

Preface

Introduction

I've written this book as a starting point for programming using the C++ language. You may reread key sections now and then to refresh your memory or even use it as a handy reference, but the primary goal is to bust open your brain and stuff the seeds of C++ into your consciousness.

It may take you a week to read this, if you finish it, or a year. Mastery of the C++ language does not come from any single source, be it book or tutor. It comes from years of experience, experimentation, and mingling with your programming brothers and sisters. Know ye that I thought to forsake this timeless advice and bound ahead with great strides. But lo! After starting to learn and use C++ in the fall of 1996 I am still discovering things. Do not be discouraged; expect neither too little nor too much, but be pleasantly surprised by both.

References in this genre of programming are plentiful and generally complete, but the city of building block books is sporadically filled with useful, useless, and flawed explanations. Authors have sown their work not only with knowledge, but with their self-same biases, prejudices, hyped enthusiasm, or lack thereof, and coding habits for good or ill. The blood and brains of scholars is fired and filled by the quirks of people they have never met in person.

Newbies, I have seen or talked to, wander lost among the maze of boundless and circular information. Prizes of cheese escape them beyond barriers unbroken, unexplained. I write this book to bring YOU to the cheese and send you home happy!

Why Choose C++?

Because that's what this book is about; deal with it!

Target Audience

This book is for those who wish to learn the C++ language, but I'd be an Oz Wizard if I said that this book was for everyone with this goal. No book can hope to target that broad of an audience. It is written as plainly as possible and helps you up from no programming or computing experience. If you are struggling with, intimidated by, or wary of this subject then I suggest you give this book a go. I do not intend to make this book a full C++ reference (though that is what I intend for NeilStuff.com) nor do I expect experts to read this without cringing at the simplistic, ye almost *human*, explanations of their closely guarded secrets.

I do not provide inline notes for differences between C++ and other languages. If you are coming from a different programming language, you should recognize some things and be unfamiliar with others. But regardless you will see the differences yourselves.

Organization

This book is written with the beginner in mind. As such I've tried to organize this book so each new chapter is a slightly more advanced topic than the last. The first chapters will teach the most basic and fundamental concepts and as the book progresses the features of the language will shine through.

I've also tried to write this without comparison to other languages. That kind of thing, in my opinion, is best saved for a separate chapter or something. What you want to learn is C++, so that's what I'm going to teach you. I won't spend time comparing it to language 'X' because I can't assume that you know that language or that you even care about it.

Part 1 is an introduction to the world of computers. It requires no computer to read through or do the exercises; it is all concepts. The part welcomes to the readers to the technobabble world of buzzwords, heaps of useless and redundant technology, and the diamonds in the rough waiting to be found out. Feel free to skip this if you have previous programming or computer operation experience.

Part 2 is a guide through the basics of the C++ language. This part will school you enough to get by with your programming.

Part 3 dives into advanced concepts for C++, but tries to maintain the thorough explanations. It also brings the reader up to date with newer additions like namespaces and the standard namespace; shining the light on "proper" C++ program structure.

Part 4 delves further into object-oriented and object-based programming with classes, polymorphism, abstraction, namespaces, static and dynamic libraries.

Part 5 is on the C++ Standard Template Library which is a collection of standard classes for common programming tasks.

I'd also like to include stub entries of things to set people on the path to GUI, multimedia, and low-level programming.

Opinions

My opinions, when I choose to disclose them, are clearly marked throughout the book. I've tried to keep them separate from the text. Every one has their own opinions and I don't want my personal ones interfering with your learning. You can choose to skip them

if you wish. Concepts that I find silly or disgusting may or may not be marked. Please do not assume that everything I teach you in this book I particularly like. This book is for learning the language, not my personal preferences.

Rules to Reading

1. Be Patient

C++ is a massive concept. Many people will never wrap their mind around the entire language and for many more it will take years. Do not let this discourage you! A house can be built of brick, board, or even bones ... but it does not have to use all of them! Such is the case with this language. Only those who have previous programming experience, are collecting what they missed when first learning the language, or simply brushing up may fly through this book.

2. Accept Text

If you are looking to be whizzed immediately off your eye balls and into a world drowned in high performance multimedia then please come down off your cloud. My way of teaching C++ is from the beginning, so you'll start with numbers and text in a plain monochrome format. I will reach graphics in this book, but if you try reading that first you will be sorely unprepared.

Notes

- The quality of an author's work is not based on their knowledge alone, but their ability to communicate effectively to the reader. That said I try not to assume that because I think a concept is simple it deserves less of an explanation. My attempt has been to tackle every concept as though it boggles my fragile little mind.
- This book is written for any decent, modern C++ compiler. It was a difficult choice to either favor the recent C++ standard (completed November 1997) or a broader range of compilers that didn't necessarily follow the entire standard for various reasons. Truly all C++ compilers will eventually support the standard, but where does that leave free, unattended compilers like TC-Lite which were built before then? This book appears to strike a bargain between compiler compatibility and standard trueness. However, later on in the book I will cover making programs more standardized, which isn't a difficult change over. Also, all standard-abiding compilers also support the old ways simply because of the amount of code that has been written. In short, the programs in this book will work with almost any C++ compiler *and* follow the standard as much as possible.
- My approach to teaching the C++ language is skewed from most authors and verily, even teachers. I've always felt that beginners are thrown head long into

the strangest mess of technical bramble with the way things usually flow. If you like the way most C++ books are laid out and it makes sense to you, this book may be a bit of a warped introduction. However, if you find that you can't seem to learn C++ regardless of the book you pick up, this one may be uniquely tasteful. At first I tried reading other C++ books to get an idea of how to write my own. But now that I am actually writing it, I'm finding I only use other books as references for specific concepts rather than structure.

- I left 'easy' out of the title because it all depends on how well my writing reaches you. A lot of publications (free or otherwise) make the claim that their content is "easy" to digest; I've never seen any content that was easy for everyone. Thus, your mileage may vary.
- The content of chapters will build upon the *knowledge* of key points in previous chapters. However, I've tried to organize this so you can enter in at any chapter without having to read the previous ones. That said, there will be no examples that must be incrementally built through a standard progression of the chapters. I always hated those programs that just kept getting bigger and bigger each new chapter and here I am trying to *learn* the language. I will save "real world" examples for specific times and they may be utilized or skimmed over.
- All explanations in this book apply to the C++ language unless otherwise explicitly stated. They may, however, be applicable to a more general arena, but this is not guaranteed.