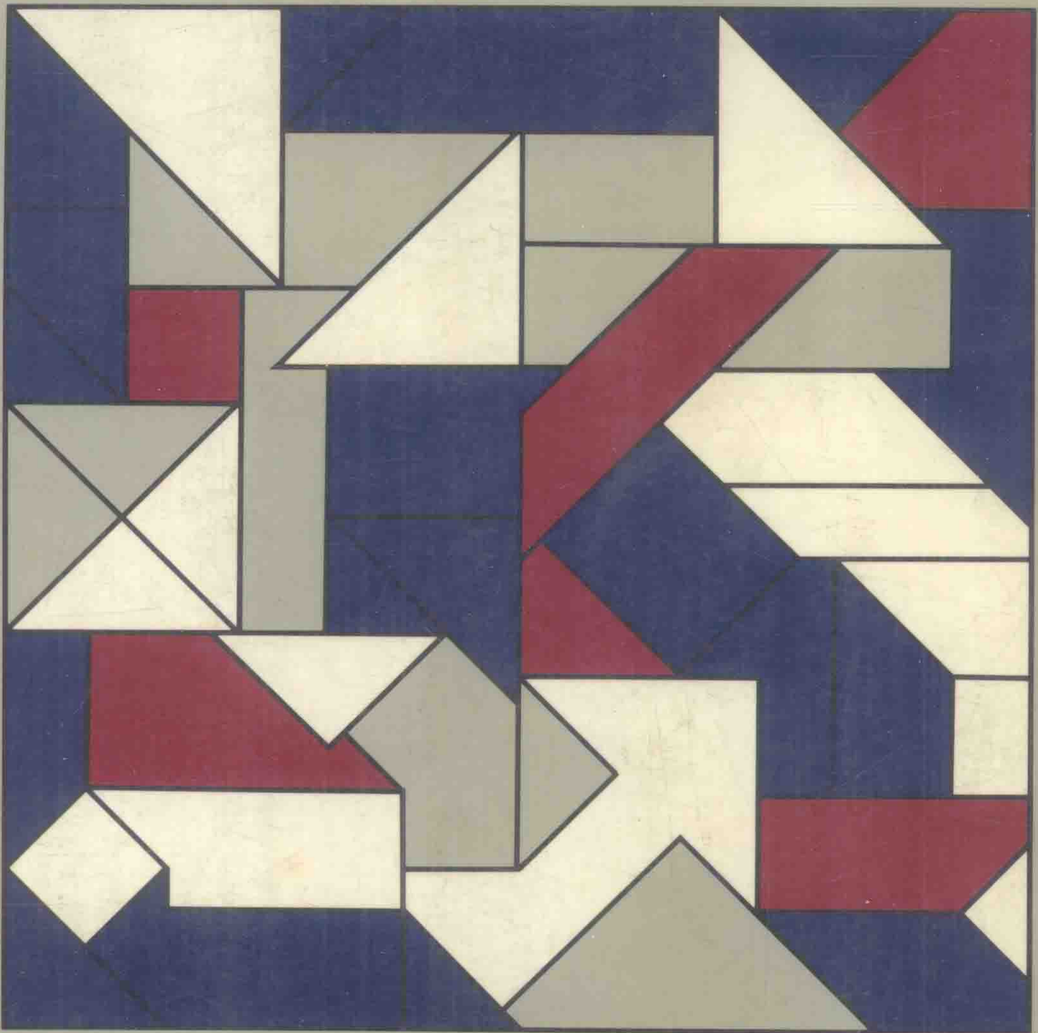
PASCAL STRUCTURE AND STYLE



RICHARD LAMB

PASCAL STRUCTURE AND STYLE

RICHARD LAMB

The Benjamin/Cummings Publishing Company, Inc.
Menlo Park, California • Reading, Massachusetts •
Don Mills, Ontario • Wokingham, U.K. • Amsterdam •
Sydney • Singapore • Tokyo • Mexico City • Bogota •
Santiago • San Juan •

Sponsoring Editor: Alan Apt
Production Supervisor: Larry Olsen
Book and Cover Designer: Rodelinde Albrecht
Artist: Robert Bausch

Designs at the opening of Chapters 1, 2, 3, and 5 by Robert Bausch.
Designs at the opening of Chapters 4, 6 through 19 by Jan Arthur Lee.
Cover drawing by Robert Bausch.

