

Thinking in C++, Volume 1, 2nd Edition

Completed January 13, 2000

Bruce Eckel, President,
MindView, Inc.



Planet PDF brings you the Portable Document Format (PDF) version of Thinking in C++ Volume 1 (2nd Edition). Planet PDF is the premier PDF-related site on the web. There is news, software, white papers, interviews, product reviews, Web links, code samples, a forum, and regular articles by many of the most prominent and respected PDF experts in the world. Visit our sites for more detail:

<http://www.planetpdf.com/>

<http://www.codecuts.com/>

<http://www.pdfforum.com/>

<http://www.pdfstore.com/>

[Click here to buy the paper version](#)

**Winner, Software Development Magazine's
1996 Jolt Award for Best Book of the Year**

“This book is a tremendous achievement. You owe it to yourself to have a copy on your shelf. The chapter on iostreams is the most comprehensive and understandable treatment of that subject I've seen to date.”

Al Stevens
Contributing Editor, Doctor Dobbs Journal

“Eckel's book is the only one to so clearly explain how to rethink program construction for object orientation. That the book is also an excellent tutorial on the ins and outs of C++ is an added bonus.”

Andrew Binstock
Editor, Unix Review

“Bruce continues to amaze me with his insight into C++, and *Thinking in C++* is his best collection of ideas yet. If you want clear answers to difficult questions about C++, buy this outstanding book.”

Gary Entsminger
Author, The Tao of Objects

“*Thinking in C++* patiently and methodically explores the issues of when and how to use inlines, references, operator overloading, inheritance and dynamic objects, as well as advanced topics such as the proper use of templates, exceptions and multiple inheritance. The entire effort is woven in a fabric that includes Eckel's own philosophy of object and program design. A must for every C++ developer's bookshelf, *Thinking in C++* is the one C++ book you must have if you're doing serious development with C++.”

Richard Hale Shaw
Contributing Editor, PC Magazine

Comments from Readers:

Wonderful book ... Great stuff! Andrew Schulman, Doctor Dobbs Journal

An absolute, unqualified must. One of the most-used, most trusted books on my shelf." TUG Lines

This is stuff a programmer can really use. IEEE Computer

A refreshing departure. PJ Plauger, Embedded Systems Programming magazine

...Eckel succeeds ... it's so readable. Unix World

Should definitely be your first buy. C Gazette

A fantastic reference for C++! Michael Brandt, Senior Analyst/Programmer, Sydney, Australia

On our project at HRB Systems we call your book "The Answer Book". It is our C++ Bible for the project. Curt Snyder, HRB Systems

Your book is really great, and I can't thank you enough for making it available for free on the web. It's one of the most thorough and useful references for C++ I've seen. Russell Davis

... the only book out there that even comes close to being actually readable when trying to learn the ropes of C++ (and the basics of good object oriented programming in general). Gunther Schulz, KawaiiSoft

I love the examples in your book. There's stuff there that I never would have thought of (and some things that I didn't know you could do)! Rich Herrick, Senior Associate Software Engineer, Lockheed-Martin Federal Systems, Owego, NY

It's an amazing book. Any questions I have I refer to this online book. Helped in every case. I'm simply happy to have access to a book of this caliber. Wes Kells, Comp Eng. Student, SLC Kingston.

You are an invaluable resource and I greatly appreciate your books, email list etc... It seems every project I have worked on has been successful because of your insights. Justin Voshell

This is the book I have been looking for on C++. Thomas A. Fink, Managing Director, Trepp, LLC

Your books are authoritative yet easy to read. To my colleagues I call you the K&R of C++. Mark Orlassino, Senior Design Engineer, Harmon Industries, Inc., Hauppauge, NY

When I first started learning C++, your book "Thinking in C++" was my shining guide light in a dark tunnel. It has been my endeavor to improve my C++ skills whenever possible, and to that effect, "Thinking in C++" has given me the strong foundation for my continuous improvement. Peter Tran, Senior Systems Analyst (IM), Compaq Computer Corporation

