Ref and Out Parameters in C#

Ref Parameters

1. **Purpose**: `ref` is used to pass arguments by reference. This allows the method to read and modify the value of the parameter.

2. **Initialization**: The variable passed as a `ref` parameter must be initialized before it is passed to the method.

3. **Usage**: When defining a method with a `ref` parameter, the parameter is preceded by the `ref` keyword. Similarly, when calling the method, the `ref` keyword must also be used.

4. **Example**:

```csharp
public void ModifyValue(ref int number)
{
    number = number + 10;
}

public void Example()
{
    int myNumber = 5;
    ModifyValue(ref myNumber);
    Console.WriteLine(myNumber); // Outputs 15
}
```

Out Parameters

1. **Purpose**: `out` is used to pass arguments by reference and is typically used when a method needs to return multiple values. Unlike `ref`, `out` parameters are intended to be used for output purposes only.

2. **Initialization**: The variable passed as an `out` parameter does not need to be initialized before it is passed to the method. However, the method is required to assign a value to the `out` parameter before it returns.

3. **Usage**: When defining a method with an `out` parameter, the parameter is preceded by the `out` keyword. Similarly, when calling the method, the `out` keyword must also be used.

4. **Example**:

```csharp
public void GetValues(out int number, out string text)
{
    number = 42;
    text = "Hello";
}

public void Example()
{
    int myNumber;
    string myText;
    GetValues(out myNumber, out myText);
    Console.WriteLine(myNumber); // Outputs 42
    Console.WriteLine(myText); // Outputs "Hello"
}
```

Key Differences
- **Initialization**: `ref` parameters must be initialized before being passed to the method, whereas `out` parameters do not need to be initialized beforehand.

- **Requirement to Assign**: Methods using `out` parameters must assign a value to them before exiting, whereas methods using `ref` parameters are not required to assign a new value.

- **Intent**: `ref` is generally used when the parameter is meant to be read and modified, while `out` is used when the parameter is meant to be written to, effectively functioning as an additional return value.