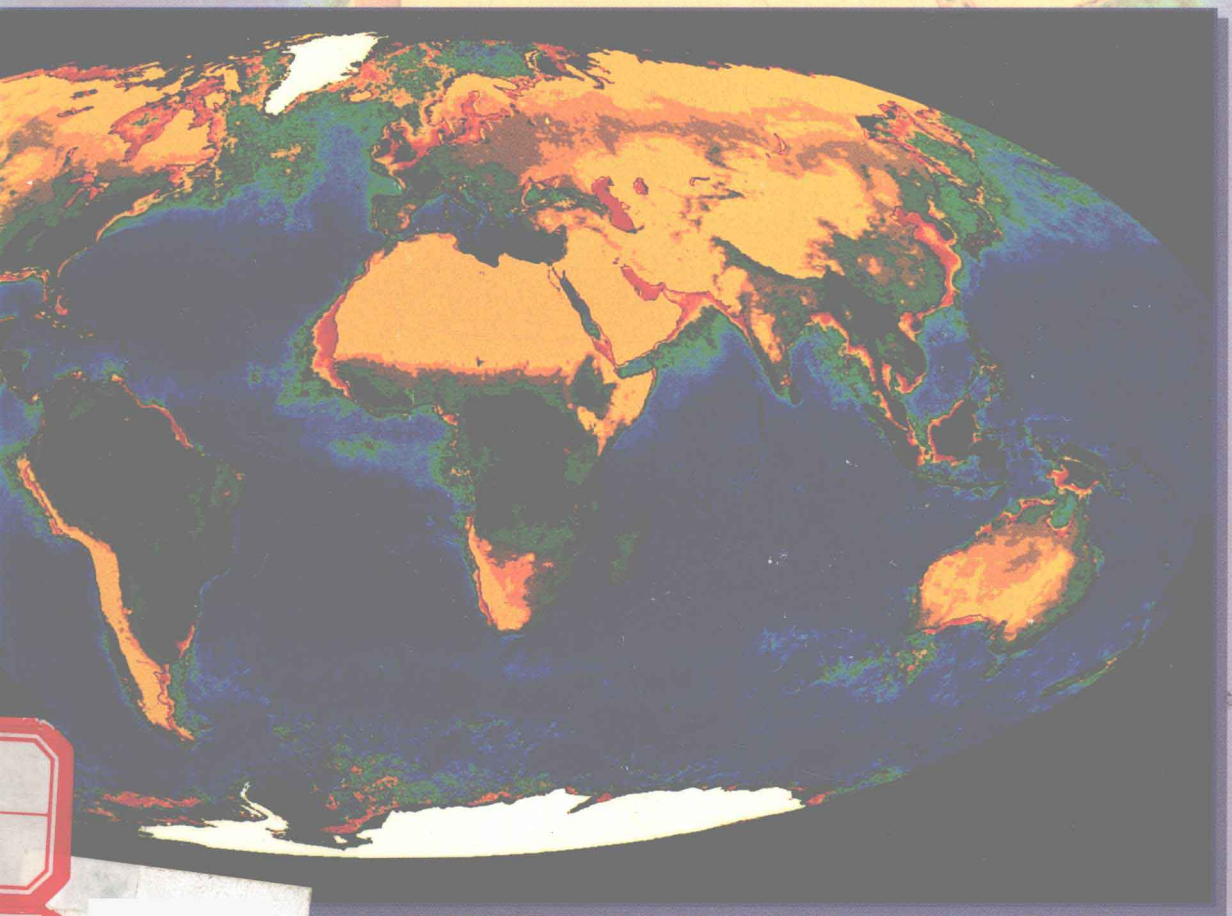
ENGINEERING PROBLEM SOLVING WITH ANSI C

DISK INCLUDED

Fundamental Concepts



Delores M. Etter

Engineering Problem Solving with ANSI C: Fundamental Concepts

Delores M. Etter
Department of Electrical and Computer Engineering
University of Colorado, Boulder

An Alan R. Apt Book

Prentice Hall, Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging-in-Publication Data
Etter, Delores M.
Engineering problem solving with ANSI C: fundamental concepts /
Delores M. Etter.
p. cm.
"An Alan R. Apt book."
Includes bibliographical references and index.
ISBN 0-13-061607-9
1. Engineering--Data processing. 2. C (Computer program language)
I. Title.
TA345.E87 1995
620'.0028553--dc20 94-17480
CIP

