- An AP Study Guide for students preparing to take the AP Computer Science examination. The study guide provides summaries of AP topics and questions similar to those on the AP exam.
- An instructor's manual that provides scheduling help, teaching suggestions, chapter tests, and solutions to questions and problems from the student textbook and workbook.

INSTRUCTIONAL ELEMENTS OF THE TEXT

The text incorporates the following instructional elements.

- **Objectives.** Each chapter begins with a list of learning goals.
- **Overview.** Following the objectives, the Overview puts the goals of the chapter into focus.
- **Sections.** Each chapter is divided into sections, which helps break the material into manageable pieces.
- **Exercises.** Hands-on exercises appear at places where reinforcement is needed. Some exercises require students to enter program code, others only require that students load, compile, and run existing programs.
- **End-of-Section Questions.** Review questions appear at the end of each section.
- **End-of-Section Problems.** Most sections include programming problems that challenge students to apply what they just learned. These problems are designed to be completed in 5–30 minutes.
- **Key Terms.** A list of the important terminology is presented in each chapter.
- **Summary.** Each chapter ends with a quick reference of the concepts covered in the chapter.
- **Projects.** The projects at the end of each chapter provide programming practice to challenge students with all levels of experience. These interdisciplinary projects typically require more time than the problems at the end of sections.
- **Case Studies.** Six case studies appear in strategic locations throughout the book. The case studies allow students to see practical programs that make use of the concepts learned in the chapters just prior to the case study. The source code for each case study is already complete, allowing the student to concentrate on analyzing the existing code. Each case study ends with suggestions of possible extensions or modifications to the program.
- **Sidebars.** Sidebars warn students of potential pitfalls and note special features or information that can enrich students' understanding.

A MESSAGE TO STUDENTS

You may be an experienced programmer, you may be new to programming, or you may be somewhere in between. Whatever the case, this book will help you develop your programming skills and understanding of the fundamentals of computer science.

In this book, you will be programming in C++, which is the language used to develop most operating systems and commercial microcomputer software. You won't be programming the next blockbuster computer program overnight, but you will be taking the first steps toward the kind of knowledge required to produce professional computer software.

Programming computers is not difficult, but programming in a professional language like C++ is challenging. The rewards for facing the challenges are many, however. The ability to program computers is undeniably a valuable skill in today's world. But even if you have no plans to make a career of computer programming, you will realize benefits from the time spent studying computer science. Learning to program computers develops your ability to solve problems of all kinds. You will learn to break a problem into manageable parts and think in logical ways. In addition, programming allows you to be creative.

This book is for you and the students who will study programming after you. I want to know what you think of the book and how it can be improved. I can be reached by electronic mail at tknowlton@aol.com. I promise a prompt and personal reply.

Todd Knowlton

ACKNOWLEDGMENTS

This book is the result of a lot of hard work by many people. I want to thank the team of editors, reviewers, instructors, and students who fulfilled distinct roles in the development of this book.

I thank **Dave Lafferty** and **Janie Schwark** of South-Western for their interest in this book and devotion to its success. I also thank those involved in the development and field testing of the original edition: **Brandon Adkins, Angela Askins, William Bakken, John Baldwin, Greg Buxkemper, Bruce Charbonneau, Rebecca Dailey, Brian Davis, Mark Durrett, Mark Gentry, Eric Hosch, Brian Jurries, Heath Keene, Mark Kim, Jonathan Kleid, Michael Landrum, Scott McGrew, Bill Perry, Tan Pham, Joe Sherwood, Macneil Shonle, Mark Tittle, Aliver Villarreal, John Walter, Michael Wester, Jeremy Wilson,** and **Todd Woolery.**

I thank **Bryan Stephens** for his work on this edition, including much of the workbook and the appendix on numerical algorithms, and **Jonathan Kleid** for his work on the analysis of algorithms appendix. I also thank **Michael Landrum** for his review of the book, and **Macneil Shonle** for his technical expertise in the area of object-oriented programming and his work on the error handling appendix.

I especially thank **Greg Buxkemper** for his hard work preparing programs and manuscript for this edition. Greg was instrumental in integrating object-oriented programming into the book.

I also thank the students and teachers who have provided suggestions and uncovered errors that we were able to address in this edition. Keep that e-mail coming.

Finally, I thank my wife **Melissa** and my daughters **Kaley** and **Amy** for making life fun and for allowing me to spend the hours necessary to produce these books.