Photo Credits:
Figure 1.1 from Iowa State University.
Figure 1.2 from Intel Corp.
Figure 1.3 from IBM Corp. and Apple Computer, Inc.
Figure 1.6 from Historical Pictures Service, Chicago.

Library of Congress Cataloging-in-Publication Data
Lamb, Richard (Richard C.)
Pascal structure and style.

(Benjamin/Cummings series in structured programming)
Includes index.
1. PASCAL (Computer program language) 2. Structured programming. I. Title. II. Series.
QA76.73.P2L345 1986 005.13'3 85-18511

ISBN 0-8053-5830-7

BCDEFGHIJ-DO-89876

The Benjamin/Cummings Publishing Company, Inc.
2727 Sand Hill Road
Menlo Park, California 94025

PASCAL: STRUCTURE AND STYLE

To my mother and father,
who have given me much,
especially a relationship with my Father,
who has given me everything.

Preface

Teaching computer programming has changed rapidly in the past few years, in large part because of the widespread availability and usefulness of computers in society. Scientists and mathematicians are no longer the only people who are using computers, and computer literacy is quickly becoming the most necessary skill that college graduates need to find a job. Many students who are not computer science majors are taking introductory computer classes in order to familiarize themselves with the field. In fact, many students have already taken computer programming courses in high school.

Unique Approach

This creates a problem. Most text books for the introductory programming course assume that students have had no experience in front of a computer and must learn everything for the first time. Other programming books assume the reader is fluent in one language and attempt to teach the reader in a minimum amount of time how to program in a second language. However, no current text book on the Pascal language assumes a mixed audience. This book assumes that the students using it will include students who have never studied any programming language before as well as students who have taken one or more programming courses in BASIC. This book does not assume that the student has a prior knowledge of BASIC, so the material will be easily understood by the novice. However, the BASIC programmer often starts with a disadvantage because of habits she or he has learned while programming with the limitations of BASIC, especially the unstructured dialects of BASIC. The book provides a Hints for BASIC Programmers section in each chapter to help encourage the good aspects and to help correct the bad aspects of prior programming experience in BASIC.

Emphasis on Examples

The current trend in computer science education focuses on teaching abstract problem-solving skills combined with the syntax and programming techniques of a particular language. The approach of this book follows this trend. Each chapter contains material that extends the student's knowledge of the syntax of Pascal as well as discussion about and complete examples of problem solving using the language elements covered in the discussion. The programming examples are interesting and are taken from common uses of the computer. The problems are discussed thoroughly; each is solved through rigorous application of the top-down approach to algorithmic development through the use of pseudocode. The problem is analyzed and developed in parts and then brought together, so the student is able to see each step in the process from problem statement to solution.

Important Features

Programming with Style: Chapter 5 is entirely devoted to development of a consistent and elegant programming style. This same consistent style is exemplified in the programs and program fragments throughout the text.

Common Errors and Debugging: Each chapter contains a discussion of syntax and execution errors that are commonly encountered after the introduction of the new material. This section also contains defensive programming hints to aid in program debugging.

Hints for BASIC Programmers: In each of the first 15 chapters, a section is devoted entirely to a comparison and contrast of the elements in BASIC that translate into Pascal. Often, the BASIC programmer's experience is a disadvantage, and bad habits need to be explicitly discouraged.

Writing Large Programs: Chapter 17, Writing and Documenting Large Programs, is a unique chapter on the issues involved in the development of large programs.

Programming Examples: Except for Chapter 1, 2, and 5, each chapter contains at least one and often two or more complete programming examples. These special sections give the problem statement, the initial main algorithm, algorithms for subtasks, and the individual procedures, often refined several times. The sections include the final program to be executed and actual input and output files.

Algorithmic Development: Each programming example illustrates fully the concept of top-down algorithmic development through the use of pseudocode. The pseudocode is informal English, allowing students to have a model of algorithmic development without being handicapped by unfamiliar formal structures.

