

A Beginner's Guide
That Makes You Feel **SMART**

C++ WITHOUT FEAR

THIRD EDITION



- **Learn** programming basics fast
- **Understand** with well-illustrated figures and examples
- **Practice** with games, exercises, and puzzles
- **Write** your first C++ program
- **Refer** to summaries, appendices, and C++14 notes

BRIAN OVERLAND

Updated
for
C++14!

Praise for Previous Editions of *C++ Without Fear*

“Dear Mr. Overland, I am in love with your *C++ Without Fear* book. It keeps me awake for hours during the night. This is a really, really good job, especially for beginners. I am a physics undergraduate student, and I needed to learn C since I am in the CERN CMS group—so I needed to do some data analyzing for some of the datas produced at CMS. I must say that thanks to you, I got most of the overall idea in just a few hours. I’m sure there are many, many others out there who are having fun with this as much as I do, while learning. Thank you very much with all my heart. Best wishes, Laura.”

—Lara Vural, particle physics student, Istanbul, Turkey

“It’s hard to tell where I began and ended with your book. I felt like I literally woke up and knew how to write C-style code. The book was written in a tone and pace, with its interludes and philosophical excursions, that was exactly what I needed. Even now when I see it on my bookshelf between Fante’s *Ask the Dust* and *Tao Te Ching*, it just seems like my talisman, my passport, out of where I was to a new place. Again, I can’t overstate the confidence that your book gave me. I knew instinctively that if I busted my tail over the next 18 months I’d be in a better place.

“And that’s exactly what happened. After demonstrating my skill as a programmer and teammate I was offered a full-time job. I’ve never looked back. Again, Brian, thanks for the knowledge, enthusiasm, inventiveness, and humanity you injected into your book.”

—Danny Grady, senior programming analyst at a Fortune 500 company

“*C++ Without Fear* breaks down writing software into logical steps, while always maintaining an easy-to-understand connection with the real world, as the reader goes from thinking about their solution, to pseudo code, to writing in C++. Other books should take note.”

—Kevin Oke, awarding-winning co-founder of LlamaZOO Interactive,
Victoria, BC, Canada

C++ Without Fear: A Beginner's Guide That Makes You Feel Smart

Table of Contents

Cover

Title Page

Copyright Page

Contents

Preface

Well Have Fun, Fun, Fun

Why C and C++?

C++: How to Think Objects

Purpose of the Third Edition

Where Do I Begin?

Icons and More Icons

Anything Not Covered?

A Final Note: Have Fun!

Acknowledgments

About the Author

Chapter 1 Start Using C++

- Install Microsoft Visual Studio

- Create a Project with Microsoft

- Writing a Program in Microsoft Visual Studio

- Running a Program in Visual Studio

Table of Contents

Compatibility Issue #1: stdafx.h

Compatibility Issue #2: Pausing the Screen

If You're Not Using Microsoft

Example 1.1. Print a Message

How It Works

Exercises

Interlude What about the #include and using?

Advancing to the Next Print Line

Example 1.2. Print Multiple Lines

How It Works

Exercises

Interlude What Is a String?

Storing Data: C++ Variables

Introduction to Data Types

Interlude Why Double Precision, Not Single?

Example 1.3. Convert Temperatures

How It Works

Optimizing the Program

Exercises

A Word about Variable Names and Keywords

Exercise

Chapter 1 Summary

Chapter 2 Decisions, Decisions

But First, a Few Words about Data Types

Decision Making in Programs

Interlude What about Artificial Intelligence (AI)?

if and if-else

Interlude Why Two Operators (= and ==)?

Table of Contents

Example 2.1. Odd or Even?

How It Works

Optimizing the Code

Exercise

Introducing Loops

Interlude Infinite Loopiness

Example 2.2. Print 1 to N

How It Works

Optimizing the Program

Exercises

True and False in C++

Interlude The bool Data Type

The Increment Operator (++)

Statements versus Expressions

Introducing Boolean (Short-Circuit) Logic

Interlude What Is true?

