Why learn C programming?

C’s been around for over thirty years. It’s an old language, and there are many other options. Here’s a bit about why you may still find learning C important:

C remains quite popular for much application development. For some graphs related to this, please see:

http://www.tiobe.com/content/paperinfo/tpci/index.html

As this page indicates, C is the number one or number two most common language since about 1987. But even this doesn’t tell the whole story of why C is important. Of the top ten most popular languages shown on this page, seven are based on or have syntax fundamentally similar to, C. For instance, Java, C++, PHP, JC#t, Perl and ObjectiveC all:

- use braces (“{“and “}”) to group statements,
- have similar control structures (while and if) and
- feature similar operators (++, --, +=, ==, etc.).

So learning C serves as an excellent background for many other popular languages.

Where is C used today?

While C has been eclipsed by Java and PHP in many large-scale web applications, it remains the best available alternative in a number of domains.

A very recent example

Curious about the software on-board the Curiosity Mars Rover, I wrote to Ben Cichy, Senior Software and Systems Engineer at the NASA Jet Propulsion Laboratory

http://ai.jpl.nasa.gov/public/home/cichy/

I asked him in what language the Mars lander and rover software were written. He responded, in part:

> “The Curiosity software is almost entirely written in the C programming language. The software is about 1 million lines of code. There is a small fraction that is written in C++ within the software that controls driving...We employed many additional tools that did automatic static code analysis, and wrote over 2 million lines of unit tests...”

As you'll see as you read on, when the applications are demanding – and the Mars landing is – C is usually the language of choice.

Where system-level software is needed.

All Unix-based operating systems, and this includes Linux and Mac OS X, are written primarily in C, with some C++. C works well for operating systems because it is efficient of machine resources, executes quickly, and is reasonably portable among differing hardware platforms.

Where performance is king.

C is still widely used where performance, that is some measure of throughput, must be optimized. This might be megabytes transferred on a LAN to a storage medium, or frames per second, number of concurrent sessions, or any measure of speed and performance one can imagine. Because it is the "low-level high-level" language, C compilers can generate some of the most efficient machine code possible short of actually writing in assembly language. This contributes to a well-earned reputation for good performance.

Where resources are scarce.

Some of the very things that make C a more difficult language for beginners than say Visual Basic are the very features that allow C to run in environments with limited memory and limited processor speed:

- Pointers, with expressions like (*p) = I ;,
- low-level, non-automated control over memory allocation (malloc and free), and
- reflection of the underlying hardware in the operators and syntax of the language (e.g. ++ for increment)

all contribute to C programs being compact enough and fast enough to function on machines with only a few kilobytes of memory, or clock rates in the 10's of megahertz.

Where direct communication with computer hardware is required.

Programs such as:

- device drivers that allow your programs to operate disks, USB devices, video displays, and the like,
- the whole range of embedded applications, those computer systems embedded in cars, DVD players, cell phones, toys and more, and
- various Physical Computing devices, that help software interact with the real world,

all benefit from C’s low-level features and efficiency.

- So when you’re learning C, you’re learning a language that is:
- Still immensely popular,
- A stepping stone to many other popular languages, and

Useful for entertaining projects that connect inexpensive hardware to the real world.

This last reason is one you may choose to explore in our C class, as we apply C to Physical Computing problems.