Functions/Methods

What is a function?

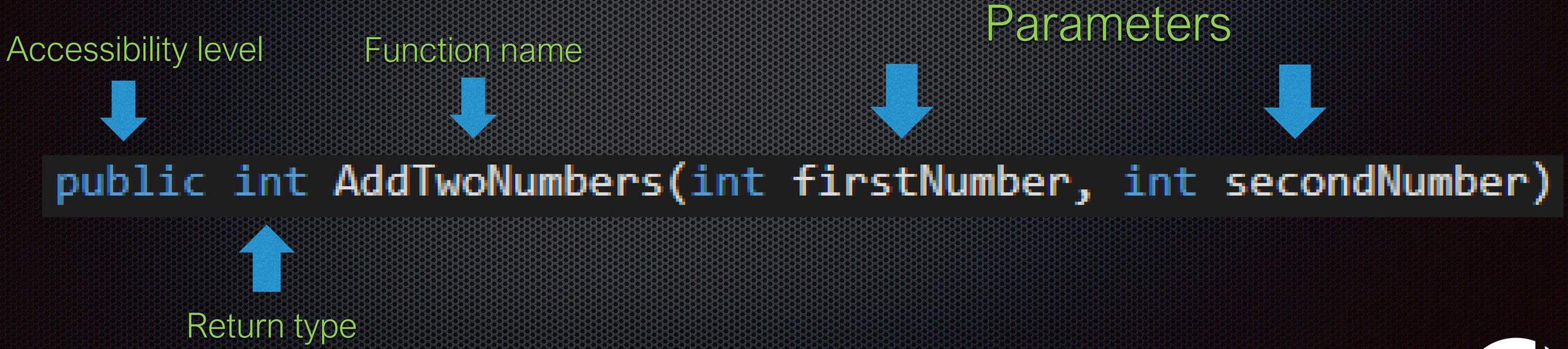
- Functions are small, reusable pieces of code
- A function should perform one task, and perform it well
- In general, if a function is more than five lines, you should consider seeing if it can be split into more than one function.

Difference Between Function and Method

- There is no functional difference between a function and a method.
- When a function is a part of a class, it is called a method.
- C# is an object-oriented programming language. That is why all functions are called methods.

Method Signature

- Methods have a signature, that consists of the accessibility level, return type, function name and parameters



Access Modifiers

- Access modifiers specify the accessibility of member or type
- public: can be accessed by any other code. Its access is never restricted
- private: can only be accessed by code in the same class
- protected: similar to private. Difference is it can be accessed in a child class (any class that inherits from the parent class)

Methods

- It is possible for a method to have no parameters
- The signature is the only part of the method that is ‘visible’ to the outside world
- Parameters are how we provide “outside” data to the method
- Methods also have a code block, this is the code that executes when the method is called

Method Example

```
public int AddTwoNumbers(int firstNumber, int secondNumber)
{
    int result;
    result = firstNumber + secondNumber;
    return result;
}
```

Calling a method

- Methods don’t do anything unless they are used (called)
- To call a method, use the method name and provide the required arguments

Calling a method

```
int myNumber = 300;
int yourNumber = 400;
int ourNumber = AddTwoNumbers(myNumber, yourNumber);
Console.WriteLine(ourNumber);
```

Calling a method

- It is possible to call the same method as many times as needed, just store different values in different variables
- This is why it is important to keep each method as simple as possible
- Simple methods are more likely to be useful in other situations

Return keyword

- In the code example, we used the “return” keyword
- Return sends the results of the method back to the caller
- This is why we need a variable to contain the result
- In the code example, we assigned the result of AddTwoNumbers() to the variable ourNumber

Void Methods

- Methods don’t have to return a result
- A method that doesn’t return a value is called a void method. The “void” keyword is used as the return type in a method signature.

Void Method Example

```
public void AddThreeNumbers(int firstNumber, int secondNumber, int thirdNumber)
{
    int result;
    result = firstNumber + secondNumber + thirdNumber;
    Console.WriteLine("This result is from a void function: " + result);
}
```

Void Method Example

```
AddThreeNumbers(1, 2, 3);
```

Call method

```
This result is from a void function: 6
```

Output

Review Questions

- What is the difference between a function and a method?
- What are public, private, protected? What is the difference between them?
- What is a return type?
- When don’t you have to store a value in a variable when calling a method?
- Why would you store a value in a variable when calling a method?

Assignment

- Refactor your pay calculator to use methods and take user input
- User input will be in hours worked and hourly rate