COMPUTING

A Problem-Solving Approach with FORTRAN 77

T. RAY NANNEY

COMPUTING

A Problem-Solving Approach with FORTRAN 77

T. RAY NANNEY

Professor of Computer Science
Furman University

Prentice-Hall, Inc.
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Nanney, T. Ray
Computing: a problem-solving approach with FORTRAN 77.

Includes index.
1. FORTRAN (Computer program language)
2. Electronic digital computers—Programming.
I. Title.
QA76.73.F25N36 001.64'24 80-28220
ISBN 0-13-165209-5

To

Elizabeth Nanney

who inspired this work and helped
with every phase of its development.

Editorial/production supervision and interior design: Service to Publishers
Cover design: Lee Cohen
Manufacturing buyer: Joyce Levatino and Gordon Osbourne

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PRENTICE-HALL INTERNATIONAL, INC., London
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, Sydney
PRENTICE-HALL OF CANADA, LTD., Toronto
PRENTICE-HALL OF INDIA PRIVATE LIMITED, New Delhi
PRENTICE-HALL OF JAPAN, INC., Tokyo
PRENTICE-HALL OF SOUTHEAST ASIA PTE. LTD., Singapore
WHITEHALL BOOKS LIMITED, Wellington, New Zealand

PREFACE

This book is a text for a beginning course in computer science in a liberal arts environment. The organization of topics in the book is the result of the continual evolution of a course that has been taught at Furman University since 1968. For the majority of students at Furman, this course would be the only one that they would take in computer science. Yet, many of them would later be required to write computer programs both in courses and in independent study in their own disciplines. It was essential, therefore, that the students have the opportunity to become good programmers. I had observed in previous introductory courses that many intelligent, motivated, nonscience students had great difficulty mastering the material in the traditional programming-oriented introductory course. I believed this problem to be a result of the teaching method used to present the material, not the inadequacies in the abilities of the students.

The format of the present course crystallized in the summer of 1975 during a six-week visit to the laboratory of Professor Seymour Papert of the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. In his laboratory, I observed average 10- and 11-year-old children writing relatively complex programs in the computer language LOGO. This reinforced my belief that any motivated college student should be able to learn programming. Many of the techniques used in this book have their origin in ideas found in the work of Professor Papert.

The material in this book has been used in approximately 24 sections of the introductory computer science course during the period from 1975 to 1980. Although the evidence is rather qualitative, there appears to have been a significant improvement in the performance of students, especially nonscience majors, using the approach. Grades have improved, more programs have been completed, and students report having more fun than in previous years.

I am grateful to my colleagues James H. Keller and E. James Runde, who were willing to use this book while it was still evolving and who gave me valuable feedback. Three of my former students, C. Joseph Bridwell, James M. Coggins, and Cary A. Coutant, helped me in many ways. I am especially grateful to Seymour Papert, who gave me a stimulating place to work during the most crucial part of the writing.

T. RAY NANNEY

TO THE STUDENT

There are many approaches to the study of computing. This book emphasizes the writing of computer programs, but it also presents many other important aspects of computing. The course is essentially nonmathematical and is designed to be useful to both science and nonscience majors. No previous knowledge of computers, programming, or other specialized skills is required.

Regardless of your future interests and activities, it will be difficult (probably impossible) to avoid contact with computers. Consequently, in this book you will be introduced to programming, terminology, computer languages, applications, and social implications of computers. If you work conscientiously to master the material, at the end of the course you should be able to use the computer to solve problems that interest you. You should also have gained insight into many other important topics in computing.

The writing of computer programs is intimately associated with problem solving, so considerable attention will be given to a systematic approach to problem solving. At the end of the course many of you will find it easier to analyze and solve problems. A thorough understanding of the contents of the course will also improve your ability to think logically and critically.

An important side benefit of our study will be an enhanced understanding of the nature of language and the difficulties associated with using language in a precise, unambiguous manner. In fact, to solve a problem using a computer *requires* that language be used with precision. There is considerable truth to the statement that "computer science is more closely associated with linguistics and communication than with mathematics."

Studying computing can and should be exciting and fun. It is, however, quite different from anything you have studied previously; if you get behind, it can be almost impossible to catch up. It can also be a disastrous mistake to wait until the last minute to undertake a programming assignment. A problem that is quite easy for 95% of the class can be time consuming and difficult for you. The only safe assumption is that the problem will take longer than you had planned.