This book is the best general reference in my on-going quest to master C++. Most books explain some topics thoroughly but are deficient in others. "Thinking in C++" 2/E does not pass the buck to another book. When I have questions it has answers. Thomas Michel

I have a whole mountain of books and none of them make sense nor do they explain things properly. I have been dying for a good template and STL book. Then I decided to read your material and I was amazed. What you did was show how to write C++ with templates and STL without bogging down with details. What you did was what I expected of the C++ community, the next generation of C++ authors. As an author I AM IMPRESSED at your writing and explanation skills. You covered topics that nobody has properly covered before. Your approach is one from a person who has actually sat down and went through the material in detail. And then you questioned the sanity of the situation and what would be the problem areas. On my bookshelf, it will definitely be one of the necessary books, right beside Petzold. Christian Gross, consultant/mentor cgross@eusoft.com

I think your book is very, very, VERY good. I have compared it to others in the bookstore, and have found that your book actually teaches me basic C++ fundamentals while I learn the STL... a very nice experience to learn about both at once, hand-in-hand. I think your book is laid out very well, and explains things in an easy-to-understand fashion. Jeff Meininger, Software Developer, boxybutgood.com

Your book is the best by far of any I've seen. Please get it right so that we can all have an excellent and "reliable" reference work! And please hurry! We are desperate for a work of this quality! Steve Strickland, Live Minds (a Puzzle business)

(On Usenet) Unlike most other C++ authors, Eckel has made a career of teaching C++ and Java classes ONLY. He's had the benefit of a GREAT deal of novice feedback, and the books reflect that. His books are not just about writing in C++/Java, but understanding the intent of the languages and the mindset that goes with thinking in them. Eckel's also the best technical writer I've read since Jeff Duntemann. Very clear and easy to read. Don't be put off by the apparent large size of his books. Either can be read in *less* than 21 days. :-} Randy Crawford, MRJ Technology Solutions, Fairfax VA

Your work is greatly appreciated and I thank you for helping me understand both C++ and Java better. Barry Wallin, Math/Computer Science Teacher, Rosemount High School, Rosemount, MN

I would like to thank you for your book "Thinking in C++" which is, with no doubt, the best book I ever read about this subject. Riccardo Tarli - SW Engineer - R&D TXT Ingegneria Informatica - Italy

I have been reading both of your books, Thinking In Java and Thinking In C++. Each of these books is easily the best in its category. Ratnakarprasad H. Tiwari, Mumbai, India

... the "Debugging Hints" section is so valuable, I'm tempted to print it and keep it with me at all times. I think this section should be a mandatory part of any introductory class after the first one or two programming problems. Fred Ballard, Synectics Inc.

Your book is really a treasure trove of C++ knowledge. I feel like you give a good overview and then explain the nuts and bolts. Raymond Pickles, Antenna Section, Radar Division, U.S. Naval Research Laboratory, Washington DC

As an Internal Medicine Specialist and Computer Scientist I spend a great deal of time trying to extract information from books and journals. My experience is that a good author is one who makes difficult concepts accessible, a great one makes it look almost easy. On this score you are certainly one of my top three technical writers. Keep up the good work. Dr. Declan O'Kane, Leicester, England

For my second-level C++ course, "Thinking in C++" is my constant reference and companion, and I urge my students to consult it regularly. I refer to the chapter on Operator Overloading constantly. The examples/code alone are worth the cost of the book many times over. So many books and development environments are predicated on the assumption that the only application for a programming language is for a Windows environment; it's great to find and use a book which concentrates on C++ so we can prepare our students for careers in fields like embedded systems, networking, etc., which require real depth of understanding. Robert Chase, Professor, Sweet Briar College

I think it's a fantastic intro to C++, especially for longtime dabblers like me - I often know "how," but rarely "why," and TIC2 is a godsend. Tony Likhite, System Administrator/DBA, Together Networks

After reading the first 80 pages of this book, I have a better understanding of oop than I've gotten out of the ton of books I've accumulated on the subject. Thanks... Rick Schneewind



Thinking

In

C++

Second Edition

Bruce Eckel
President, MindView Inc.



