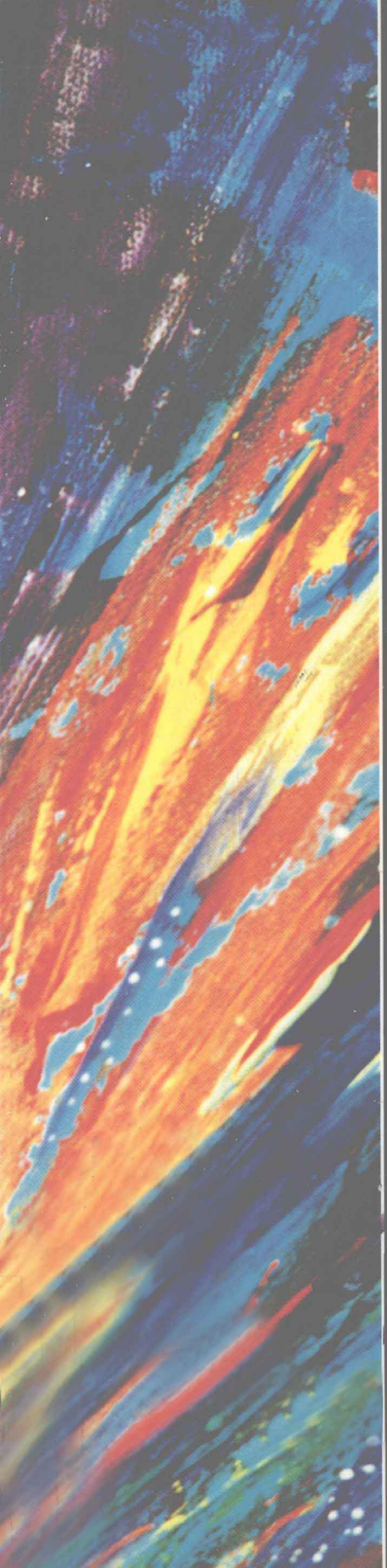

COMPUTER SCIENCE

A Structured Programming Approach Using C

Behrouz A. Forouzan
Richard F. Gilberg

COMPUTER SCIENCE:
A Structured Programming Approach Using C

Behrouz A. Forouzan

De Anza College

Richard F. Gilberg

De Anza College

PWS PUBLISHING COMPANY

I(T)P *An International Thomson Publishing Company*

Boston • Albany • Bonn • Cincinnati • Detroit • London • Madrid
Melbourne • Mexico City • New York • Pacific Grove • Paris
San Francisco • Singapore • Tokyo • Toronto • Washington

PWS Publishing Company
20 Park Plaza, Boston, MA 02116–4324

Reprinted in 1998 by PWS Publishing Company,
a division of International Thomson Publishing, Inc.

I(T)P ®
International Thomson Publishing
The trademark ITP is used under license.

Printed in the United States of America.
98 99 00 01 — 10 9 8 7 6 5 4

ISBN: 0-314-09573-X

This book is printed on recycled, acid-free paper.

Sponsoring Editor: *Mike Sugarman*
Production Editor: *Andrea Goldman*
Manufacturing Coordinator: *Andrew Christensen*
Marketing Manager: *Nathan Wilbur*
Composition: *Carlisle Communications*
Text and Cover Printer: *West Publising Company*

Library of Congress Cataloging-in-Publication-Data

Forouzan, Behrouz A.
Computer science: a structured programming approach using C / Behrouz A. Forouzan, Richard F. Gilberg
p. cm.
Includes index.
ISBN 0–314–09573-X (soft : alk. paper)
1. C (Computer program language) 2. Structured programming. I. Gilberg, Richard II. Title.
QA76.73.C15F66 1997
005.13'3--dc20 97-31288
CIP

For more information, contact:
PWS Publishing Company
20 Park Plaza
Boston, MA 02116

International Thomson Publishing Europe
Berkshire House
168-173 High Holborn
London WC1V 7AA
England

Thomas Nelson Australia
102 Dodds Street
South Melbourne, 3205
Victoria, Australia

Nelson Canada
1120 Birchmont Road
Scarborough, Ontario
Canada M1K 5G4

International Thomson Editores
Campos Eliseos 385, Piso 7
Col. Polanco
11560 Mexico D.F., Mexico

International Thomson Publishing GmbH
Königswinterer Strasse 418
53227 Bonn, Germany

International Thomson Publishing Asia
221 Henderson Road
#05-10 Henderson Building
Singapore 0315

International Thomson Publishing Japan
Hirakawacho Kyowa Building, 31
2-2-1 Hirakawacho
Chiyoda-ku, Tokyo 102
Japan

COMPUTER SCIENCE:
A Structured Programming Approach Using C

TO OUR STUDENTS

who are the rationale for this text.

