The "Standard C Library" - A programmer's toolbox!

The Standard C Library: A collection of pre-existing functions (blocks of code) for performing common programming tasks.

These functions have been written for you, and are included when you purchase a C compiler. We use these pre-existing functions to avoid re-inventing the wheel. This is known as *software reusability*.

We have already had some exposure to the notion of program modularity! See that? You have already done all of this stuff! In the programs we have reviewed/created thus far, we have written the code for function **main** for each program. Then, we have made calls to other functions in the Standard C Library, namely **printf** and **scanf**! Functions **printf** and **scanf** are simply chunks of code that perform the input and output processing that we need. They are located somewhere on your hard drive (placed there when you installed your compiler), and linked with the code in function **main** during the link phase. These functions in the Standard C Library have already been compiled, and only the "object code" resides on your hard drive.

Because C provides a wealth of wonderful, useful code in its library, the first step to becoming a good C programmer is learning *which* functions are available in the Standard C Library. What good is having a library if you don't know what it contains?!

C categorizes related functions together and places their "function definitions or prototypes" in the files with the .h extension. These files are known as "header files" or "include files" and are discussed in more detail during Weeks 4 and 11. For example, the function definitions for all of the functions which perform some sort of input/output (I/O) operations are grouped together in the **stdio.h** header file (stdio is short for standard I/O). We have "included" this header file in our programs so far because we wanted to use the input and output functions defined in that particular header file, namely **scanf** and **printf**. There are other input/output functions in C, as you will see in the summary below.

Appendix B introduces the Standard C Library. Please be sure to reference it throughout this course as you will perhaps need to use some of the functions provided by the library.

A Summary of Appendix B - The Standard C Library.

Note that header files may contain "constants" instead of, or in addition to functions. Constants provide vital information to our programs. We discuss constants in greater detail in Week 7.

Header file (group)	Type of information included in this group.	Pages in Text
stddef.h	Standard constant definitions	471
limits.h	Constants defining the limits for integer data types.	472
float.h	Constants defining the limits for floating point data types.	473
string.h	Functions for processing string data.	475
	Functions for more efficient processing of string data in memory	
ctype.h	Functions for processing character data.	476
stdio.h	Functions for processing input/output data.	477
	Functions for processing in-memory format conversions	
stdlib.h	Functions for converting string data to numeric data.	483
	Functions for processing dynamic memory allocation.	
	General Utility Functions	
math.h	Functions for processing complex math functions.	485

I am not requiring you to study or know the Standard C Library, but just to know of its existence.

Well. That's it for this week! Not too bad huh? I realize that this topic was a bit "dry", but it was one of those things that I needed to discuss at this point. It's like fresh spinach. It may not taste so good, but it's good for you!

Next week we will discuss the very important, fundamental concept of: Program Looping. (I'm starting to feel a bit "loopy" myself. Speaking of dry, I think I'll go make myself a martini! Did I say martini?!?!? I meant to say a nice dry ginger ale! Yah that's it!)

And, as a reminder, your **1st programming assignment** is **due by the end of this week**.