Library Mechanics

Functions are named "chunks" of code that calculate a value or that carry out an action. I think of them like the "Magic 8-ball"; you ask a question, and get an answer, never knowing how it is accomplished.

In C++, a function associates a computation—specified by a block of code that forms the body of the function—with a particular name. If a function calculates a value, what we call a **fruitful** function, then it may be used in an expression; if it carries out an action (called a void function in C++ or **procedure** in other languages), it cannot.

Using functions reduces bugs and make maintenance more effective by allowing you to reuse proven code, instead of duplicating it. Placing related functions **into a library**, allows you to reuse them in many different contexts.

Organization

Programs using functions can be organized in several different ways. The question is, "Where do definitions and declarations go?"

1. You may **define** your functions before calling them. If you have only one or two functions in a "throw-away" program, this is fine. Because your functions need to appear in a particular order, though, your code is often harder to understand and maintain. In general, you **will not to do this**.
2. You can **prototype** (or declare) your functions at the top of your file (usually under the library `#include` statements) and then **define** the functions later in the file, usually after the `main` function. This makes it easier to read and understand your program, because the primary logic appears before the function details.

   It also makes your code much easier to maintain, because you can then define your functions in any order you like. **Do this for small programs** and for functions that are unique to a particular program.

3. If you have functions which you want to **reuse** in different programs, you should **place those function in a library**, (a collection of similar functions) and **place the prototype into a header file**. That is what you'll generally do in this class for all of your functions from now on.

In the next Lesson, you will learn to use **separate compilation** for your programs.