To my family

BAF

To Evelyn

RFG

PREFACE

This text has two primary objectives. First, to teach the student the basic principles of programming as outlined in the ACM curriculum for a CS1 class. Second, to teach the basic constructs of the C Language. While accomplishing both of these objectives, this text also puts them in the context of good software engineering concepts that we have developed through over thirty years of experience in industry and academia.

A C LANGUAGE PERSPECTIVE

While C is a complex and professional language, our experience of using an early version of this book in the classroom has shown that beginning students can easily understand it. We believe that if the language is put into a perspective that allows the student to understand its design it is not difficult.

There are two aspects of C that separate it from most other languages: expressions and pointers. The concept of expressions, as used in C, is unique. This is one of the reasons that we felt this book had to be built for C from the ground up, not based on a book that used Pascal. The first half of this text, therefore, builds a firm understanding of expressions, introducing pointers only to the extent necessary to cover pass-by-reference and arrays. Our experiences have shown that with a firm grasp of the expression concept, much of the mystery of C disappears.

Beginning slowly with Chapter 9, we develop the concept of pointers ending with a simple introduction to linked lists. While not all courses will have time to cover linked lists, those that do will give the students a head start into data structures. Good students will also find that they can study the linked lists material as a preview to data structures during their term break. The last chapter of the text covers C constructs that are generally beyond the scope of a CS1 class. This material is included to provide the students with references to these subjects that will be useful when they take other courses that require the complete C Language.

FEATURES OF THE BOOK

There are several features of this book that not only make it unique, but make it easier for beginning students to understand.

Structure and Style

One of our basic tenets is that good habits are formed early. The corollary to this tenet is that bad habits are hard to break. Therefore, we consistently emphasize the principles of structured programming and software engineering. Every complete program in the book uses a consistent style. As programs are analyzed, style and standards are further explained. We are not saying that there aren't other good styles, but our experience has shown that if students are exposed to a good style and implement it, they will be better able to adapt to other good styles. On the other hand, unlearning sloppy short-cut habits is very difficult.

Principle Before Practice

Whenever possible, we develop the principle of a subject before we introduce the language implementation. For example, in Chapter 5 we first introduce the concept of logical data and selection and then we introduce the if...else and switch statements. This approach gives the student an understanding of selection before introducing the nuances of the language.

Visual Approach

A brief scanning of the book will demonstrate that our approach is very visual. There are over 400 figures, plus over 70 tables and 180 program examples. While this tends to create a large book, the visual approach makes it much easier for students to follow the material.

Examples

While the programming examples vary in complexity, each of them uses a consistent style. Our experience working with productional programs that live for 10 to 20 years convinced us that readable and understandable programs are easier to work with than programs written in a terse, cryptic manner. For that reason, and to emphasize the structure of the language, we label the sections in a function with comments. We consistently follow a style that places only one declaration, definition, or statement on a line. These programs are available to allow students not only to run them, but to explore related topics through modification and experimentation.

Coding Techniques

Throughout the text we include coding techniques that make programs more readable and often more efficient. For example, on page 248 you will find the following discussion:

> Where do we check for greater than? The answer is that we default the greater than condition to the nested else. When coding a two-way selection statement, try to code the most probable condition first; with nested selection statements, code the most probable first and the least probable last.

These techniques are drawn from our extensive industry experience and are not found in most texts.

Software Engineering

A discussion of software engineering principles concludes each chapter. Our intent is not to replace a separate course in software engineering. Rather, we firmly believe that by incorporating basic software engineering principles early in their studies, students will be better prepared for a formal treatment of the subject. Even more important, by writing well engineered programs from the beginning, they will not be forced to unlearn and relearn. They will also relate more to, and comprehend more of, software discussions in their subsequent classes.

While the Software Engineering Sections are found at the end of the chapter, they are most successfully taught by introducing them as the chapter unfolds. Then, a short review at the end of the chapter summarizes the principles that have been demonstrated during the lectures.

You will note that these sections are visually distinguishable from the rest of the chapter. They have been set apart for several reasons. First, they are in reality a small book in a book. While these sections contain important material, the rest of the book stands on its own without them. You may, therefore, decide to cover the Software Engineering Sections formally as a topics for lectures or informally while the chapter material is being covered. You may decide to leave them to the student as additional reading. Or, you may decide to exclude them entirely from the formal materials for your class.

