

Preface

This book is designed to be a textbook and reference for programming in the C++ language. Although it does include programming techniques, it is organized around the features of the C++ language, rather than any particular curriculum of techniques. The main audience I had in mind is undergraduate students who had not had extensive programming experience with the C++ language. As such, this book is a suitable C++ text or reference for a wide range of users. The introductory chapters are written at a level that is accessible to beginners, while the boxed sections of those chapters serve to introduce more experienced programmers to basic C++ syntax. Later chapters are also understandable to beginners, but are written at a level suitable for students who have progressed to these more advanced topics. *Absolute C++* is also suitable for anyone learning the C++ language on their own. (For those who want a textbook with more pedagogical material and more on very basic programming technique, try my text *Problem Solving with C++*, Eighth Edition, Pearson Education.)

The C++ coverage in this book goes well beyond what a beginner needs to know. In particular, it has extensive coverage of inheritance, polymorphism, exception handling, and the Standard Template Library (STL), as well as basic coverage of patterns and the unified modeling language (UML).

CHANGES IN THIS EDITION

This fifth edition presents the same programming philosophy as the fourth edition. For instructors, you can teach the same course, presenting the same topics in the same order with no changes in the material covered or the chapters assigned. Changes include:

- Chapter 1 now includes a short introduction to the string class and Chapter 2 includes a short introduction to reading data from a text file. Together, this material allows instructors to present students with problems of larger scale and real-world applicability as early as Chapter 2.
- Chapter 12 now includes a section on using `stringstream` to convert between strings and other data types.
- Chapter 13 briefly describes tail recursion and gives an example of mutual recursion.
- Ten new self-test exercises have been added along with twenty-five new Programming Projects. By request, some of these are longer and less prescriptive projects that give the student more practice designing programming solutions.
- Several errors that were found in the fourth edition have been corrected.
- Fifteen new VideoNotes have been added to the book's website (www.pearsonhighered.com/savitch) for a total of forty-six videos. These VideoNotes walk students through the process of problem solving and coding to reinforce key programming concepts. An icon appears in the margin of the book when a video is available regarding the corresponding topic in the text.

ANSI/ISO C++ STANDARD

This edition is fully compatible with compilers that meet the latest ANSI/ISO C++ standard.

STANDARD TEMPLATE LIBRARY

The Standard Template Library (STL) is an extensive collection of preprogrammed data structure classes and algorithms. The STL is perhaps as big a topic as the core C++ language, so I have included a substantial introduction to STL. There is a full chapter on the general topic of templates and a full chapter on the particulars of STL, as well as other material on, or related to, STL at other points in the text.

OBJECT-ORIENTED PROGRAMMING

This book is organized around the structure of C++. As such, the early chapters cover aspects of C++ that are common to most high-level programming languages but are not particularly oriented toward object-oriented programming (OOP). For a reference book—and for a book for learning a second language—this makes sense. However, I consider C++ to be an OOP language. If you are programming in C++ and not C, you must be using the OOP features of C++. This text offers extensive coverage of encapsulation, inheritance, and polymorphism as realized in the C++ language. Chapter 20, on patterns and UML, gives additional coverage of OOP-related material.

FLEXIBILITY IN TOPIC ORDERING

This book allows instructors wide latitude in reordering the material. This is important if a book is to serve as a reference. This is also in keeping with my philosophy of accommodating the instructor's style, rather than tying the instructor to my own personal preference of topic ordering. Each chapter introduction explains what material must already have been covered before each section of the chapter can be covered.

ACCESSIBLE TO STUDENTS

It is not enough for a book to present the right topics in the right order. It is not even enough for it be correct and clear to an instructor. The material also needs to be presented in a way that is accessible to the novice. Like my other textbooks, which proved to be very popular with students, this book was written to be friendly and accessible to the student.

