم \***

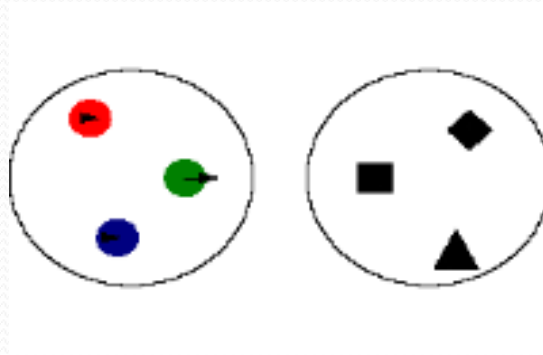
مدرس : نوید همراهی

[www.navidhamrahi.ir](http://www.navidhamrahi.ir)

[n.hamrahi@gmail.com](mailto:n.hamrahi@gmail.com)

## متدها و توابع :

تابع رابطه‌ای است که رابطه بین اعضای یک مجموعه را با اعضای دیگر از مجموعه‌ای دیگر را بیان می‌کند.



**\* شکستن یک کار بزرگ به چندین کار کوچک با وظیفه‌های مشخص**

## تعريف توابع در C# :

**Return Type**   **Method Name**   (**[Parameters]**)

**1**

**2**

**3**

۱ - مقدار بازگشتی تابع

۲ - نام تابع

۳ - پارامترهای ورودی

## انواع تابع :

- دسته بندی توابع از لحاظ **دسترسی** : سراسری – محلی
- دسته بندی توابع از لحاظ **مقدار بازگشتی** : بدون مقدار – با مقدار
- دسته بندی توابع از لحاظ **پارامتر ورودی** : بدون پارامتر – با پارامتر

## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }
        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }

        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

**تابع تعریف شده:**

## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }
        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

علت بدون مقدار بازگشتی:



## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }
        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

علت بدون پارامتر: 

## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }
        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

علت سراسری :

static

## مثال ۱: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت سراسری

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Main");
            Test();
            Console.ReadKey();
        }
        static void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

Test();

نحوه فراخوانی تابع :

## مثال ۲: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت محلی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Program TestFunction = new Program() ;
            Console.WriteLine("Main");
            TestFunction.Test();
            Console.ReadKey();
        }
        void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

## مثال ۲: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت محلی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Program TestFunction = new Program() ;
            Console.WriteLine("Main");
            TestFunction.Test();
            Console.ReadKey();
        }
        void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

علت محلی :

void Test() خالی

## مثال ۲: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت محلی

```
namespace ConsoleApplication9
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main()
```

```
        {
```

```
            Program TestFunction = new Program() ;
```

```
            Console.WriteLine("Main");
```

```
            TestFunction.Test();
```

```
            Console.ReadKey();
```

```
        }
```

```
    void Test()
```

```
    {
```

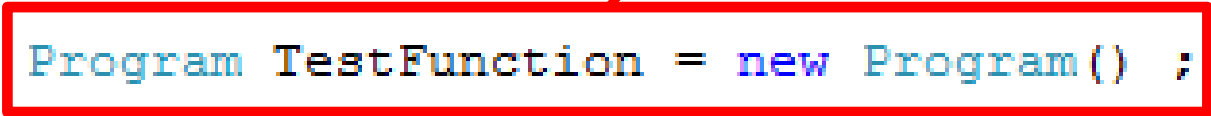
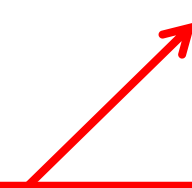
```
        Console.WriteLine("Test");
```

```
    }
```

```
}
```

```
}
```

نحوه تعریف تابع در تابع :



## مثال ۲: تعریف تابع بدون مقدار بازگشتی و بدون پارامتر ورودی به صورت محلی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            Program TestFunction = new Program() ;
            Console.WriteLine("Main");
            TestFunction.Test();
            Console.ReadKey();
        }
        void Test()
        {
            Console.WriteLine("Test");
        }
    }
}
```

نحوه فراخوانی تابع :

## مثال ۳: تعریف تابع با مقدار بازگشتی و بدون پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            string Temp;
            Console.WriteLine("Main");
            Temp = Test();
            Console.WriteLine("{0}", Temp);
            Console.ReadKey();
        }
        static string Test()
        {
            string YourName;
            YourName = "Test";
            return YourName;
        }
    }
}
```



## مثال ۳: تعریف تابع با مقدار بازگشتی و بدون پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            string Temp;
            Console.WriteLine("Main");
            Temp = Test();
            Console.WriteLine("{0}", Temp);
            Console.ReadKey();
        }
        static string Test()
        {
            string YourName;
            YourName = "Test";
            return YourName;
        }
    }
}
```

نوع مقدار بازگشتی:

## مثال ۳ : تعریف تابع با مقدار بازگشتی و بدون پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            string Temp;
            Console.WriteLine("Main");
            Temp = Test();
            Console.WriteLine("{0}", Temp);
            Console.ReadKey();
        }
        static string Test()
        {
            string YourName;
            YourName = "Test";
            return YourName;
        }
    }
}
```

بازگشت مقدار خروجی :



