Preface

Today, the ANSI C++ programming language is widely used throughout the world in both academia and industry. In many educational institutions it is the language of choice for a first programming course and for a language to be used for computer science instruction. A key reason for this is that C++ has drifted down the curriculum from more advanced courses to more introductory courses. Further, C++ comes with many useful libraries, and is supported by sophisticated integrated environments. It is a language that efficiently supports object-oriented programming (OOP) the dominant contemporary programming methodology.

C++ by Dissection presents a thorough introduction to the programming process by carefully developing working programs to illuminate key features of the C++ programming language. Program code is explained in an easy-to-follow, careful manner throughout. The code has been tested on several platforms and is found on the bundled CD-rom accompanying this text. The code in *C++ By Dissection* can be used with most C++ systems, including those found in operating systems such as MacOS, MS-DOS, OS/2, UNIX, and Windows.

C++, invented at Bell Labs by Bjarne Stroustrup in the mid-1980s, is a powerful, modern, successor language to C. C++ adds to C the concept of *class*, a mechanism for providing user-defined types, also called *abstract data types*. C++ supports *object-oriented* programming by these means and by providing inheritance and runtime type binding.

Dissections

This book presents readers with a clear and thorough introduction to the programming process by carefully developing working C++ programs, using the method of *dissection*. Dissection is a unique pedagogical tool first developed by the author in 1984 to illuminate key features of working code. A dissection is similar to a structured walk-through of the code. Its intention is to explain to the reader newly encountered programming elements and idioms as found in working code. Programs and functions are explained in an easy-to-follow step-by-step manner. Key ideas are reinforced throughout by use in different contexts.

No Background Assumed

This book assumes no programming background and can be used by students and first time computer users. Experienced programmers not familiar with C++ will also benefit from the carefully structured presentation of the C++ language. For student use, the book is intended as a first course in computer science or programming.

It is suitable for a CS1 course or beginning programming course for other disciplines. Each chapter presents a number of carefully explained programs, which lead the student in a holistic manner to ever-improving programming skills. From the start, the student is introduced to complete programs, and at an early point in the text is introduced to writing functions as a major feature of structured programming. The function is to the program as the paragraph is to the essay. Competence in writing functions is the hallmark of the skilled programmer and hence is emphasized. Examples and exercises are plentiful in content and level of difficulty. They allow instructors to pick assignments appropriate to their audiences.

Special Features

C++ by Dissection: The Essentials of C++ Programming incorporates a number of special features:

- A CD-Rom with a working compiler.
- A website with and the full electronically searchable text of this book. Also included are active links to useful web-sites and complete working code for this text
- Software engineering practice is described throughout
- Dr. P's prescriptions are concise programming tips provided for the beginner for each chapter
- Early explanation of simple recursion to reflect its earlier introduction in beginning computer science courses
- Coverage of program correctness and type-safety
- In-depth explanation of functions and pointers because these concepts are typically stumbling blocks for the beginner
- Object-oriented programming concepts are emphasized
- Generic programming and STL are carefully described
- UML diagrams are introduced as an aid to understanding object-oriented programming
- Comparison to Java, optional Java exercises and coordinating references to *Java by Dissection* (with Charlie McDowell)
- Active links to online code by clicking on the infile line above each major program section.
- Active links to online sites via clicking on blue underlined text.

Chapter Features

Each chapter contains the following pedagogical elements:

Dissections. Major elements of the important example programs are explained by the method of dissection. This step-by-step discussion of new programming ideas helps the reader encountering these ideas for the first time to understand them.

Object-oriented programming. The reader is led gradually to the object-oriented style. Chapter 4, *Classes and Abstract Data Types*, introduces classes, which are the basic mechanism for producing modular programs and implementing abstract data types. Class variables are the objects being manipulated. Chapter 8, *Inheritance and OOP*, develops inheritance and virtual functions, two key elements in this paradigm. Chapter 11, *OOP Using C++*, discusses OOP programming philosophy. This book develops in the programmer an appreciation of this point of view.

Programming Style and Software Engineering. Programming style and software methodology is stressed throughout. Important concepts such as structured branching statements, nested flow of control, top-down design, and object-oriented programming are presented early in the book. A consistent and proper coding style is adopted from the beginning with careful explanation as to its importance and rationale. The coding style used in the book is one commonly used by working programming professionals in the C++ community.

Working Code. Right from the start the student is introduced to full working programs. With the executable code, the student can better understand and appreciate the programming ideas under discussion. Many programs and functions are explained through dissections. Variations on programming ideas are often presented in the exercises.