Exercises and Self Tests: Each chapter contains many Self Tests at the ends of sections as well as Exercises at the end of the chapter. The answers to the Self Tests are given at the end of the book. The Exercises contain error-checking or analysis problems as well as a wide range of programming problems.

Instructor's Guide: Answers to Exercises and discussion about them are included in the *Instructor's Guide.*

All the programs and program fragments have been executed on a DEC-20 mainframe and are written in Standard Pascal as defined in the *Pascal User Manual and Report,* Second Edition (Jensen and Wirth, Springer-Verlag, 1975).

Organization

Each chapter is meant to contain enough material for one to three lectures in a normal semester or quarter course. (Usually, one-quarter courses will not finish the book.) The first chapter introduces computers and the need for programming languages. The second chapter introduces Pascal and the building blocks of the language.

The third and fourth chapters introduce variables and procedures, respectively. The first control structure taught is the procedure, in order to emphasize program breakdown and good design. The sixth chapter continues with the emphasis on procedural development as value and variable parameters are introduced. In this way, complete and consistent parameterization of procedures is emphasized and exemplified.

The fifth chapter, Programming with Style, is a detour from the introduction of the syntax of Pascal to the issue of *style.* This chapter collects many programming proverbs and motivates the student to write consistent, elegant code that is well commented and structured. Material in this chapter is referred to repeatedly throughout the text as a reminder.

Chapters 7, 8, 9, and 11 focus on other control structures and their use. The FOR loop, IF statement, REPEAT and WHILE loops, and the CASE statement are introduced, with particular emphasis on which control structure is most appropriate in particular situations.

Chapter 10 does for input and output issues what Chapter 5 does for the issue of style. It collects what has been mentioned up to that point and discusses how to use multiple input and output files.

Chapter 12 begins the discussion of data types and data structures. Chapter 12 introduces the notion of a type. Chapters 13 and 14 introduce arrays and multidimensional arrays. Chapter 15 discusses the concept of a record and illustrates the multipurpose data structure, the array of records. Optional Chapter 16 details the use of structured files and sets.

Chapter 17 provides a unique approach to the issue of writing large programs. It focuses on the trade-off between the complexity of the control structure and the complexity of the data structure needed to solve a large problem, and it outlines steps to make each of the control and data structures appropriate to the given problem. Chapter 17 considers a large problem, writing a database manager, and outlines its data structure and control structures.

Chapters 18 and 19 complete the discussion of the Pascal language with the topics of recursion and pointers. The disadvantages as well as the advantages of recursive control and data structures are presented.

Acknowledgments

Many individuals contributed to the development of this text. First, I am grateful to the members of the Department of Computer Science at Stanford University. Many of the ideas presented here came as a result of learning and teaching there. I would especially like to thank Denny Brown, Arthur Keller, Jock MacKinlay, Rich Pattis, and Stuart Reges for much of what I know about teaching computer programming. Arthur Keller and Stuart Reges were especially instrumental in my decision to write this book and its early development.

I would also like to express my gratitude to those who reviewed the manuscript and who gave very helpful comments at various stages in its development: James Gips, Boston College; Peter Flanagan, Boston University; Michael Main, University of Colorado; Tom Lynch, Northwestern University; Tom Bailey, University of Wyoming; Dave Lamb and Martin Widyono, Stanford University.

I also am indebted to individuals in Stanford Christian Fellowship who have played a part in the development of this text. Dave Lamb, Rob Perkins, and Martin Widyono have been helpful as teaching assistants and fellow instructors in developing examples throughout the text. Thanks also go to the many graders and other helpers with whom I have worked: Nate Bacon, Sue Dante, Ruben DiRado, Michael Dittmar, Kevin Gillette, Bill Glad, Curtis Gruenler, Mark Kennedy, Michael O'Brien, Rae Lee Olson, Katie Stock, and Kathy Wolfe. The text, like teaching itself, has been something of a team effort, and I am grateful to Sue Dante, Tom Nygren, Alissa Riper, Liz Argetsinger, and Mark Hornbostle for help in the typing of the manuscript. Greg Read provided helpful nontechnical advice throughout the development process. Finally, I am grateful for the prayers of the Christian Fellowships at Stanford University and University of California, Santa Cruz, as I began and completed this project.

Richard Lamb

Brief Contents

Contents