

The Seven Habits of Effective C++

Jerry Fitzpatrick

I'm on my fourth cup of coffee this morning and feeling a bit plagiaristic, so please indulge my rip-off of Stephan Covey's popular (and terrific) book *The Seven Habits of Highly-Effective People*. Here are my top picks for successful C++ habits:

1. **Don't expect miracles:** Object-oriented programming is valuable and C++ is a good way to make a transition from C. These tools can make development easier, but not effortless. Be realistic in your expectations of them.
2. **Get plenty of training:** Don't be content to learn OO/C++ by reading the book that came with your compiler. Learn to use OO and C++ effectively by attending expert seminars and workshops. Supplement these sessions by also reading various OO/C++ books and magazines.
3. **Plan prior to coding:** Contrary to what you might think, the oft-recommended iterative design process does not recommend that you dive recklessly into coding. Software development is about problem solving, and good solutions require up-front planning.
4. **Design for maintainability:** Many developers focus on program size or speed as if 8080 processors and 64K of RAM were still mainstream. Size and speed are still important factors, but the primary costs of software are in maintenance. C++ provides many ways to enhance the structure, modularity, and reusability of your code, but it's up to you to use them.
5. **Don't feel obliged to use every language feature:** The C++ language is overflowing with features (e.g. templates, namespaces, RTTI, etc). Although each of these has its place, you can write very effective programs without using each and every one. Such showmanship is best left for the obfuscated C++ contest.
6. **Apply design heuristics and metrics:** Don't throw away the entire software rulebook for OO designs. Modules should still be small, cohesive, and have minimal coupling. You can improve your designs and project planning by using simple metrics and design rules-of-thumb.
7. **Help stamp out memory leaks:** C++ programs rely heavily on dynamic memory allocation, and "garbage collection" is not a built-in feature. For this reason, C++ programs have a nasty habit of leaking memory like crazy. Fortunately there are several design techniques and software utilities that are invaluable for preventing and detecting leaks. Use them!