Prentice Hall
Upper Saddle River, New Jersey 07458
<http://www.prenhall.com>



Publisher: Alan Apt
Production Editor: Scott Disanno
Executive Managing Editor: Vince O'Brien
Vice President and Editorial Director: Marcia Horton
Vice President of Production and Manufacturing: David W. Riccardi
Project Manager: Ana Terry
Book Design, Cover Design and Cover Line Art:
Daniel Will-Harris, daniel@will-harris.com
Cover Watercolor: Bruce Eckel
Copy Editor: Stephanie English
Production Coordinator: Lori Bulwin
Editorial Assistant: Toni Holm
Marketing Managers: Jennie Burger, Bryan Gambrel

©2000 by Bruce Eckel, MindView, Inc.



Published by Prentice Hall Inc.
Pearson Higher Education
Upper Saddle River, New Jersey 07632

The information in this book is distributed on an "as is" basis, without warranty. While every precaution has been taken in the preparation of this book, neither the author nor the publisher shall have any liability to any person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by instructions contained in this book or by the computer software or hardware products described herein.

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means including information storage and retrieval systems without permission in writing from the publisher or author, except by a reviewer who may quote brief passages in a review. Any of the names used in the examples and text of this book are fictional; any relationship to persons living or dead or to fictional characters in other works is purely coincidental.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-979809-9

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada, Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Pearson Education Asia Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*



Public C++ Seminars

Check www.BruceEckel.com
for in-depth details and the date
and location of the next:

Hands-On C++ Seminar

- Based on this book
- Get a solid grounding in Standard C++ fundamentals
- Includes in-class programming exercises
- Personal attention during exercises

Intermediate C++ Seminar

- Based on Volume 2 of this book (downloadable at www.BruceEckel.com)
- In-depth coverage of the Standard C++ Library
- Strings, containers, iterators, algorithms
- In-depth templates & exception handling

Advanced C++ Topics

- Based on advanced topics in Volume 2 of this book
- Design patterns
- Building robust systems
- Creating testing & debugging frameworks

**Subscribe to the free newsletter
to be automatically informed
of upcoming seminars**

Also visit www.BruceEckel.com for:

- Consulting Services
- Exercise solutions for this book





Seminars-on-CD-ROM

If you like the *Thinking in C* Seminar-on-CD packaged with this book, then you'll also like:

Bruce Eckel's Hands-On C++ Seminar Multimedia CD ROM

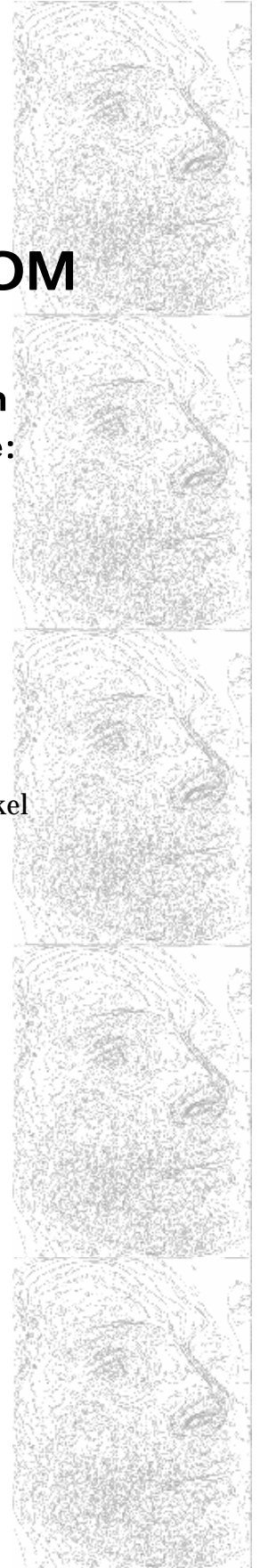
It's like coming to the seminar!