In writing computer programs, the following somewhat contradictory attitude is suggested: strive for perfection, but do not be upset or embarrassed by a mistake. When you write a program, you should try to write it so perfectly that correct answers are obtained when the program is first run on the computer. Yet it is not the nature of human beings to be perfectly accurate. So if you make a mistake, it is not a catastrophe—correct the mistake and rerun the program.

A major goal of this course is to learn to write computer programs using the FORTRAN language. Traditionally, in such a course you would learn a small part of the language and immediately begin programming. In contrast, we will not begin writing FORTRAN programs until an overview of computing has been presented and

some principles have been discussed. This approach appears to have the following advantages:

1. The introductory material will aid you in organizing your thinking about how to solve the problem for which you want to write a program. This is helpful for everyone and is especially valuable for students with nonscientific or nonmathematical backgrounds.
2. The initial study of programming principles helps you to develop and use good programming techniques from the beginning of the course. This is invaluable when the programs become longer and more difficult.
3. The knowledge of programming principles gained in the first part of the course makes a language like FORTRAN easier to learn and reduces frustration. The experimental evidence suggests strongly that students become better programmers because of the introductory material.

CONTENTS

Chapter

LIST OF FIGURES

Figure

LIST OF TABLES

CHAPTER 1 COMPUTERS AND COMPUTER SCIENCE

There is hardly an area of human endeavor that has not been affected by computers. Banks use computers to print your bank statements; utility companies compute your usage of their services and a computer prints your bill; computers are used as inventory control in department stores; airline personnel check your reservations using a computer. They are everywhere—in business, industry, government, education, sciences, and all present-day technology. This widespread use of computers in our society is due primarily to the generality of tasks that computers can perform, the tremendous speed of computer operations, and the decrease in the cost of computers.

1.1 COMPUTERS IN SOCIETY

In 1974, I heard an IBM executive refer to the rapid spread of computer usage as "the insidious revolution." He explained that the general population rarely sees a computer, almost never works with a computer (computer billing being an exception), and is unaware of the diversity of computer applications and their pervasiveness in society. His claim was that without the public's realization, we have become so dependent upon computers that society could not be maintained in its present form without them.

Let us expand our perspective regarding computer applications by briefly considering the history* of modern computing. The first large-scale electronic computer, ENIAC, was completed in 1946 and was used by the United States Army Ordnance Corps primarily to compute ballistic tables. In the early 1950s all computers were owned by the federal government and were used for census studies, weapons calculations, weapons delivery and control, cryptographic applications, nuclear design, nuclear engineering, inventory, and logistic applications. In 1954, the UNIVAC I computer was delivered to General Electric Company for commercial use, and the

*Much of this information has been abstracted from Ruth M. Davis, "Evolution of Computers and Computing," *Science, 195* (March 18, 1977), 1096–1102, copyright 1977 by the American Association for the Advancement of Science; and Saul Rosen, "Electronic Computers: A Historical Survey," *Computing Surveys, 1* (March 1969), 7–36. Copyright 1969, Association for Computing Machinery, Inc., reprinted by permission.

TABLE 1.1 *Number of computers used in the United States*

Year	*Number of computers*
1950	~12
1955	1,000
1960	6,000
1965	30,000
1976	220,000

TABLE 1.2 *Ownership of general-purpose conventional computers within the United States, 1976*

Ownership by industrial classification		*Percent of computers*
Manufacturing industry		31.0
Electric machinery	3.5%	
Nonelectric machinery	4.5%	
Other process manufacturing	9.7%	
Other manufacturing	11.0%	
Transportation equipment	2.3%	
Miscellaneous business		13.3
Advertising, employment, equipment, rental, engineering services, other professional services		
Banking, credit, insurance, real estate, and other financial institutions		13.4
Trade (wholesale and retail)		13.1
Educational institutions (schools, universities, libraries)		5.7
State and local government		5.7
Federal government		3.4
Transportation carriers		2.9
Medical and health services		2.7
Printing and publishing		2.4
Communications		1.9
Utilities (electric, gas, and sanitary services)		1.6
Other professional services		1.9
Petrochemical industry		1.0

Source: Ruth M. Davis, "Evolution of Computers and Computing," *Science, 195* (March 18, 1977), 1100; by permission.

computer explosion had begun. The approximate number of computers in use in the United States at the end of various years is given in Table 1.1. In a 1967 report to the president of the United States,* it was estimated that 80,000 computers would be in use by the end of 1975. This estimate was far too low; improved computer tech-

**Computers in Higher Education*, Report of the President's Science Advisory Committee, The White House, Washington, D.C., February 1967. p. 58.