C/C++ Programming Resources

Written by Jay Koutavas
Wednesday, 07 November 2007 12:00

Recently I was asked for some recommendations on books for C/C++ development. Here we are.

First there's The C++ Programming Language by the guy who started it all, Bjarne Stroustrup.

I highly recommend you become fluent in the uses of STL -- C++'s Standard Template Library. The book I consider as the "STL Bible" is The C++ Standard Library: A Tutorial and Reference . A must-have book for your library, it's a well-organized, thorough, and practical guide to the STL.

To gain further insight on C++ templates, there's C++ Templates: The Complete Guide by Vandervoorde and Josuttis.

One corner of the STL that doesn't get covered in most STL books to any great lengths is IOStreams. They're a powerful class of STL objects. I stumbled upon an entire book dedicated to them. It's titled Standard C++ IOStreams and Locales: Advanced Programmer's Guide and Reference   by Langer and Kreft.

Once having become fluent with the STL, the next step on your journey is to master it. Here's where Scott Meyer's excellent series of books come in. Get Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library
. Here you'll learn the "gotchyas" on STL use. Very eye-opening in numerous cases.

You should also aquire this Scott Meyers book: Effective C++: 55 Specific Ways to Improve Your Programs and Designs

C/C++ Programming Resources

Written by Jay Koutavas
Wednesday, 07 November 2007 12:00

Good coding standards are not to be underestimated. Good structured code goes hand in hand with well designed software. Here's the book I recommend on code standards by two other legends in the C++ community, Herb Sutter and Andrei Alexandrescu. [C++ Coding Standards: 101 Rules, Guidelines, and Best Practices]
.

Okay, you gotta learn [boost] . And here's the missing user's manual I recommend: [Beyond the C++ Standard Library: An Introduction to Boost]
by Börn Karlsson.

Another "must have" reference for the software developer, which isn't actually C++ specific, is [Design Patterns Elements of Reusable Object-Oriented Software]
by Gamma, Helm, Johnson and Vlissides.

**Update: It's 2018 now and my has C++ grown and improved since I compiled this list. There's C+11, C++14, and C++17. Here's a wonderful modern resource: [C++ Core Guidelines]
maintained by Stroustrup and Sutter.**