Available at **www.BruceEckel.com**

- Overhead slides and synchronized audio recorded by Bruce Eckel
- All the lectures from the Hands-On C++ Seminar
- Based on this book
- Get a solid grounding in Standard C++ Fundamentals
- Just play it to see and hear the lectures!
- Lectures are indexed so you can rapidly locate the discussion of any subject
- Details and sample lecture can be found on the Web site

See www.BruceEckel.com for other Seminars-on-CD ROM

- The Intermediate C++ Seminar
- Advanced C++ Topics



Dedication

To my parents, my sister, and my brother

What's inside...

Preface	1	1: Introduction to Objects	23
What's new in the second edition ...	2	The progress of abstraction	25
What's in Volume 2 of this book.....	3	An object has an interface.....	27
How to get Volume 2.....	3	The hidden implementation	30
Prerequisites	4	Reusing the implementation	32
Learning C++	4	Inheritance: reusing the interface	34
Goals	6	Is-a vs. is-like-a relationships	38
Chapters.....	7	Interchangeable objects with polymorphism	40
Exercises	13	Creating and destroying objects ...	45
Exercise solutions	13	Exception handling: dealing with errors...	46
Source code	13	Analysis and design.....	48
Language standards	15	Phase 0: Make a plan.....	51
Language support	16	Phase 1: What are we making?	52
The book's CD ROM	16	Phase 2: How will we build it?.....	56
CD ROMs, seminars, and consulting.....	17	Phase 3: Build the core	61
Errors.....	17	Phase 4: Iterate the use cases.....	62
About the cover	18		
Book design and production	19		
Acknowledgements .	20		

Phase 5: Evolution	63
Plans pay off	65
Extreme	
programming	66
Write tests first.....	66
Pair programming	68
Why C++	
succeeds	70
A better C	71
You're already on	
the learning curve.....	71
Efficiency	71
Systems are easier to	
express and understand.....	72
Maximal leverage	
with libraries	73
Source-code reuse	
with templates.....	73
Error handling	73
Programming in the large.....	74
Strategies for	
transition.....	74
Guidelines.....	75
Management obstacles	77
Summary	79

2: Making & Using Objects 83

The process of	
language	
translation	84
Interpreters	85
Compilers	86
The compilation process.....	87
Tools for separate	
compilation.....	89
Declarations vs. definitions..	90
Linking	96
Using libraries	97
Your first	
C++ program.....	99
Using the iostreams class.....	99
Namespaces.....	100
Fundamentals of	
program structure.....	102
"Hello, world!"	103
Running the compiler	105

More about	
iostreams	105
Character array	
concatenation	106
Reading input.....	107
Calling other programs.....	107
Introducing	
strings	108
Reading and	
writing files.....	110
Introducing	
vector	112
Summary	118
Exercises.....	119

3: The C in C++ 121

Creating	
functions	122
Function return values.....	125
Using the C	
function library.....	126
Creating your own	
libraries with the librarian..	127
Controlling	
execution	128
True and false	128
if-else	128
while.....	130
do-while	131
for	131
The break and	
continue keywords.....	132
switch	134
Using and misusing goto....	136
Recursion	137
Introduction to	
operators	138
Precedence	138
Auto increment	
and decrement	139
Introduction to	
data types	140
Basic built-in types	140
bool, true, & false.....	142
Specifiers	143
Introduction to pointers.....	145

Modifying the outside object.....	149
Introduction to C++ references	151
Pointers and references as modifiers	153
Scoping	155
Defining variables on the fly	156
Specifying storage allocation.....	159
Global variables	159
Local variables.....	161
static	161
extern	163
Constants	165
volatile	167
Operators and their use.....	168
Assignment	168
Mathematical operators	169
Relational operators	171
Logical operators	171
Bitwise operators	172
Shift operators	173
Unary operators.....	176
The ternary operator	177
The comma operator	178
Common pitfalls when using operators	179
Casting operators.....	179
C++ explicit casts.....	181
sizeof – an operator by itself	186
The asm keyword.....	187
Explicit operators.....	187
Composite type creation.....	188
Aliasing names with typedef.....	188
Combining variables with struct	189
Clarifying programs with enum.....	192
Saving memory with union.....	195
Arrays	196
Debugging hints	208
Debugging flags.....	208
Turning variables and expressions into strings	211
The C assert() macro.....	212