Example 2.3. Testing a Persons Age

How It Works

Exercise

Introducing the Math Library

Example 2.4. Prime-Number Test

How It Works

Optimizing the Program

Exercise

Example 2.5. The Subtraction Game (NIM)

How It Works

Exercises

Chapter 2 Summary

Chapter 3 And Even More Decisions!

Table of Contents

The do-while Loop

Example 3.1. Adding Machine

How It Works

Exercises

Introducing Random Numbers

Example 3.2. Guess-the-Number Game

How It Works

Optimizing the Code

Exercises

The switch-case Statement

Example 3.3. Print a Number

How It Works

Exercises

Chapter 3 Summary

Chapter 4 The Handy, All-Purpose for Statement

Loops Used for Counting

Introducing the for Loop

A Wealth of Examples

Interlude Does for Always Behave Like while?

Example 4.1. Printing 1 to N with for

How It Works

Exercises

Declaring Loop Variables On the Fly

Example 4.2. Prime-Number Test with for

How It Works

Exercise

Comparative Languages 101: The Basic For Statement

Chapter 4 Summary

Table of Contents

Chapter 5 Functions: Many Are Called

The Concept of Function

The Basics of Using Functions

Step 1: Declare (Prototype) the Function

Step 2: Define the Function

Step 3: Call the Function

Example 5.1. The avg() Function

How It Works

Function, Call a Function!

Exercises

Example 5.2. Prime-Number Function

How It Works

Exercises

Local and Global Variables

Interlude Why Global Variables at All?

Recursive Functions

Example 5.3. Prime Factorization

How It Works

Interlude Interlude for Math Junkies

Exercises

Example 5.4. Euclids Algorithm for GCF

How It Works

Interlude Who Was Euclid?

Exercises

Interlude Interlude for Math Junkies: Rest of the Proof

Example 5.5. Beautiful Recursion: Tower of Hanoi

How It Works

Exercises

Example 5.6. Random-Number Generator

Table of Contents

How It Works

Exercises

Games and More Games

Chapter 5 Summary

Chapter 6 Arrays: All in a Row

A First Look at C++ Arrays

Initializing Arrays

Zero-Based Indexing

Interlude Why Use Zero-Based Indexes?

Example 6.1. Print Out Elements

How It Works

Exercises

Example 6.2. How Random Is Random?

How It Works

Exercises

Strings and Arrays of Strings

Example 6.3. Print a Number (from Arrays)

How It Works

Exercises

Example 6.4. Simple Card Dealer

How It Works

Exercises

2-D Arrays: Into the Matrix

Chapter 6 Summary

Chapter 7 Pointers: Data by Location

What the Heck Is a Pointer, Anyway?

The Concept of Pointer

Interlude What Do Addresses Look Like?

Table of Contents

Declaring and Using Pointers

Example 7.1. Print Out Addresses

Example 7.2. The double_it Function

How It Works

Exercises

Data Flow in Functions

Swap: Another Function Using Pointers

Example 7.3. Array Sorter

How It Works

Exercises

Reference Arguments (&)

Pointer Arithmetic

Pointers and Array Processing

Example 7.4. Zero Out an Array

How It Works

Writing More Compact Code

Exercises

Chapter 7 Summary

Chapter 8 Strings: Analyzing the Text

Text Storage on the Computer

Interlude How Does the Computer Translate Programs?

It Don't Mean a Thing if It Ain't Got that String

String-Manipulation Functions

Example 8.1. Building Strings

How It Works

Exercises

Interlude What about Escape Sequences?

