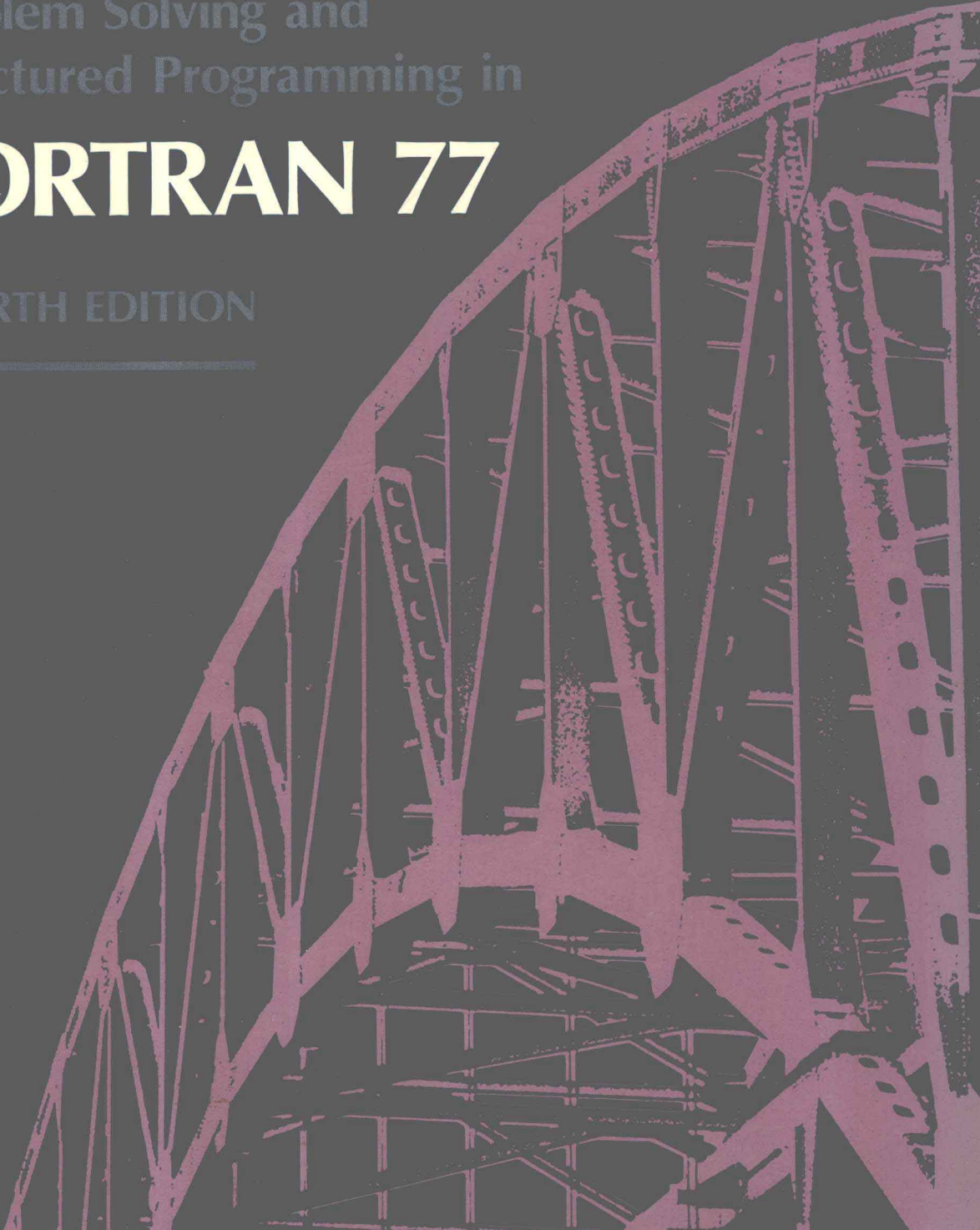


KOFFMAN / FRIEDMAN

Problem Solving and
Structured Programming in

FORTRAN 77

FOURTH EDITION



Problem Solving and Structured Programming in **FORTRAN 77**

FOURTH EDITION

Elliot B. Koffman
Frank L. Friedman

Temple University

江苏工业学院图书馆
藏书章



Addison-Wesley Publishing Company, Inc.

