Defining a Function

Learn the syntax for defining a function.

A function must be defined in the following way:

```
returnType functionName (arg1Type arg1Name, arg2Type arg2Name) {
   function_statement;
   function_statement;
   return returnVar;
}
```

This seems rather abstract, but we will see a concrete example in a moment. On the first line, we have to begin by declaring what data **type** the function will return once it finishes. You can define a function that doesn't return anything by using `void`. The next item is the name of the function, which you get to choose. Next, in round brackets, is a list of input arguments that the function expects to see when it is called. Each argument is declared by listing first the **type** of the argument, followed by its name. Then in the body of the function is code, which does whatever you want the function to do. Finally, we have to **return** a value (if the return type is non-void).

Here is a more concrete example. Let's say we want to write a function that writes a message to the screen, "Hello, Paul":

```
void writeMessage(void) {
   printf("Hello, Paul\n");
}
```

Here the return type is `void` since the function doesn't return a value, and the input argument list is also `void`, since it doesn't expect any inputs. We can call the function in a program like this:

```
#include <stdio.h>

void writeMessage(void) {
   printf("Hello, Paul\n");
}

int main() {
    writeMessage();
        return 0;
}
```

We’ll learn more about functions by providing some input arguments to work on.