

The Function Body

The body of the function is a block consisting of the statements that implement the function, along with the declarations of any local variables. For functions that **return a value** to their caller, at least one of those statements must be a **return** statement:

```
return expression;
```

Executing the **return** statement causes the function to immediately to return to its caller, passing back the value of the expression as the value of the function.

Functions that return a value to their caller are called **fruitful functions**, because they **can be treated as an operand in expressions**. Functions can return values of any type. Once you have defined a fruitful function, it can be used **as if it were a value**. For instance, the *f2c* program **calls** `convert()` like this:

```
double celsius = convert(temperature);
```

In this case **temperature** is the **argument** that is used to initialize the **parameter temp**.

Functions **do not need to return a value**. Such a function is often called a **procedure**. Procedures must have some kind of **side-effect**, such as printing, to be useful.

To define a procedure, use **void** as the function's return type. Procedures ordinarily finish by reaching the end of the statements in the body, but you may leave the procedure early by executing a **return** statement by itself.

Two C++ Function Pitfalls

- Unlike Java and C#, **unreachable** code is not illegal. (It is a bug, though!)
- If you forget to add a **return** statement to a fruitful function, your code will still compile. The actual returned value will be random. This may cause your program to crash, or simply act erratically.



This course content is offered under a [CC Attribution Non-Commercial](#) license. Content in this course can be considered under this license unless otherwise noted.