Using Parameters in Methods

Introduction

Parameters in methods allow flexibility by enabling data (arguments) to be passed into the method. This allows methods to operate on different inputs, making the code more reusable and adaptable.

What Are Parameters?

Parameters are variables defined within a method’s signature that specify what input the method can accept. They act as placeholders for the values (called arguments) that will be passed into the method when it is called. This makes methods more flexible, allowing them to perform specific tasks based on different inputs without needing to rewrite the method.

Types of Parameters

Value Parameters

- Receive a copy of the argument.
- Changes to the parameter inside the method don’t affect the original argument.

Reference Parameters

- Receive a reference to the original argument.
- Changes made inside the method affect the original value.

How to Pass Data into Methods

The method signature defines parameters, and arguments matching the parameter types are passed when the method is called.

Example: Single Parameter

```
public void GreetUser(string userName) {
    MessageBox.Show("Hello, " + userName + "!");
}
GreetUser("Alice");
```

Output: "Hello, Alice!".

Example: Multiple Parameters

```
public void GreetUser(string userName, int lastLogin) {
    MessageBox.Show("Hello, " + userName + "! It's been " + lastLogin + " days since you last logged in.");
}
GreetUser("Alice", 2);
```

Output: "Hello, Alice! It's been 2 days since you last logged in".

Conclusion

Parameters allow methods to handle a variety of inputs, enabling more flexible and reusable code. Understanding how to define and pass parameters is fundamental to creating adaptable methods.