Reading, Massachusetts Menlo Park, California New York
Don Mills, Ontario Wokingham, England Amsterdam Bonn
Sydney Singapore Tokyo Madrid San Juan

Keith Wollman, Sponsoring Editor
Bette J. Aaronson, Production Supervisor
Nancy Benjamin, Production Editor
Joyce Cameron Weston, Text and Cover Design
Roy Logan, Manufacturing Supervisor

Library of Congress Cataloging-in-Publication Data

Koffman, Elliot B.

Problem solving and structured programming in
FORTRAN 77.

1. FORTRAN 77 (Computer program language).
 2. Structured programming. I. Friedman, Frank L.
- II. Title.

QA76.73.F25K64 1990 005.26 89-18494
ISBN 0-201-51216-5

The programs and applications presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Copyright © 1990, 1987, 1981, 1977 by Addison-Wesley
Publishing Company, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers. Printed in the United States of America.

Reprinted with corrections December, 1990

4 5 6 7 8 9 10 DO 9594939291

To our families
Caryn, Richard, Deborah, and Robin Koffman
and
Martha, Shelley, and Dara Friedman

Preface

This is a textbook for a first course in problem solving and programming methods using the FORTRAN language. It assumes no prior knowledge of computers or programming. A course in high-school algebra is sufficient mathematics background for most of the material in this textbook, except for sections of the last chapter. In other chapters, examples that require more mathematical background are clearly marked.

The most significant change in this edition is a substantial increase in the number of engineering and science examples, including at least one new case study in each chapter. We have also retained most of the general examples. The net result is a textbook that can be used either in a FORTRAN course for engineering and science majors or in an introductory programming course for a general audience.

Problem Solving, Structured Programming, and Modularization

The first edition of this textbook was published in 1977, the same year that FORTRAN 77 became the standard version of the FORTRAN language. The first edition pioneered the teaching of problem solving and structured programming in a FORTRAN course, a tradition that has continued in successive editions.

Problem solving continues to be emphasized in this edition. Chapter 2 contains a new section on the art and science of problem solving, which includes a discussion of the engineering and scientific method.

This edition has an increased emphasis on modular programming, which is an important component of structured programming. Chapter 4 introduces the use of library functions; Chapter 5 is a new chapter on user-defined functions and subroutines. In this edition, we discuss both kinds of FORTRAN subprograms before coverage of arrays (Chapter 6). This allows us to use subprograms in many of the program examples that manipulate arrays.

Organization of the Book

The first seven chapters represent the core of the book and should be studied by all students. Students in a general course should skip many of the engineering examples; students in a predominately engineering or science section may skip some of the business-oriented case studies.

Chapter 1 provides an overview of computer hardware and software, including an expanded introduction to the history of computers. It contains an introduction to FORTRAN statements and data types, and the use of FORTRAN in different programming environments. It also introduces the essentials of using files, so that students can write programs that access data files prepared by their instructor or send their program results to an output file.

Chapter 2 describes problem solving, algorithm refinement, and decisions in programs. The chapter begins by discussing problem solving in engineering and science followed by a general discussion of problem solving and algorithm refinement. It also shows how to write conditions and implement decisions in FORTRAN using the IF statement. The chapter covers all aspects of the use of the IF statement, including the Block IF form for implementing decisions with multiple alternatives.

Chapter 3 describes two control structures for implementing loops: the DO loop and the WHILE loop. As in earlier editions, we include the WHILE loop because we believe students should know how to use this control structure, which is fundamental to structured programming. Even though the WHILE loop is not part of the FORTRAN 77 standard, it is provided as an optional feature in many versions of FORTRAN 77 and will be included in the new FORTRAN standard (FORTRAN 8X). For those who do not have access to this control structure, we show how to implement it in standard FORTRAN 77.

Chapter 4 provides additional details on FORTRAN data types, including the use of library functions in expressions. The chapter also introduces the FORMAT statement.

Chapter 5 is a new chapter on subprograms. The chapter covers both function and subroutine subprograms. It introduces preconditions and postconditions as part of a function's documentation. It also shows how to use structure charts to document the flow of control and data between the main program and its subprograms.

Chapter 6 introduces the array, focusing on one-dimensional arrays. Chapter 7 covers advanced features of arrays, including multidimensional arrays, searching an array, and sorting an array. The chapter also discusses additional subprogram concepts, including COMMON storage and the SAVE statement. The chapter ends with a discussion of random number generation and computer simulation.

