

Re: new

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2003-10/1957.html

From: David B. Held (dheld_at_codelogicconsulting.com)

Date: 10/12/03

Date: Sun, 12 Oct 2003 02:35:51 -0500

"Willie Clement" <wclement7@msn.com> wrote in message
news:2544924f.0310111857.2e64d78d@posting.google.com...

> *i really want to get into programming. I am only 15 but what*
> *a time to start eh? i would really like to know how a lot of*
> *you started progrrmaing and what some things i would*
> *need to learn. I know about reading books but what is*
> *something that you actually use to program*
> *in something such as c? we'll thanx a lot,!*

You are so lucky. When I was a youngin, the "internet" was just a mysterious thing used by some universities, and "online services" were something too expensive for me and my modemless house to afford. I had to scrape by with whatever meager computer books my public library had to offer, and whatever compiler tools came with whatever version of DOS I happened to be using (which meant BASICA, GW-BASIC, or debug). I was happy as a clam to get A86, which was a shareware assembler.

My first "serious" development tool was Turbo C++. I still think it's great. But you have an unbelievable array of options available to you. If anything, you have the opposite problem I had. You have so many ways to go about learning, that your problem is choosing the path. My advice to you is: get yourself on a *nix box. If you can, get your own computer, and install Linux on it. You probably are already comfortable with Windows, so getting familiar with a fundamentally different OS will give you a broader range of experience that will serve you well later on. Also, it makes it much easier to use gcc. And if you should have any compiler, it should be gcc. If you like, get all the front ends for it, and learn a little about every language (I think that includes Fortran, Ada, Lisp, C, and C++, at least...probably several more).

The best way to learn how to program is to pick a small, doable program that you think is interesting, and try to write it. One of my favorite programs for learning a new

comp.lang.c++: Re: new

language is Tetris. It's small, it works the terminal I/O capabilities of the language, and it's fun to test once you're done. If the language or the terminal can't handle Tetris, I move on. I wrote Tetris for a TI-85, so you shouldn't have any shortage of platforms on which to try this project. By picking something you think would be fun to work on, you will be motivated to learn and not give up. If you finish a project, pick another one that tries out a different part of the language. But most importantly, pick projects that you think are fun. There will be plenty of opportunity to do other people's work when you get older.

You're young, have fun! That's why I got into programming, and if you manage to keep the fun in coding, then maybe someday you'll get paid to do your hobby. Life doesn't get much better than that.

If you go with C++, there are endless tutorials from which to learn. Some are better than others. That's ok. You can't help but learn a few wrong things now and then. Just keep reading, and try to stay away from Herb Schildt books. For good books, check out www.accu.org.

Dave

Outgoing mail is certified Virus Free.

Checked by AVG anti-virus system (<http://www.grisoft.com>).

Version: 6.0.521 / Virus Database: 319 - Release Date: 9/23/2003