*Common Programming Errors***.** Many typical programming bugs, along with techniques for avoiding them, are described. Much of the frustration of learning a programming language is caused by encountering obscure errors. Many books discuss correct code but leave the reader to a trial-and-error process for finding out about bugs. This book explains how typical errors in C++ are made and what must be done to correct them.

Dr. P's Prescriptions. A series of programming tips is based on wide experience. A concise rationale is given for each tip.

Comparison to Java. An optional section describes the programming elements of Java that are comparable to the C++ examples. Exercises supporting these sections are included as well. For the most part, C++ and Java have equivalent elements. The text aids the student already conversant in Java to migrate to C++. Also the C++ student who later takes up Java will benefit from this section. Furthermore, as the book is a companion volume to *Java by Dissection* (with Charlie McDowell) the reader has access to complete explanations of the Java concepts fully utilizing this book's pedagogy.

Summary. A succinct list of points covered in the chapter serves as a review for the reader, reinforcing the new ideas that were presented in the chapter.

Exercises. The exercises test the student's knowledge of the language. Many exercises are intended to be done interactively while reading the text. This encourages self-paced instruction by the reader. In addition to exercising features of the language, some exercises look at a topic in more detail, and others extend the reader's knowledge to an advanced area of use.

Classroom Usage

This book can be used as a text in a one-semester course that teaches students how to program. Chapters 1 through 5 cover the C++ programming language through the use of arrays, pointers, and basic object programming. A second-semester course can be devoted to more advanced data types, OOP, generic programming and STL, file processing, and software engineering as covered in Chapters 6 through 11. In a course designed for students who already have some knowledge of programming, not necessarily in C++, the instructor can cover all the topics in the text. This book can also be used as a text in other computer science courses that require the student to use C++. In a comparative language course, it can be used with companion volumes for C, Java, and C# that follow the same dissection approach and share many of the same examples done uniquely in each language.

Interactive Environment

This book is written explicitly for an interactive environment. Experimentation via keyboard and screen is encouraged throughout. For PCs, there are many vendors that supply interactive C++ systems, including Borland, IBM, Metroworks, Microsoft, and Symantec.

Professional Use

While intended for the beginning programmer, *C++ by Dissection: The Essentials of C++ Programming* is a friendly introduction to the entire language for the experienced programmer as well. In conjunction with *A Book on C, Fourth Edition* by Al Kelley and Ira Pohl (Addison Wesley Longman, Inc., Reading, MA, 1998, ISBN 0-201183994), the computer professional will gain a comprehensive understanding of both languages. As a package, the two books offer an integrated treatment of the C/C++ programming language and its use that is unavailable elsewhere. Furthermore, in conjunction with *Java by Dissection* by Ira Pohl and Charlie McDowell (Addison Wesley Longman, Inc., Reading, MA, 1999, ISBN 0-201-61248-8), the student or professional is also given an integrated treatment of the object-oriented language Java.

This book is the basis of many on-site professional training courses given by the author, who has used its contents to train professionals and students in various forums since 1986. The text is the basis for Web-based training in C++ available from www.digitalthink.com.

Supplements

Support materials are available to instructors adopting this textbook for classroom use and include the following:

- Solutions to exercises
- Code for example programs
- Powerpoint slides of all the figures

Please check on-line information for this book at www.aw.com/cssupport for more information on obtaining these supplements.

Acknowledgments

Our special thanks go to Uwe F. Mayer, George Belotsky, and Bruce Montague, who were careful readers of the technical content of this work and suggested numerous improvements, without being responsible for my errors. Thanks to our reviewers, Charles Anderson, Colorado state University; Parris Egbert, Brigham Young University; Chris Eagle, Naval Postgraduate School; Nigel Gwee, Louisiana State University; Stephen P. Leach, Florida State University; and Steven C. Shaffer, Penn State University. Thanks also to John dePillis, Debra Dolsberry and Laura Pohl who developed and drew many of the cartoons. Most importantly further thanks to Debra Dolsberry, who acted as the chief technical editor for much of the material in this book and the CD-Rom. In addition, she was largely responsible for using FrameMaker to create files suitable for typesetting this book. Thanks also to Charlie McDowell and Al Kelley for writing companion volumes in C and Java.

We would also like to thank Maite Suarez-Rivas, Acquisitions Editor, Katherine Harutunian, Project Editor, and Patty Mahtani, Associate Managing Editor for their enthusiasm, support, and encouragement; and we would like to thank Caroline Roop and Sally Boylan at Argosy, for the careful attention to the production of this book.

Ira Pohl
University of California, Santa Cruz