Function addresses.....	213
Defining a function pointer.....	213
Complicated declarations & definitions.....	214
Using a function pointer	215
Arrays of pointers to functions	216
Make: managing separate compilation.....	217
Make activities	219
Makefiles in this book.....	222
An example makefile.....	223
Summary	226
Exercises.....	226

4: Data Abstraction 233

A tiny C-like library.....	235
Dynamic storage allocation	239
Bad guesses	244
What's wrong?	246
The basic object	247
What's an object?..	255
Abstract data typing	256
Object details.....	257
Header file etiquette	259
Importance of header files.....	260
The multiple-declaration problem	262
The preprocessor directives #define, #ifdef, and #endif.....	263
A standard for header files.	264
Namespaces in headers.....	265
Using headers in projects ..	266
Nested structures ..	266
Global scope resolution	271
Summary	271

Exercises272

5: Hiding the Implementation 277

Setting limits.....278
C++ access
control279
protected281
Friends281
Nested friends284
Is it pure?287
Object layout288
The class289
Modifying Stash to use
access control.....292
Modifying Stack to use
access control.....293
Handle classes.....294
Hiding the
implementation295
Reducing recompilation.....295
Summary298
Exercises299

6: Initialization & Cleanup 301

Guaranteed
initialization with the
constructor303
Guaranteed cleanup
with the
destructor305
Elimination of the
definition block.....308
for loops310
Storage allocation311
Stash with
constructors and
destructors313
Stack with constructors
& destructors316

Aggregate
initialization 320
Default
constructors..... 323
Summary 324
Exercises 325

7: Function Overloading & Default Arguments 327

More name
decoration 329
Overloading on
return values 331
Type-safe linkage 331
Overloading
example 333
unions 336
Default
arguments 340
Placeholder arguments 342
Choosing overloading
vs. default
arguments 342
Summary 348
Exercises 349

8: Constants 353

Value substitution.. 354
const in header files..... 355
Safety consts 356
Aggregates 358
Differences with C 358
Pointers..... 360
Pointer to const..... 361
const pointer..... 362
Assignment and
type checking..... 363
Function arguments &
return values 364
Passing by const value 365

Returning by const value ...	366
Passing and returning addresses	369
Classes	373
const in classes	374
Compile-time constants in classes	377
const objects & member functions	380
volatile	386
Summary	388
Exercises	388

9: Inline Functions 393

Preprocessor	
pitfalls	394
Macros and access	398
Inline functions	399
Inlines inside classes	400
Access functions	401
Stash & Stack	
with inlines	408
Inlines & the compiler	412
Limitations	413
Forward references	414
Hidden activities in constructors & destructors	415
Reducing clutter ...	417
More preprocessor features	418
Token pasting	419
Improved error checking	420
Summary	423
Exercises	424

10: Name Control 427

Static elements	
from C	428
static variables inside functions	428
Controlling linkage	434

Other storage	
class specifiers	436
Namespaces	437
Creating a namespace	437
Using a namespace	440
The use of namespaces	445

Static members in C++

Defining storage for static data members	446
Nested and local classes	451
static member functions	452

Static initialization

dependency	455
What to do	457

Alternate linkage

specifications	465
Summary	466
Exercises	467

11: References & the Copy-Constructor 473

Pointers in C++

References

in C++	475
References in functions	476
Argument-passing guidelines	479

The copy-

constructor	479
Passing & returning by value	480
Copy-construction	487
Default copy-constructor ...	493
Alternatives to copy- construction	496

