

## ref and out parameters

C# provides keywords when you defining a method signature:

1. out
2. ref

By default, method arguments are passed by value.

That means calling method pass a duplicate copy of data to the called method. If called method modifies the data, it will not visible to calling method.

By using ref and out changes done by called method will be visible to calling method. Params is used for passing any number of arguments to called method.

## Ref Keyword

The *ref* keyword passes arguments to calling method by reference. But before calling the method, the arguments must be initialized with some values. Below is the example of ref keyword.

```
static void Main(string[] args)
{
    int par = 2;

    RefMethod(ref par);

    Console.WriteLine(par);

    //Result: 45
}

static void RefMethod(ref int i)
{
    i = 45;
}
```

In the above example, we have declared *par* parameter. We initialized par parameter with "2" value. If we don't initialize the parameter, the compiler will issue an error of "Use of unassigned local variable 'par' ". That means we must initialize the parameters before calling the method.

***There is a scenario, when we don't know the parameter value before calling the method. But according to C#, we cannot call a method without initializing its parameters. In that case we have no option and we must initialize the parameters.***

## Out Keyword

Out keyword solves the above scenario. We need not to initialize the parameter before calling the method. But C# compiler puts a constraint on the called method, that parameter must be initialize within the method.

```
static void Main(string[] args)
{
    int par;

    RefMethod(out par);

    Console.WriteLine(par);
}
```

```
static void RefMethod(out int i)
{
    i = 45;
}
```

In the Main method, we have not initialized the par parameter. In the Main method, we used *out* keyword before parameter. Same is done in RefMethod method and we use *out* keyword before the *i* parameter. In the RefMethod, if we don't initialize the *i* parameter, C# compiler will issue a error "The out parameter 'i' must be assigned to before leaves the current method". For example below code will not work.

```
static void Main(string[] args)
{
    int par;

    RefMethod(out par);

    Console.WriteLine(par);
}

static void RefMethod(out int i)
{
    //i = 45;    //Compile time error occurred
}
```

## Ref vs Out differences

	Ref	Out
1.	ref is useful when we already know the parameter value and called method can only modify the data.	out is useful when we don't know the parameter value before calling the method.
2.	Called method has no responsibility to initialize the parameter.	Called method has the responsibility to initialize the parameter.
3.	ref allows us to pass data in both directions parent to child method and child to parent method.	out allows us to pass data only one direction child to parent method.
4.	ref should not be used to return multiple values from method.	out should be used to return multiple values from method.
5.	There is no restriction that called method must work on ref parameter. Called method may not use the ref parameter.	Called method must work on out parameter and before returning must initialize the parameter.