SUMMARY BOXES

Each major point is summarized in a boxed section. These boxed sections are spread throughout each chapter. They serve as summaries of the material, as a quick reference source, and as a quick way to learn the C++ syntax for a feature you know about in general but for which you do not know the C++ particulars.

SELF-TEST EXERCISES

Each chapter contains numerous self-test exercises. Complete answers for all the self-test exercises are given at the end of each chapter.

VIDEO NOTES



VideoNotes are step-by-step videos that guide readers through the solution to an end of chapter problem or further illuminate a concept presented in the text. Icons in the text indicate where a VideoNote enhances a topic. Fully navigable problems allow for self-paced instruction. VideoNotes are located at www.pearsonhighered.com/savitch.

OTHER FEATURES

Pitfall sections, programming technique sections, and examples of complete programs with sample input and output are given throughout each chapter. Each chapter ends with a summary and a collection of programming projects.

ONLINE PRACTICE AND ASSESSMENT WITH MyProgrammingLab



MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students who often struggle with the basic concepts and paradigms of popular high-level programming languages.

A self-study and homework tool, a MyProgrammingLab course consists of hundreds of small practice problems organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong—and why. For instructors, a comprehensive gradebook tracks correct and incorrect answers and stores the code inputted by students for review.

MyProgrammingLab is offered to users of this book in partnership with Turing's Craft, the makers of the CodeLab interactive programming exercise system. For a full demonstration, to see feedback from instructors and students, or to get started using MyProgrammingLab in your course, visit www.myprogramminglab.com.

SUPPORT MATERIAL

The following support materials are available to all users of this book at www.pearsonhighered.com/cssupport:

- Source code from the book

The following resources are available to qualified instructors only at www.pearsonhighered.com/irc. Please contact your local sales representative for access information.

- Instructor's Manual with Solutions
- PowerPoint® slides

ACKNOWLEDGMENTS

Numerous individuals have contributed invaluable help and support to making this book happen. Frank Ruggirello and Susan Hartman at Addison-Wesley first conceived the idea and supported the first edition, for which I owe them a debt of gratitude. A special thanks to Matt Goldstein who was the editor for the second, third, and fourth editions. His help and support were critical to making this project succeed. Chelsea Kharakozova, Marilyn Lloyd, Yez Alayan, and the other fine people at Pearson Education also provided valuable support and encouragement.

The following reviewers provided suggestions for the book. I thank them all for their hard work and helpful comments.

Richard Albright	University of Delaware
J. Boyd Trolinger	Butte College
Jerry K. Billbrey, Jr	Francis Marion University
Albert M. K. Cheng	University of Houston
David Cherba	Michigan State University
Fredrick H. Colclough	Colorado Technical University
Drue Coles	Boston University
Stephen Corbesero	Moravian College
Christopher E. Cramer	
Ron DiNapoli	Cornell University
Qin Ding	Pennsylvania State University, Harrisburg
Martin Dulberg	North Carolina State University
H. E. Dunsmore	Purdue University
Evan Golub	University of Maryland
Terry Harvey	University of Delaware
Joanna Klukowska	Hunter College, CUNY
Lawrence S. Kroll	San Francisco State University
Stephen P. Leach	Florida State University
Alvin S. Lim	Auburn University
Tim H. Lin	Cal Poly Pomona
R. M. Lowe	Clemson University
Jeffrey L. Popyack	Drexel University
Amar Raheja	Cal Poly Pomona
Victoria Rayskin	University of Central Los Angeles
Loren Rhodes	Juniata College
Jeff Ringenberg	University of Michigan

Victor Shtern	Boston University
Aaron Striegel	University of Notre Dame
J. Boyd Trolinger	Butte College
Chrysafis Vogiatzis	University of Florida
Joel Weinstein	Northeastern University
Dick Whalen	College of Southern Maryland

A special thanks goes to Kenrick Mock (University of Alaska Anchorage) who executed the updating of this edition. He once again had the difficult job of satisfying me, the editor, and himself. I thank him for a truly excellent job.

Walter Savitch