Pointers

to members	498
Functions	501

Summary

Exercises	505
-----------------	-----

12: Operator Overloading 511

Warning & reassurance	512
Syntax	513
Overloadable operators.....	515
Unary operators.....	515
Binary operators.....	520
Arguments & return values.....	531
Unusual operators.....	535
Operators you can't overload.....	544
Non-member operators.....	545
Basic guidelines.....	547
Overloading assignment.....	548
Behavior of operator=.....	549
Automatic type conversion	561
Constructor conversion.....	561
Operator conversion.....	563
Type conversion example.....	566
Pitfalls in automatic type conversion	567
Summary	570
Exercises	570

13: Dynamic Object Creation 575

Object creation.....	577
C's approach to the heap.....	578
operator new.....	580
operator delete.....	581
A simple example.....	581
Memory manager overhead	582
Early examples redesigned.....	583
delete void* is probably a bug	584
Cleanup responsibility with pointers	585

Stash for pointers.....	586
new & delete for arrays	592
Making a pointer more like an array	593
Running out of storage.....	594
Overloading new & delete.....	595
Overloading global new & delete.....	597
Overloading new & delete for a class	599
Overloading new & delete for arrays	603
Constructor calls.....	605
placement new & delete	607
Summary	609
Exercises.....	610

14: Inheritance & Composition 613

Composition syntax.....	614
Inheritance syntax.....	616
The constructor initializer list	618
Member object initialization	619
Built-in types in the initializer list	620
Combining composition & inheritance	621
Order of constructor & destructor calls.....	623
Name hiding	625
Functions that don't automatically inherit	630
Inheritance and static member functions.....	635

Choosing composition vs. inheritance	635
Subtyping	637
private inheritance	640
protected.....	641
protected inheritance.....	643
Operator overloading & inheritance.....	643
Multiple inheritance.....	645
Incremental development	645
Upcasting	647
Why "upcasting?"	648
Upcasting and the copy-constructor.....	649
Composition vs. inheritance (revisited)	652
Pointer & reference upcasting	653
A crisis	654
Summary	654
Exercises	655

15: Polymorphism & Virtual Functions 659

Evolution of C++ programmers	660
Upcasting	662
The problem	663
Function call binding.....	663
virtual functions	664
Extensibility	666
How C++ implements late binding.....	669
Storing type information	670
Picturing virtual functions...	672
Under the hood.....	674
Installing the vpointer	676
Objects are different	676
Why virtual functions?.....	677