The remaining chapters cover advanced material. These chapters do not have to be studied in any particular order. If time permits, instructors in classes with a general audience should concentrate on Chapters 8 and 9, which cover additional features of formatting, file processing, and string manipulation. Instructors in classes with science and engineering students should spend more time on Chapters 10 and 11, which cover graphing, computer-aided design, and numerical methods.

FORTRAN Versions and Computers

The text uses FORTRAN 77 as its language of instruction rather than FORTRAN 8X. The reason for this choice is the current widespread availability of FORTRAN 77 compilers. At the time of publication of this book, FORTRAN 8X was still awaiting final approval so there were no FORTRAN 8X compilers. As mentioned earlier, we have included one important new control structure from FORTRAN

8X, the `WHILE` loop. Appendix D summarizes many of the important features of FORTRAN 8X.

The text is not oriented toward any specific computer. Chapter 1 discusses booting a personal computer and logging onto a timeshared computer. Most of the examples are written as interactive programs, which is the most prevalent mode of programming today. All of the new programs have been tested on an IBM-compatible personal computer using Microsoft FORTRAN 77. Appendix C describes how to use this popular compiler for microcomputers.

Pedagogical Features

We employ many pedagogical features to enhance the usefulness of this book as a teaching tool. Some of these features are discussed below.

End of Section Exercises: Most sections end with a number of self-check exercises. These include exercises that require analysis of program segments as well as short programming exercises. Answers to these exercises appear in the back of the book.

End of Chapter Exercises and Projects: Each chapter ends with a set of quick-check exercises with answers. There are also chapter review exercises whose solutions appear in the back of the book. Finally, there is a set of programming projects whose solutions appear in the instructor's manual.

Examples and Case Studies: The book contains a large number and variety of programming examples. Whenever possible, examples contain complete programs or subprograms rather than incomplete program fragments. There are also substantial case studies that help a student integrate and apply concepts studied over several chapters.

Syntax Display Boxes: The syntax displays describe the syntax and semantics of each new FORTRAN feature and also provide examples.

Program Style Displays: The program style displays discuss issues of good programming style.

Error Discussions and Chapter Reviews: Each chapter ends with a discussion of common programming errors. A chapter review includes a table of new FORTRAN statements.

Appendices and Supplements

A reference table of FORTRAN statements appears on the inside covers of the book. There are appendices on FORTRAN Library Functions and Character Sets. Also, there is an appendix on MS-DOS and the Microsoft FORTRAN 77 compiler and one covering FORTRAN 8X.

Supplements include an instructor's manual, transparency masters, and a program disk with all the programs that appear in the book. Use the reference numbers below to order these supplements from your Addison-Wesley sales representative.

Instructor's Manual: 1-201-51217-3

Transparency Masters: 1-201-52147-4

Program Disk: 1-201-51218-1

We are very grateful to Thomas Cunningham at Indiana University of Pennsylvania for his work on producing the instructor's manual.

Acknowledgments Many people participated in the development of this book. Professors Bart Childs and Glen Williams, and Mr. Mitch Fincher of Texas A & M University carefully reviewed the previous edition and provided many suggestions for improvement. They also prepared a host of engineering examples, exercises, case studies, and programming projects for inclusion in the book. Their help was invaluable in increasing the number of engineering and science applications presented.

The principal reviewers were most essential in suggesting improvements and finding errors. The principal reviewers include: Professor James Allert, University of Minnesota; Walter Brainerd, UNICOMP; Professor William H. Dodge, Rensselaer Polytechnic University; Professor George J. Dudas, Pennsylvania State University, Behrend College; Professor Sundaresan Jayaraman, Georgia Institute of Technology; Professor Richard A. Lejk, University of North Carolina at Charlotte; Professor Ernie Rilki, William Rainer Harper College; Abolfazl Sirjani, IBM; Professor Conrad L. White, Northern Illinois University.

We would also like to acknowledge the contribution of Professor Richard Epstein of George Washington University. Professor Epstein provided the original draft of much of the material in Chapters 10 and 11.

The personnel at Addison-Wesley responsible for the production of this book worked diligently to meet a very demanding schedule. The sponsoring editor, Keith Wollman, was closely involved in all phases of the manuscript preparation and provided much help and guidance. His assistant, Debbie Lafferty, did an excellent job of coordinating the reviewing process. Bette Aaronson supervised the design and production of the book, while Nancy Benjamin coordinated the conversion of the manuscript to a finished book.