Publisher: Alan Apt
Editor-in-Chief: Marcia Horton
Project Manager: Mona Pompili
Developmental Editor: Sondra Chavez
Copy Editor: Peter Zurita
Marketing Manager: Tom McElwee
Design Director: Anne T. Neiglos
Designers: Meryl Poweski , Mona Pompili, Delores M. Etter
Cover Designer: Anthony Gemmelaro
Production Coordinator: Linda Behrens
Editorial Assistant: Shirley McGuire
Cover Photo: Satellite image of Earth's biosphere showing the distribution of vegetation and phytoplankton (the microscopic plants that drift with the ocean currents and that are the basis of the ocean's complex food chain).

A Simon & Schuster Company
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

10 9 8 7 6 5 4 3

ISBN 0-13-061607-9

PRENTICE-HALL INTERNATIONAL (UK) LIMITED, *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL CANADA, INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
SIMON & SCHUSTER ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*

TRADEMARK INFORMATION

In memory of my dearest Mother,
Muerladene Janice Van Camp

Preface

Engineers use computers to solve a variety of problems ranging from the evaluation of a simple function to solving a system of nonlinear equations. C has become the language of choice of many engineers and scientists not only because it has powerful commands and data structures, but also because it can easily be used for system-level operations. Since C is the language that a new engineer is most likely to encounter in a job, it is a good choice for an introduction to computing for engineers. Therefore, this text was written to introduce engineering problem solving with the following objectives:

- to develop a consistent **methodology for solving engineering problems,**
- to present the **fundamental capabilities of C,** the language of choice of many practicing engineers and scientists, and
- to illustrate the problem solving process with C through a variety of **engineering examples and applications.**

To accomplish these objectives, Chapter 1 presents a five-step process that is used consistently in the rest of the text for solving engineering problems. Chapters 2–7 present the fundamental capabilities of C for solving engineering problems. Throughout all these chapters, we present a large number of examples from many different engineering and science disciplines. The solutions to these examples are developed using the five-step process and ANSI C, the standard developed by the American National Standards Institute.

PREREQUISITES

No prior experience with the computer is assumed. The mathematical prerequisites are **college algebra and trigonometry.** Of course, the initial material can be covered much faster if the student has used other computer languages or software tools.

COURSE STRUCTURE

The material in these chapters was selected to provide the basis for a **one-term course** in engineering computing. These chapters contain the essential topics of mathematical computations, control structures, functions, arrays, pointers, and character handling. Students with background in another computer language should be able to complete this material in less than a semester. A minimal

course that provides only an introduction to C can be designed using the nonoptional sections of the text. (Optional sections are indicated in the Contents.) Three ways to use the text, along with the recommended chapter sections, are

- **Introduction to C** Many freshman introductory courses introduce the student to several computer tools in addition to an introduction to a language. For these courses, we recommend covering the nonoptional sections of Chapters 1–5. This material introduces students to the fundamental capabilities of C, and they will be able to write substantial programs using mathematical computations, control structures, functions, and arrays.
- **Problem Solving with C** In a semester course devoted specifically to teaching students to master the C language, we recommend covering all nonoptional sections of Chapters 1–7. This material covers all the fundamental concepts of the C language, including mathematical computations, control structures, functions, arrays, pointers, and character handling.
- **Problem Solving with C and Numerical Techniques** Upper-level students or students who are already familiar with other high-level languages will be able to cover the material in this text very quickly. In addition, they will be able to apply the numerical technique material to their other courses. Therefore, we recommend that these students cover all sections of Chapters 1–7, including the optional material.

PROBLEM-SOLVING METHODOLOGY

The **emphasis on engineering and scientific problem solving** is an integral part of the text. Chapter 1 introduces a **five-step process for solving engineering problems** using the computer:

1. State the problem clearly.
2. Describe the input and output information.
3. Work a simple example by hand.
4. Develop an algorithm and convert it to a computer program.
5. Test the solution with a variety of data.