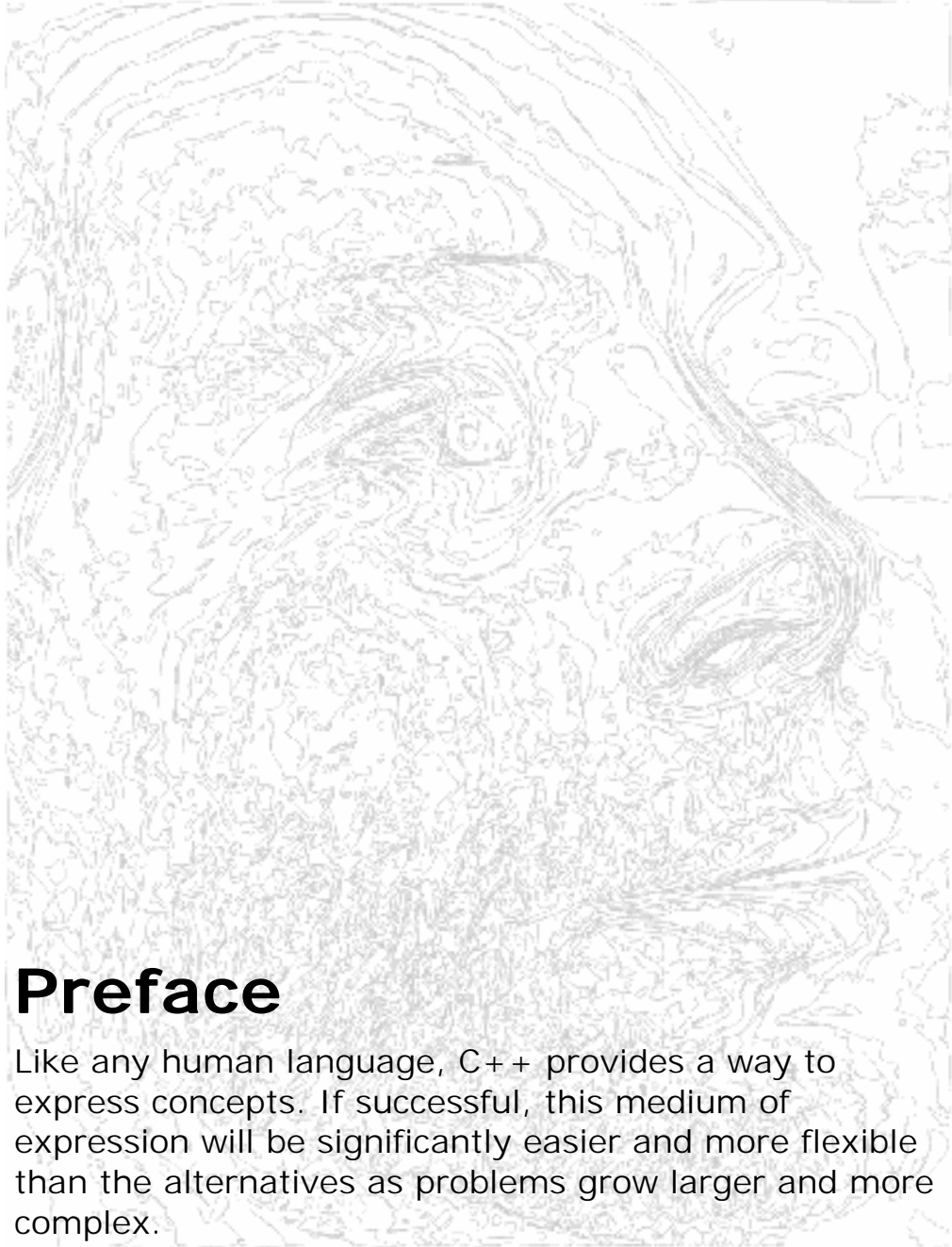
Abstract base classes and pure virtual functions	679
Pure virtual definitions	684
Inheritance and the VTABLE.....	685
Object slicing	688
Overloading & overriding.....	691
Variant return type	693
virtual functions & constructors.....	695
Order of constructor calls ..	696
Behavior of virtual functions inside constructors.....	697
Destructors and virtual destructors .	699
Pure virtual destructors	701
Virtuals in destructors	704
Creating an object-based hierarchy.....	705
Operator overloading.....	709
Downcasting	712
Summary	716
Exercises.....	717

16: Introduction to Templates 723

Containers	724
The need for containers.....	726
Overview of templates	727
The template solution.....	730
Template syntax	732
Non-inline function definitions	734
IntStack as a template	735
Constants in templates.....	738
Stack and Stash as templates.....	739
Templatized pointer Stash .	742
Turning ownership	

on and off	748
Holding objects	
by value	751
Introducing	
iterators	754
Stack with iterators.....	764
PStash with iterators.....	768
Why iterators?.....	774
Function templates.....	778
Summary	779
Exercises	780

A: Coding Style	785
B: Programming Guidelines	797
C: Recommended Reading	815
C	816
General C++	816
My own list of books.....	817
Depth & dark corners	818
Analysis & design...	819
Index	823



Preface

Like any human language, C++ provides a way to express concepts. If successful, this medium of expression will be significantly easier and more flexible than the alternatives as problems grow larger and more complex.

You can't just look at C++ as a collection of features; some of the features make no sense in isolation. You can only use the sum of the parts if you are thinking about *design*, not simply coding. And to understand C++ this way, you must understand the problems with C and with programming in general. This book discusses programming problems, why they are problems, and the approach C++ has taken to solve such problems. Thus, the set of features I explain in each chapter will be based on the way that I see a particular type of problem being solved with the language. In this way I hope to move you, a little at a time, from understanding C to the point where the C++ mindset becomes your native tongue.