## مثال ۳ : تعریف تابع با مقدار بازگشتی و بدون پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            string Temp;
            Console.WriteLine("Main");
            Temp = Test();
            Console.WriteLine("{0}", Temp);
            Console.ReadKey();
        }
        static string Test()
        {
            string YourName;
            YourName = "Test";
            return YourName;
        }
    }
}
```

نحوه فراخوانی تابع :

## مثال ۴ : تعریف تابع با مقدار بازگشتی و با پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            int A;int B;int C;
            Console.WriteLine("Main");
            A = 10;B = 24;
            C = Test(A, B);
            Console.WriteLine("{0}", Convert.ToString(C));
            Console.ReadKey();
        }
        static int Test(int Num1 , int Num2)
        {
            int Sum;
            Sum = Num1 + Num2;
            return Sum;
        }
    }
}
```

## مثال ۴ : تعریف تابع با مقدار بازگشتی و با پارامتر ورودی

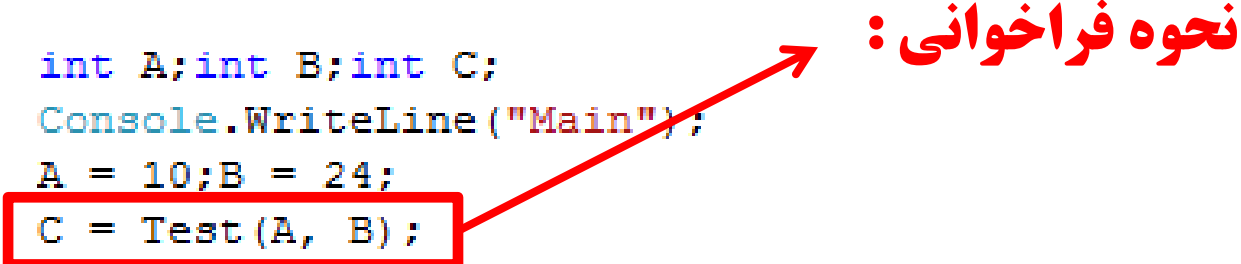
```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            int A;int B;int C;
            Console.WriteLine("Main");
            A = 10;B = 24;
            C = Test(A, B);
            Console.WriteLine("{0}", Convert.ToString(C));
            Console.ReadKey();
        }
        static int Test(int Num1 , int Num2)
        {
            int Sum;
            Sum = Num1 + Num2;
            return Sum;
        }
    }
}
```

پارامتر ورودی :

## مثال ۴ : تعریف تابع با مقدار بازگشتی و با پارامتر ورودی

```
namespace ConsoleApplication9
{
    class Program
    {
        static void Main()
        {
            int A;int B;int C;
            Console.WriteLine("Main");
            A = 10;B = 24;
            C = Test(A, B);
            Console.WriteLine("{0}", Convert.ToString(C));
            Console.ReadKey();
        }
        static int Test(int Num1 , int Num2)
        {
            int Sum;
            Sum = Num1 + Num2;
            return Sum;
        }
    }
}
```

نحوه فراخوانی :



## مثال ۵: تعریف تابع در یک کلاس مجزا

```
namespace ConsoleApplication9
{
    class Test
    {
        public String PrintName ()
        {
            string MyName;
            MyName = "Test";
            return MyName;
        }
    }
    class Program
    {
        static void Main ()
        {
            string Temp;
            Test PrintFunction = new Test ();
            Console.WriteLine ("Main");
            Temp = PrintFunction.PrintName ();
            Console.WriteLine ("{0}", Temp);
            Console.ReadKey ();
        }
    }
}
```

```
namespace ConsoleApplication9
```

```
{
```

```
class Test
```

```
{
```

```
public String PrintName()
```

```
{
```

```
string MyName;
```

```
MyName = "Test";
```

```
return MyName;
```

```
}
```

```
}
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

```
string Temp;
```

```
Test PrintFunction = new Test();
```

```
Console.WriteLine("Main");
```

```
Temp = PrintFunction.PrintName();
```

```
Console.WriteLine("{0}", Temp);
```

```
Console.ReadKey();
```

```
}
```

```
}
```

```
}
```

**مثال ۵: تعریف تابع در یک کلاس مجزا:**

**نحوه تعریف تابع در کلاس:**



## مثال ۵: تعریف تابع در یک کلاس مجزا:

```
namespace ConsoleApplication9
{
    class Test
    {
        public String PrintName()
        {
            string MyName;
            MyName = "Test";
            return MyName;
        }
    }
}
```

```
class Program
```

```
{
    static void Main()
    {
        string Temp;
        Test PrintFunction = new Test();
        Console.WriteLine("Main");
        Temp = PrintFunction.PrintName();
        Console.WriteLine("{0}", Temp);
        Console.ReadKey();
    }
}
```

نحوه فراخوانی:

سوال ؟



## تمرین عملی :

۱ – برنامه ای بنویسید که  $N!$  را در قالب یک تابع اجرا نماید .

راهنمایی: یک عدد بعنوان  $N$  را از ورودی دریافت کرده و با ارسال آن به تابع مقدار فاکتوریل را محاسبه و سپس در خروجی چاپ نمایید .



1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
... n

۲ – برنامه ای بنویسید که با دریافت یک عدد از ورودی مثلث زیر را چاپ نماید :  
( محاسبه هر سطر در تابع صورت گیرد )