To reinforce the development of problem solving skills, each of these five steps is clearly identified each time that a complete engineering problem is solved. In addition, **top-down design** and **stepwise refinement** are presented with the use of **decomposition outlines, pseudocode,** and **flowcharts.**

ENGINEERING AND SCIENTIFIC APPLICATIONS

Throughout the text, emphasis is placed on incorporating real-world engineering and scientific examples and problems. This emphasis is centered around a theme of **grand challenges,** which include:

- prediction of weather, climate, and global change
- computerized speech understanding
- mapping of the human genome
- improvements in vehicle performance
- enhanced oil and gas recovery

Each chapter begins with a photograph and a discussion of some aspect of one of these grand challenges that provides a glimpse of some of the exciting and interesting areas in which engineers might work. Later in the chapter, we solve a problem that not only relates to the introductory problem, but also has applications in other problem solutions. The grand challenges are also referenced in many of the other examples and problems.

ANSI C

The statements presented and all programs developed use the C standards developed by the **American National Standards Institute.** By using ANSI C, students learn to write **portable** code that can be transferred from one computer platform to another. Many of the capabilities of ANSI C are contained in the Standard C Library; some of these capabilities are discussed in the text and additional ones are discussed in Appendix A.

SOFTWARE ENGINEERING CONCEPTS

Engineers and scientists are expected to develop and implement **user-friendly** and **reusable** computer solutions. Learning software engineering techniques is therefore crucial to successfully developing these computer solutions. **Readability** and **documentation** are stressed in the development of programs. Additional topics that relate to software engineering issues are discussed throughout the text and include issues such as **software life cycle, portability, maintenance, modularity, recursion, abstraction, reusability, structured programming, validation,** and **verification.**

THREE TYPES OF PROBLEMS

Learning any new skill requires practice at a number of different levels of difficulty. We have developed three types of exercises that are used throughout the text to develop problem solving skills. The first set of exercises are **Practice!** problems. These are short-answer questions that relate to the section of material just presented. Most sections are immediately followed by a set of Practice! problems so that students can determine if they are ready to continue to the next section. Complete solutions to all the Practice! problems are included at the end of the text.

The **Modify!** problems are designed to provide hands-on experiences with the programs developed in the Problem Solving Applied sections. In these sec-

tions, we develop a complete C program using the five-step process. The Modify! problems ask students to run the program (which is stored on the text diskette) with different sets of data to test their understanding of how the program works and of the relationships among the engineering variables. These exercises also ask the students to make simple modifications to the program and then run the program to test their changes. Selected solutions to some of the Modify! problems are included at the end of the text.

Finally, each chapter ends with a set of **end-of-chapter** problems. These are new problems that relate to a variety of engineering applications, and the level of difficulty ranges from very straightforward to longer project assignments. Each problem requires that the student develop a complete C program or function. Engineering data sets are included for many of the problems to use in testing. Selected solutions to some of the end-of-chapter problems are included at the end of the text.

STUDY AND PROGRAMMING AIDS

Margin notes are used to help the reader not only identify the important concepts, but also to easily locate specific topics. In addition, margin notes are used to identify programming style guidelines and debugging information. Style guidelines show students how to write C programs that incorporate good software discipline; debugging sections help students recognize common errors so that they can avoid them. The programming style notes are indicated with the margin note *Style,* and the debugging notes are indicated with a **bug icon.** Each Chapter Summary contains a summary of the style notes and debugging notes, plus a list of the **Key Terms** from the chapter and a **C Statement Summary** of the new statements to make the book easier to use as a reference. The combined list of these key terms, along with their definitions, is included in a **Glossary** at the end of the text.

OPTIONAL NUMERICAL TECHNIQUES

Numerical techniques that are commonly used in solving engineering problems are also discussed in optional sections in the chapters, and include **interpolation, linear modeling (regression), root finding,** and the **solution to simultaneous equations.** The concept of a **matrix** is also introduced and then illustrated using a number of examples. All of these topics are presented assuming only a trigonometry and college algebra background.

MATLAB AND VISUALIZATION

The visualization of the information related to a problem and its solution is a critical component in understanding and developing the intuition necessary to be a creative engineer. Therefore, we have included a number of plots of data throughout the text to illustrate the relationships of the information needed to solve specific problems. All the plots were generated using MATLAB, a powerful environment for numerical computations, data analysis, and visualization. We

have also included an appendix that shows how to generate a simple plot from data that have been stored in an ASCII data file; this ASCII file could be generated with a word processor or it could be generated by a C program. If a course is planned that will include material on both C and MATLAB, a special package that includes this text and *Engineering Problem Solving with MATLAB®* is available from Prentice Hall.