Throughout, I'll be taking the attitude that you want to build a model in your head that allows you to understand the language all the way down to the bare metal; if you encounter a puzzle, you'll be able to feed it to your model and deduce the answer. I will try to convey to you the insights that have rearranged my brain to make me start "thinking in C++."

What's new in the second edition

This book is a thorough rewrite of the first edition to reflect all of the changes introduced in C++ by the finalization of the C++ Standard, and also to reflect what I've learned since writing the first edition. The entire text present in the first edition has been examined and rewritten, sometimes removing old examples, often changing existing examples and adding new ones, and adding many new exercises. Significant rearrangement and re-ordering of the material took place to reflect the availability of better tools and my improved understanding of how people learn C++. A new chapter was added which is a rapid introduction to the C concepts and basic C++ features for those who don't have the C background to tackle the rest of the book. The CD ROM bound into the back of the book contains a seminar that is an even gentler introduction to the C concepts necessary to understand C++ (or Java). It was created by Chuck Allison for my company (MindView, Inc.), and

it's called "Thinking in C: Foundations for Java and C++." It introduces you to the aspects of C that are necessary for you to move on to C++ or Java, leaving out the nasty bits that C programmers must deal with on a day-to-day basis but that the C++ and Java languages steer you away from (or even eliminate, in the case of Java).

So the short answer to the question "what's different in the 2nd edition?" is: what isn't brand new has been rewritten, sometimes to the point where you wouldn't recognize the original examples and material.

What's in Volume 2 of this book

The completion of the C++ Standard also added a number of important new libraries, such as **string** and the containers and algorithms in the Standard C++ Library, as well as new complexity in templates. These and other more advanced topics have been relegated to Volume 2 of this book, including issues such as multiple inheritance, exception handling, design patterns, and topics about building and debugging stable systems.

How to get Volume 2

Just like the book you currently hold, *Thinking in C++, Volume 2* is downloadable in its entirety from my Web site at www.BruceEckel.com. You can find information on the Web site about the expected print date of Volume 2.

The Web site also contains the source code for both of the books, along with updates and information about other seminars-on-CD ROM that MindView, Inc. offers, public seminars, and in-house training, consulting, mentoring, and walkthroughs.

- [The Science of Ocean Waves here](#)
- [download online *Painting Nature in Watercolor with Cathy Johnson: 37 Step-by-Step Demonstrations Using Watercolor Pencil and Paint*](#)
- [The History of Mathematics: An Introduction \(6th Edition\) pdf](#)
- [click **The Cat Sitter's Pajamas \(A Dixie Hemingway Mystery, Book 7\) pdf, azw \(kindle\), epub**](#)
- [Noise Uprising: The Audiopolitics of a World Musical Revolution book](#)
- [click Touched \(Sense Thieves, Book 1\) pdf, azw \(kindle\), epub](#)

- <http://korplast.gr/lib/Planning-and-Environmental-Protection--A-Review-of-Law-and-Policy.pdf>
- <http://wind-in-herleshausen.de/?freebooks/Painting-Nature-in-Watercolor-with-Cathy-Johnson--37-Step-by-Step-Demonstrations-Using-Watercolor-Pencil-and>
- <http://wind-in-herleshausen.de/?freebooks/The-History-of-Mathematics--An-Introduction--6th-Edition-.pdf>
- <http://aseasonedman.com/ebooks/The-Cat-Sitter-s-Pajamas--A-Dixie-Hemingway-Mystery--Book-7-.pdf>
- <http://conexdx.com/library/Noise-Uprising--The-Audiopolitics-of-a-World-Musical-Revolution.pdf>
- <http://patrickvincitore.com/?ebooks/Eye-of-the-Forest--Children-of-the-Lamp--Book-5-.pdf>