In all but two chapters the Software Engineering Sections directly pertain to the chapter material. Of the other two chapters, the Software Engineering Section in Chapter 12, Strings, covers an important general topic, Software Quality. Then, in Chapter 15, we develop a case study that builds a structure chart.

Pedagogical End Material

The end material of each chapter contains two parts. The first part includes three sections that are intended to aid the student in reviewing the chapter. The second part, Practice Sets, includes three sections of questions that test the student's knowledge of the chapter material.

Tips and Common Programming Errors points out helpful hints and possible problem areas.

Key Terms provides a list of all of the boldface terms introduced in the chapter.

Summary contains a concise overview of all of the key points for students to understand in the chapter.

Practice Sets

Exercises are short questions covering the material in the chapter. The answers to the odd numbered questions are included in the back of the book.

Problems are short coding problems generally intended to be run on a computer. They can usually be developed in two to three hours.

Projects are longer, major assignments that may take the average student six to nine hours to develop.

The Instructor's Solution Manual contains a complete set of solutions to all exercises and problems.

APPENDIXES AND COVER MATERIAL

The appendixes are intended to provide quick reference material, such as the ANSI Character Set, or a review of material, such as numbering systems, usually covered in a general computer class. An appendix is also included that discusses the standards and guidelines used throughout the book. Inside the covers are two important figures that are used continually throughout the course: the Precedence Table and the Formatted I/O Codes.

ACKNOWLEDGMENTS

No text of this scope can be developed without the support of many people. This is especially true for this text, which was "field tested"for two years by our students at De Anza College. Our first acknowledgment, therefore, has to be to the hundreds of students who by using the text and commenting on it made a vital contribution. We thank four students for their tremendous assistance in correcting and clarifying the material: Madeline Damiano, Sophia Fegan, Shouli Tiechman and Tomo Nagai.

We would also like to acknowledge the support of the De Anza staff. Their encouragement helped us launch the project and their comments contributed to its success. To name them all is impossible, but we especially thank Beverly D'Urso, Anne Oney, and George Rice.

To anyone who has not been through the process, the value of peer reviews cannot be appreciated enough. Writing a text rapidly becomes a myopic process. The important guidance of reviewers who can stand back and review the text as a whole cannot be measured. To twist an old cliche, "They are not valuable, they are priceless." We would especially like to acknowledge the contributions of the following reviewers:

Stephen Allen, *Utah State University*
Ali Behforooz, *Towson State University*
George Berry, *Wentworth Institute of Technology*
Ernest Carey, *Utah Valley State College*
Constance Conner, *City College of San Francisco*
John S. DaPonte, *Southern Connecticut State University*
Maurice L. Eggen, *Trinity University*
Peter Gabrovsky, *CSU Northridge*
Robert Gann, *Hartwick College*
Rick Graziani, *Cabrillo College*
Jerzy Jaromczyk, *University of Kentucky*
John Kinio, *Humber College*
Roberta Klibaner, *College of Staten Island*
Joseph A. Konstan, *University of Minnesota*
Krishna Kulkarni, *Rust College*
John Lowther, *Michigan Technological University*
Mike Michaelson, *Palomar College*
Jo Ann Parikh, *Southern Connecticut State University*
Mark Parker, *Shoreline Community College*
Oskar Reiksts, *Kutztown University*
Jim Roberts, *Carnegie Mellon University*
Ali Salenia, *South Dakota State University*
Larry Sells, *Oklahoma City University*
Shashi Shekhar, *University of Minnesota*
Robert Signorile, *Boston College*
Brenda Sonderegger, *Montana State University*
Deborah Sturm, *College of Staten Island*
Venkat Subramanian, *University of Houston*
John B. Tappen, *University of Southern Colorado*
Marc Thomasm, *California State University, Bakersfield*
John Trono, *St. Michael's College*

Choosing a publisher is a difficult task. We chose West for several reasons, two of the most important ones being our editors, Rick Mixter and Keith Dodson. Our thanks to them for helping us to write the best book we could write. Paul O'Neill ably guided the book through production.

Last, and most obviously not the least, is the support of our families and friends. Many years ago an author described writing a text as a "locking yourself in a room" process. While the authors suffer through the writing process, families and friends suffer through their absence. We can only hope that as they view the final product, they feel that their sacrifices were worth it.

Behrouz A. Forouzan
Richard F. Gilberg
Cupertino, CA

CONTENTS