E.B.K.

F.L.F.

Application Areas

Engineering/Science

Converting inches to centimeters, 20
Heat loss and pressure loss formulas, 40
Parallel resistance, 59
Testing column safety, 83
Computing compass bearing, 101
Physics of falling bodies, 121, 145
Temperature conversion, 137
Computing radiation levels, 161
Hydrostatic pressure, 176
Engineering properties of lumber, 176
Theory of relativity, 190
Shooting an arrow through a window, 193
Flow of water through a pipe, 215, 233
Computing maximum tensile loads, 221, 265
Radioactive decay, 232
Image enhancement, 346
Simulation and random numbers, 374
Movement of radioactive particles, 393
The game of Life, 394
Survey of land boundaries, 427
Wind chill factor, 449
Parametric equations, 508
Contour plots, 511, 515
Computer-aided design, 520
Interference pattern, 526

Business

Value of coin collection, 33
Payroll computations, 30, 43, 78, 89, 129
Price of concrete, 91
Savings certificate interest, 146, 231
Balancing a checkbook, 156
Mortgage interest, 231, 399
Computing electric company bills, 282
Processing time cards, 283
Budget categories, 322
Airline reservations, 393
Inventory control, 395
Bookstore inventory, 420, 438
Merging mailing lists, 422

Mathematics and Statistics

Area and circumference of a circle, 69
Average of measurements, 125
Computing value of e , 138
Powers of 2, 144
Square root approximation, 175
Roots of a quadratic equation, 191
Factorial function, 236
Combinations of r items from n items, 240
Rounding a number, 244
Prime numbers, 251
Area of a triangle, 280
Average value and standard deviation, 292
Area of a polygon, 310
Plotting a function, 319, 498
Adding matrixes, 345
Estimating π , 376
Dot product of arrays, 392
Trigonometric functions, 452
Roots of equations, 528
Newton's method, 536
Vector and matrix operations, 537, 539
Linear equations, 543
Linear regression and correlation, 553
Numerical integration, 561

General

Smallest of three numbers, 81
Sum of integers, 117, 133
Counting down, 134
Worm attacks apple, 140
Product of non-zero values, 151, 154
Guessing a number in an interval, 200
Registering voters, 217
Sorting, 268
Largest value in a list, 304
Tic-tac-toe display, 341
Course enrollment, 343
Searching an array, 355
Sorting an array, 357
Student grading, 360
Cryptogram generator, 474
Syntax analysis, 477
Text editor, 481

Contents

1. Introduction to Computers and Programming 1

- 1.1 Electronic Computers Then and Now 2
- 1.2 Components of a Computer 7
- 1.3 Problem Solving and Programming 13
- 1.4 Programming Languages 14
- 1.5 Processing a High-Level Language Program 16
- 1.6 Introduction to FORTRAN 18
- 1.7 FORTRAN Programs 30
- 1.8 Introduction to Data Types and Expressions 35
- 1.9 Introduction to Files and Batch Processing 42
- 1.10 Using the Computer 47
- 1.11 Common Programming Errors 50
- Chapter Review 53

2. Solving Problems and Making Decisions 61

- 2.1 The Art and Science of Problem Solving 62
- 2.2 Representing and Refining Algorithms 64
 - Case Study: Robbie Serving Breakfast 64
 - Case Study: Changing a Flat Tire 66
- 2.3 Problem Solving Illustrated 68
 - Case Study: Finding the Area and Circumference of a Circle 69
- 2.4 Comments and Documentation 71
- 2.5 Decision Steps, Conditions, and IF Statements 73
- 2.6 Solving Problems with Decision Steps 78
 - Case Study: Modified Payroll Problem 78
 - Case Study: Finding the Smallest of Three Numbers 81
 - Case Study: Testing Column Safety 83
- 2.7 Tracing an Algorithm or Program 86
- 2.8 Problem-Solving Strategies 88
 - Case Study: Computing Overtime Pay 89
 - Case Study: Computing the Price of Concrete 91
- 2.9 The General IF Statement 93
 - Case Study: Computing Compass Bearings 102
- 2.10 Common Programming Errors 106
- Chapter Review 106