Reading String Input

Table of Contents

Example 8.2. Get a Number

How It Works

Exercise

Example 8.3. Convert to Uppercase

How It Works

Exercises

Individual Characters versus Strings

Example 8.4. Breaking Up Input with strtok

How It Works

Exercises

The C++ String Class

Include String-Class Support

Declare and Initialize Variables of Class string

Working with Variables of Class string

Input and Output

Example 8.5. Building Strings with the string Class

How It Works

Exercises

Example 8.6. Adding Machine #2

How It Works

Exercises

Other Operations on the string Type

Chapter 8 Summary

Chapter 9 Files: Electronic Storage

Introducing FileStream Objects

How to Refer to Disk Files

Example 9.1. Write Text to a File

How It Works

Exercises

Table of Contents

Example 9.2. Display a Text File

How It Works

Exercises

Text Files versus Binary Files

Interlude Are Binary Files Really More Binary?

Introducing Binary Operations

Example 9.3. Random-Access Write

How It Works

Exercises

Example 9.4. Random-Access Read

How It Works

Exercises

Chapter 9 Summary

Chapter 10 Classes and Objects

OOP, My Code Is Showing

Whats an Object, Anyway?

Interlude OOPIs It Worth It?

Point: A Simple Class

Interlude Interlude for C Programmers: Structures and Classes

Private: Members Only (Protecting the Data)

Example 10.1. Testing the Point Class

How It Works

Exercises

Introducing the Fraction Class

Inline Functions

Find the Greatest Common Factor

Find the Lowest Common Denominator

Example 10.2. Fraction Support Functions

Table of Contents

How It Works

Exercises

Example 10.3. Testing the Fraction Class

How It Works

Interlude A New Kind of #include?

Exercises

Example 10.4. Fraction Arithmetic: add and mult

How It Works

Exercises

Chapter 10 Summary

Chapter 11 Constructors: If You Build It

Introducing Constructors

Multiple Constructors (Overloading)

C++11/C++14 Only: Initializing Members

The Default Constructor and a Warning

Interlude Is C++ Out to Trick You with the Default Constructor?

C++11/C++14 Only: Delegating Constructors

Example 11.1. Point Class Constructors

How It Works

Exercises

Example 11.2. Fraction Class Constructors

How It Works

Exercises

Reference Variables and Arguments (&)

The Copy Constructor

Interlude The Copy Constructor and References

A Constructor from String to Fract

Chapter 11 Summary

Table of Contents

Chapter 12 Two Complete OOP Examples

Dynamic Object Creation

Other Uses of new and delete

Blowin in the Wind: A Binary Tree App

The Bnode Class

The Btree Class

Example 12.1. Names in Alpha Order

How It Works

Exercises

Interlude Recursion versus Iteration Compared

Tower of Hanoi, Animated

After Mystack Class Design

Using the Cstack Class

Example 12.2. Animated Tower

How It Works

Exercises

Chapter 12 Summary

Chapter 13 Easy Programming with STL

Introducing the List Template

Interlude Writing Templates in C++

Creating and Using a List Class

Creating and Using Iterators

C++11/C++14 Only: For Each

Interlude Pointers versus Iterators

Example 13.1. STL Ordered List

How It Works

A Continually Sorted List

Exercises

Table of Contents

Designing an RPN Calculator

Interlude A Brief History of Polish Notation

Using a Stack for RPN

Introducing the Generalized STL Stack Class

Example 13.2. Reverse Polish Calculator

How It Works

Exercises

Correct Interpretation of Angle Brackets

Chapter 13 Summary

Chapter 14 Object-Oriented Monty Hall

Whats the Deal?

TV Programming: Good Deal, Bad Deal

Example 14.1. The PrizeManager Class

How It Works

Optimizing the Code

Exercises

Example 14.2. The DoorManager Class

How It Works

Exercises

Example 14.3. The Full Monty Program

How It Works

Exercises

The Monty Hall Paradox, or Whats Behind the Door?

Improving the Prize Manager

Chapter 14 Summary

Chapter 15 Object-Oriented Poker

Winning in Vegas

How to Draw Cards

Table of Contents

The Card Class

The Deck Class

Doing the Job with Algorithms

Example 15.1. Primitive Video Poker

How It Works

Exercises

The Vector Template

Getting Nums from the Player

Example 15.2. Draw Poker

How It Works

Exercises

How to Evaluate Poker Hands

Example 15.3. Draw-Poker Payout!

