7.1 — Function parameters and arguments

BY ALEX ON JULY 18TH, 2007 | LAST MODIFIED BY ALEX ON APRIL 17TH, 2016

In Chapter 1, we covered function basics in the following sections:

- **1.4 -- A first look at functions and return values**
- **1.4a -- A first look at function parameters and arguments**
- **1.7 -- Forward declarations and definitions**
- **1.8 -- Programs with multiple files**
- **1.9 -- Header files**

You should be familiar with the concepts discussed in those lessons before proceeding.

Parameters vs Arguments

In the next three lessons, we'll talk quite a bit about parameters and arguments, so let's revisit those definitions before proceeding.

In common usage, the terms parameter and argument are often interchanged. However, for the purposes of further discussion, we will make a distinction between the two:

A **function parameter** (sometimes called a **formal parameter**) is a variable declared in the function declaration:

```
void foo(int x); // declaration (function prototype) -- x is a parameter

void foo(int x) // definition (also a declaration) -- x is a parameter
{
}
```

An **argument** (sometimes called an **actual parameter**) is the value that is passed to the function by the caller:

```
foo(6); // 6 is the argument passed to parameter x
foo(y+1); // the value of y+1 is the argument passed to parameter x
```

When a function is called, all of the parameters of the function are created as variables, and the value of the arguments are copied into the parameters. For example:

```
void foo(int x, int y)
{
}

foo(6, 7);
```

When foo() is called with arguments 6 and 7, foo's parameter x is created and assigned the value of 6, and foo's parameter y is created and assigned the value of 7.

Even though parameters are not declared inside the function block, function parameters have local scope. This means that they are created when the function is invoked, and are destroyed when the function block terminates:

```
void foo(int x, int y) // x and y are created here
{
} // x and y are destroyed here
```

There are 3 primary methods of passing arguments to functions: pass by value, pass by reference, and pass by address. We'll look at each of those in the next set of lessons.

7.2 -- Passing arguments by value