3. ■ Repetition and Loops 113

- 3.1 Repetition in Programs 114
 - Case Study: Sum of Integers 117
 - Case Study: Physics of Falling Bodies 121
- 3.2 Generalizing a Solution 125
 - Case Study: Average of a Set of Measurements 125
- 3.3 Repeating a Program Body 129
 - Case Study: Computing the Payroll for Several Employees 129
- 3.4 The General DO Loop 133
- 3.5 The WHILE Loop 140
- 3.6 Nested Control Structures 152
- 3.7 Problem Solving Illustrated 155
 - Case Study: Balancing a Checkbook 156
 - Case Study: Computing Radiation Levels 161
- 3.8 Debugging and Testing Programs 165
- 3.9 Common Programming Errors 166
 - Chapter Review 168

4. ■ Simple Data Types 178

- 4.1 Declaring a Data Type 179
- 4.2 Numeric Data Types—INTEGER and REAL 181
- 4.3 Functions in Arithmetic Expressions 186
- 4.4 LOGICAL Data Type 194
- 4.5 CHARACTER Data Type 202
- 4.6 Introduction to FORMAT Statements 208
- 4.7 Problem Solving Illustrated 217
 - Case Study: Registered Voters List 217
 - Case Study: Computing Maximum Tensile Loads 221
- 4.8 Common Programming Errors 224
 - Chapter Review 226

5. ■ Subprograms 234

- 5.1 User-Defined Functions 235
- 5.2 Top-Down Design Using Functions 251
 - Case Study: Finding Prime Numbers 251

- 5.3 Defining a Subroutine 257
- 5.4 Calling a Subroutine 260
- 5.5 Structure Chart 264
 - Case Study: Computing Maximum Tensile Loads with Subprograms 265
- 5.6 Multiple Calls to a Subroutine 268
 - Case Study: Simple Sorting Problem 268
- 5.7 Debugging and Testing a Program System 272
- 5.8 Common Programming Errors 275
- Chapter Review 276

6. Arrays 284

- 6.1 Declaring and Referencing Arrays 285
- 6.2 Array Subscripts 289
- 6.3 Using DO Loops to Process Arrays 291
- 6.4 Input and Output of Arrays 295
- 6.5 Array Arguments 303
 - Case Study: Finding the Area of a Polygon 310
- 6.6 The DATA Statement 316
- 6.7 Array Processing Illustrated 322
 - Case Study: Budget Problem 322
- 6.8 Common Programming Errors 328
- Chapter Review 330

7. More Arrays and Subprograms 338

- 7.1 Multidimensional Arrays 339
- 7.2 Multidimensional Array Arguments 345
 - Case Study: Image Enhancement 346
- 7.3 Searching an Array 355
- 7.4 Sorting an Array 357
- 7.5 Using Subprograms in a Large-Scale Problem 360
 - Case Study: Grading Problem 360
- 7.6 The SAVE Statement and Simulation 374
- 7.7 COMMON Blocks 378
- 7.8 Common Programming Errors 385
- Chapter Review 386

8. ■■■■ Formats and Files 397

- 8.1 Review of Formatted Output 398
- 8.2 Input Formats 401
- 8.3 Using Formats with Arrays 408
- 8.4 Sequential Files 413
- 8.5 Problem Solving Illustrated 422
 - Case Study: Merging Files 422
 - Case Study: Land Boundary Survey 427
- 8.6 Direct Access Files 436
- 8.7 Updating a Direct Access File 440
 - Case Study: Direct Access File Update 440
- 8.8 Common Programming Errors 444
 - Chapter Review 445

9. ■■■■ String Manipulation 457

- 9.1 Character String Declaration 458
- 9.2 Substrings 459
- 9.3 Character Expressions 462
- 9.4 String Comparisons 465
- 9.5 String Length and Search Functions 470
- 9.6 Examples of String Manipulation 474
 - Case Study: Generating Cryptograms 474
 - Case Study: Scanning a DO Loop Header 477
 - Case Study: Text Editing Problem 481
- 9.7 Common Programming Errors 485
 - Chapter Review 487

10. ■■■■ Plotting Functions and Computer-Aided Design 497

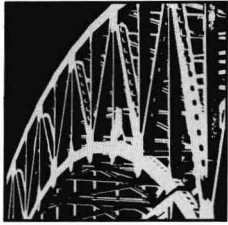
- 10.1 Plotting Functions of One Variable 498
- 10.2 Plotting Parametric Equations 508
- 10.3 Plotting Functions of Two Variables 509
 - Case Study: Contour Plot for a Hemispherical Surface 515