How It Works

Exercises

Chapter 15 Summary

Chapter 16 Polymorphic Poker

Multiple Decks

Switching Decks at Runtime

Polymorphism Is the Answer

Example 16.1. A Virtual Dealer

How It Works

Exercises

Interlude What Is the Virtual Penalty?

Pure Virtual and Other Abstract Matters

Abstract Classes and Interfaces

Object Orientation and I/O

cout Is Endlessly Extensible

Table of Contents

But cout Is Not Polymorphic

Example 16.2. True Polymorphism: The IPrintable Class

How It Works

Exercises

A Final Word (or Two)

An (Even More) Final Word

Chapter 16 Summary

Chapter 17 New Features of C++14

The Newest C++14 Features

Digit-Group Separators

String-Literal Suffix

Binary Literals

Example 17.1. Bitwise Operations

Exercises

Features Introduced in C++11

The long long Type

Interlude Why a Natural Integer?

Working with 64-Bit Literals (Constants)

Accepting long long Input

Formatting long long Numbers

Example 17.2. Fibonacci: A 64-Bit Example

How It Works

Exercises

Localizing Numbers

Interlude Who Was Fibonacci?

Range-Based for (For Each)

Example 17.3. Setting an Array with Range-Based for

How It Works

Exercises

Table of Contents

The auto and decltype Keywords

The nullptr Keyword

Strongly Typed Enumerations

enum Classes in C++11 Onward

Extended enum Syntax: Controlling Storage

Raw-String Literals

Chapter 17 Summary

Chapter 18 Operator Functions: Doing It with Class

Introducing Operator Functions

Operator Functions as Global Functions

Improve Efficiency with References

Example 18.1. Point Class Operators

How It Works

Exercises

Example 18.2. Fraction Class Operators

How It Works

Optimizing the Code

Exercises

Working with Other Types

The Class Assignment Function (=)

The Test-for-Equality Function (==)

A Class Print Function

Example 18.3. The Completed Fraction Class

How It Works

Exercises

A Really Final Word (about Ops)

Chapter 18 Summary

Appendix A: Operators

Table of Contents

The Scope (::) Operator

The sizeof Operator

Old- and New-Style Type Casts

Integer versus Floating-Point Division

Bitwise Operators (&, |, ^, ~, <<, and >>)

Conditional Operator

Assignment Operators

Join (,) Operator

Appendix B: Data Types

Precision of Data Types

Data Types of Numeric Literals

String Literals and Escape Sequences

Two's-Complement Format for Signed Integers

Appendix C: Syntax Summary

Basic Expression Syntax

Basic Statement Syntax

Control Structures and Branch Statements

 The if-else Statement

 The while Statement

 The do-while Statement

 The for Statement

 The switch-case Statement

 The break Statement

 The continue Statement

 The goto Statement

 The return Statement

 The throw Statement

 Variable Declarations

Table of Contents

Function Declarations

Class Declarations

Enum Declarations

Appendix D: Preprocessor Directives

The #define Directive

The ## Operator (Concatenation)

The defined Function

The #elif Directive

The #endif Directive

The #error Directive

The #if Directive

The #ifdef Directive

The #ifndef Directive

The #include Directive

The #line Directive

The #undef Directive

Predefined Constants

Appendix E: ASCII Codes

Appendix F: Standard Library Functions

String (C-String) Functions

Data-Conversion Functions

Single-Character Functions

Math Functions

Randomization Functions

Time Functions

Formats for the strftime Function

Table of Contents

Appendix G: I/O Stream Objects and Classes

- Console Stream Objects

- I/O Stream Manipulators

- Input Stream Functions

- Output Stream Functions

- File I/O Functions

Appendix H: STL Classes and Objects

- The STL String Class

- The <bitset> Template

- The <list> Template

- The <vector> Template

- The <stack> Template

Appendix I: Glossary of Terms

Index

