Preface

Knowledge of problem solving and programming concepts is necessary for those who develop applications for users. Unfortunately, many students have greater difficulty with problem solving than they do with the syntax of computer languages. The art of programming is learning multiple techniques and applying those techniques to specific problems. When students learn basic programming and problem-solving techniques, they can then concentrate on the syntax when learning specific languages. These techniques may be presented in a separate class on problem solving or with a first language course that concentrates on problem solving. This approach tends to decrease students’ frustration and improves their success rate.

This book is intended for a one-semester introductory course for programming majors. It can serve as a primary text or as a supplement. Although this book is written for students who have little or no computer experience, those who have studied a computer language can benefit from the generic presentation of the material.

The text provides a step-by-step progression of ideas with detailed explanation and many illustrations, from the basics of mathematical functions and operators to the design and use of techniques such as codes, arrays, pointers, other data structures, database concepts, and object-oriented programming concepts. The text uses problem-solving tools, such as problem analysis charts, interactivity charts, IPO charts, algorithms, and flowcharts and Universal Modeling Language (UML), to design a solution to a problem. The appendices present additional tools, including Nassi-Schneiderman charts, and Warnier-Orr diagrams. Putting It All Together sections illustrate a complete solution for a given problem, using the concepts previously presented. In some cases, an earlier solution is updated to incorporate more sophisticated techniques. Throughout the text, problems presented are typical of the business world and provide excellent experience for students. These problems then can be presented in a language course so that students can finish the solution on the computer.

Organization of the Text

- Unit One, Introduction to Problem Solving and Programming, presents basic concepts of problem solving, an introduction to how problems are solved on computers, mathematical concepts required for problem solving using a computer, and steps for analyzing a problem and designing an appropriate solution.
- Unit Two, Logic Structures, presents basic concepts of programming, including local and global variables, parameters, and four basic logic structures. The three basic logic structures are sequential, decision, and loop logic.
- Unit Three, Data Structures, presents the concepts of arrays, sorting techniques, stacks, queues, linked lists, and binary trees.
- Unit Four, Database Management Systems, presents terminology and techniques to implement an application using a database management system.
- Unit Five, Object-Oriented Programming, presents basic concepts in the design of a solution using object-oriented languages. UML is used as the basic problem-solving tool.
- Unit Six, Introduction to Game Development, presents basic concepts of game development through the use of object-oriented programming. An introduction
to assembly language is also presented because many game developers are requiring the knowledge of assembly language to make their games run faster.

- Unit Seven, File Processing, presents techniques of file processing. While these chapters are important to those students studying COBOL, the industry is rapidly converting to database management systems. Therefore, these chapters are not as important as they were 10 or 15 years ago.

- The appendices present the Otto the Robot problem, ASCII and EBCDIC codes, blank forms, other problem-solving tools, and some common functions. Otto software is available to instructors via the Prentice Hall Instructor Resource Center. Visit www.prenhall.com for access.

### New in This Edition

The ninth edition responds to suggestions from reviewers and changes in the programming industry. The following changes were made:

- Added a third column to the algorithm and flowchart figures that includes pseudocode. Several reviewers recommended the addition of pseudocode to the chapters.

- Combined the case structure chapter with the decision chapter. The case logic structure is really a type of decision logic structure, therefore the two chapters have been combined.

- Updated the section on cohesion and coupling.

- Rewrote of the Object Oriented Programming (OOP) chapters. The reviewers requested an update of these chapters.

- Introduced Universal Modeling Language (UML) in Chapter 3 with follow up in the chapters on OOP. UML is being used in industry with OOP, therefore it has been used as a problem-solving tool in the chapters on OOP.

- Added a section on software development life cycle.

- Added more problems at the end of the chapters.

### Resources

Instructor Resources include lecture slides, Instructor’s Manual, and Otto software. Instructor Resources are located at www.prenhall.com/sprankle. Contact your local Prentice Hall Sales Representative for access information.

### Acknowledgments

We are indebted to those who reviewed the manuscript and offered suggestions and constructive comments. In particular, we thank John Arena, Augusta Technical College Augusta (GA); Janet Brown-Sederberg, Massasoit Community College Brockton (MA); Mark Whigham, Lawson St-Birmingham Birmingham (AL); Meg McManus, Northwest Florida State College Niceville (FL); Mark Jones, Lock Haven University Lock Haven (PA); Franklin Fondjo Fotou, Langston University Langston (OK); Humayan Zafar, University of Texas-San Antonio San Antonio (TX); Casey Cegielski, Auburn U Main Campus Auburn (AL); John Mensing, Brookdale Community College Lincroft (NJ); Pamella Johnson, KCTCS-Elizabethtown Elizabethtown (KY); Dennis Roebuck, Delta College University Center MI (NA); Linda Shepherd, University of Texas-San Antonio San Antonio (TX), and Joyce Welby, San Jacinto Coll-South San Jacinto (CA). Thanks to Tracy Dunkelberger and Jane Bonnell at Prentice Hall and to Bruce Hobart at Pine Tree Composition for their support throughout the production process.

Maureen Sprankle
Jim Hubbard