10.4	Introduction to Computer-Aided Design	520
10.5	Common Programming Errors	521
	Chapter Review	521

11. ■ Introduction to Numerical Methods 527

11.1	Finding Roots of Equations	528
11.2	Vectors and Matrixes	537
11.3	Solving Systems of Linear Equations	543
11.4	Linear Regression and Correlation	553
11.5	Numerical Integration	561
11.6	Using Numerical Methods Libraries	567
11.7	Double Precision and Complex Numbers	569
11.8	Common Programming Errors	576
	Chapter Review	577

Appendix A	FORTRAN Library Functions	585
Appendix B	Character Sets	588
Appendix C	Introduction to MS-DOS and the Microsoft FORTRAN Compiler	590
Appendix D	FORTRAN 8X	594
	Answers to Self-Check Exercises	599
	Answers to Review Questions	634
	Index	645



1. Introduction to Computers and Programming

- 1.1 Electronic Computers Then and Now
- 1.2 Components of a Computer
- 1.3 Problem Solving and Programming
- 1.4 Programming Languages
- 1.5 Processing a High-Level Language Program
- 1.6 Introduction to FORTRAN
- 1.7 FORTRAN Programs
- 1.8 Introduction to Data Types and Expressions
- 1.9 Introduction to Files and Batch Processing
- 1.10 Using the Computer
- 1.11 Common Programming Errors
- Chapter Review

IN THIS chapter, we introduce computers and computer programming. We begin with a brief history of computers and a description of the major components of a computer, including memory, central processor, input devices, and output devices. We also discuss how information is represented in a computer and how it is manipulated.

Following this introduction, we begin a discussion of the main topics of this book: problem solving, programming, and FORTRAN. We first discuss problem solving with a computer. Then languages for computer programming are described. Finally, some simple computer operations are presented along with some short FORTRAN programs that demonstrate these operations. The FORTRAN programs contain statements for reading and displaying information and for performing simple computations.

Also described are the steps involved in creating a FORTRAN program and the roles performed by special programs that are part of a computer system. These programs include the operating system, compiler, editor, and loader.

1.1 Electronic Computers Then and Now

It is difficult to live in today's society without having some contact with computers. Computers are used to provide instructional material in schools, print transcripts, send out bills, reserve airline and concert tickets, play games, and even help authors write books.

However, it wasn't always this way. Just a short time ago, computers were fairly mysterious devices that only a small percentage of our population knew much about. Computer "know-how" turned around when advances in solid-state electronics led to large reductions in the size and cost of electronic computers. Today, a personal computer (see Fig. 1.1), which costs less than \$3000 and sits on a desk, has as much computational power as one that 10 years ago would cost more than \$100,000 and would fill a 9' × 12' room. This price reduction is even more remarkable when we consider the effects of inflation over the last decade.

If we take the literal definition for a *computer* as a device for counting or computing, then the abacus might be considered the first computer. However, the first electronic digital computer was designed in the late 1930s by Dr. John Atanasoff at Iowa State University. Atanasoff designed his computer to perform mathematical computations for graduate students.

The first large-scale, general-purpose electronic digital computer, called the ENIAC, was built in 1946 at the University of Pennsylvania with funding supplied by the U.S. Army. The ENIAC was used for computing ballistics

Figure 1.1. *IBM Personal Computer*



tables, for weather prediction, and for atomic energy calculations. The ENIAC weighed 30 tons and occupied a 30' \times 50' space (see Fig. 1.2).

Although we are often led to believe otherwise, computers cannot think! They are basically devices for performing computations at incredible speeds (more than one million operations per second) and with great accuracy. However, in order to accomplish anything useful, a computer must be *programmed*, or given a sequence of explicit instructions (the *program*) to carry out.

To program the ENIAC it was necessary to connect hundreds of wires and arrange thousands of switches in a certain way. In 1946, Dr. John von Neumann of Princeton University proposed the concept of a *stored program computer* in which the instructions of a program would be stored in computer memory rather than be set by wires and switches. Since the contents of computer memory can be easily changed, it would not be nearly as difficult to reprogram this computer to perform different tasks as it was to reprogram the ENIAC. Von Neumann's design is the basis of the digital computer as we know it today.

Brief History of Computers

Table 1.1 lists some of the important milestones along the path from the abacus to modern-day electronic computers. We often use the term *first gener-*