APPENDICES

To further enhance reference use, the appendices include a number of important topics. Appendix A contains a discussion of the components in the **ANSI C Standard Library.** Appendix B presents the **ASCII character codes.** Appendix C shows how to use **MATLAB to plot data from ASCII files;** this allows students to generate ASCII files with their C programs and to then plot the values using MATLAB. Finally, Appendix D contains a list of **references** used throughout the text.

SOFTWARE DISKETTE

Since the text includes a large number of examples that illustrate the various engineering applications, a **program** and **data file diskette** is included, which contains all the example programs and data files so that students can have immediate access to them. Also, additional data files that are referenced in the end-of-chapter problems are included on the diskette, including climatology data and speech signals. A **diskette icon** is included in the margin next to items that are contained on the diskette. The files on the diskette are ASCII files in a DOS format; they are ready to upload to other computer systems or to be converted to a Macintosh format.

STUDENT WORKBOOK

An **optional student workbook** has been developed to provide additional exercises for mastering the material in the textbook. This workbook contains **fill-in-the-blank problems, true-or-false problems,** and **multiple-choice problems** to test the student's understanding of the material. In addition, a number of programming problems are presented with **short but instructive computer solutions.** Students are then asked to provide memory snapshots at different points in the solutions or to show the output generated by the program. Multiple solutions are given for many of the problems to illustrate the different ways in which problems can be solved. The material in this workbook has been organized to follow the outline of the text, and thus assignments can be made in the workbook to follow the progression through the text.

INSTRUCTOR'S MANUAL

An **Instructor's Manual** is available, which contains complete solutions to all the Modify! problems and end-of-chapter problems. Also, transparency masters and a disk are included to assist in preparing lecture material.

NONTECHNICAL SKILLS

The engineer of the 21st century needs many skills and capabilities in addition to the technical ones learned in an engineering program. In Chapter 1 we present a brief discussion on some of these nontechnical skills that are so important to engineers. Specifically, we discuss developing both oral and written **communications skills,** understanding the **design/process/manufacture path** that takes an idea and leads to a product, working in interdisciplinary teams, understanding the **world and its marketplace,** the importance of **synthesis as well as analysis,** and the importance of **ethics** and other **societal concerns** in engineering solutions. While this text is devoted primarily to teaching problem solving skills and the C language, we have attempted to tie these other nontechnical topics into many of the problems and discussions in the text.

ACKNOWLEDGMENTS

I appreciate the encouragement of a number of people relative to the development of this text, and would like to especially recognize Bernard Goodwin (who was the first person to begin telling me that I should write a C text) and Alan Apt (who convinced me that the time was right). I also want to acknowledge the outstanding work of the publishing team at Prentice Hall, including Tom McElwee, Marcia Horton, Gary June, Kathleen Schiaparelli, Mona Pompili, Sondra Chavez, Alice Dworkin, and Mike Sutton. This text has been significantly improved by the suggestions and comments of the reviewers, who included Arnold Robbins (Georgia Tech College of Computing), Avelino Gonzalez (University of Central Florida), Thomas Cargill (Private Consultant), Jonathan Haines (Software/Hardware/Systems Consultant), Thomas Walker (Virginia Polytechnic Institute and State University), Christopher Skelly (Insight Resource Inc.), Betty Barr (The University of Houston), John Cordero (University of Southern California), A. R. Marundarajan (Cal Poly, Pomona), Lawrence Genalo (Iowa State University), Karen Davis (University of Cincinnati), Petros Gheresus (General Motors Institute), Leon Levine (UCLA), Harry Tyrer (University of Missouri—Columbia), Caleb Drake (University of Illinois at Chicago), John Miller (University of Michigan—Dearborn), Elden Heiden (New Mexico State University), Joe Hootman (University of North Dakota), and Nazeih Botros (Southern Illinois University).

I also want to recognize the important contributions of three groups of students (ranging from freshmen who had never used the computer, to undergraduates who had done a little computing with other languages, to graduate students who wanted to use C to do their research) who class-tested and carefully reviewed the various drafts of the manuscript and gave their feedback on the explanations, the examples, and the problems. A final note of gratitude goes to my husband, a mechanical engineer, for his help in developing some of the engineering application problems, and to my daughter, a veterinarian student, for her help in developing some of the genetic engineering problems.

Delores M. Etter
Department of Electrical/Computer Engineering
University of Colorado, Boulder

Contents

*Optional Section

*Optional Section

*Optional Section

*Optional Section

Engineering Applications

Aerospace Engineering

Biomedical Engineering

Chemical Engineering

Electrical Engineering

Environmental Engineering