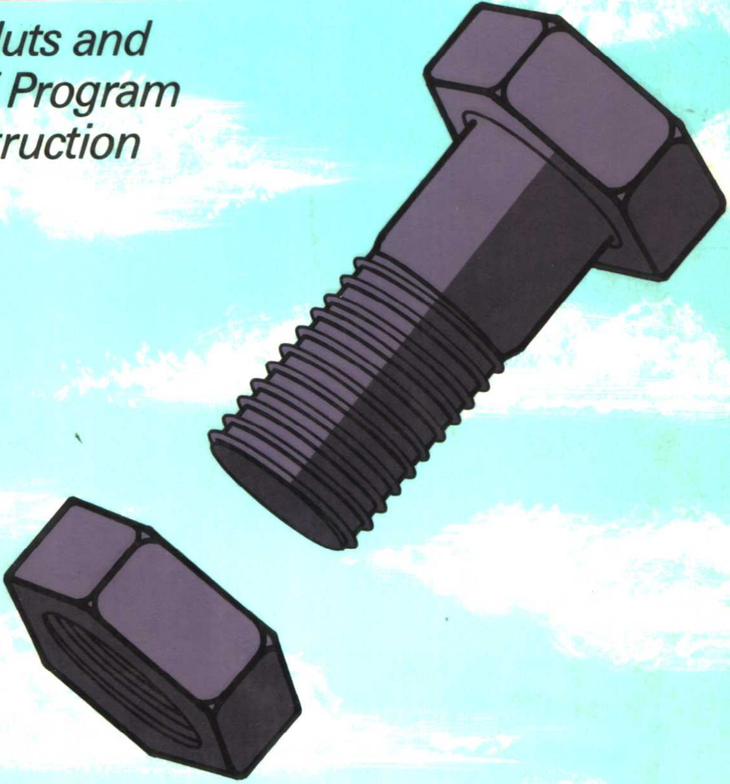
Turbo Pascal 6.0

The Nuts and Bolts of Program Construction

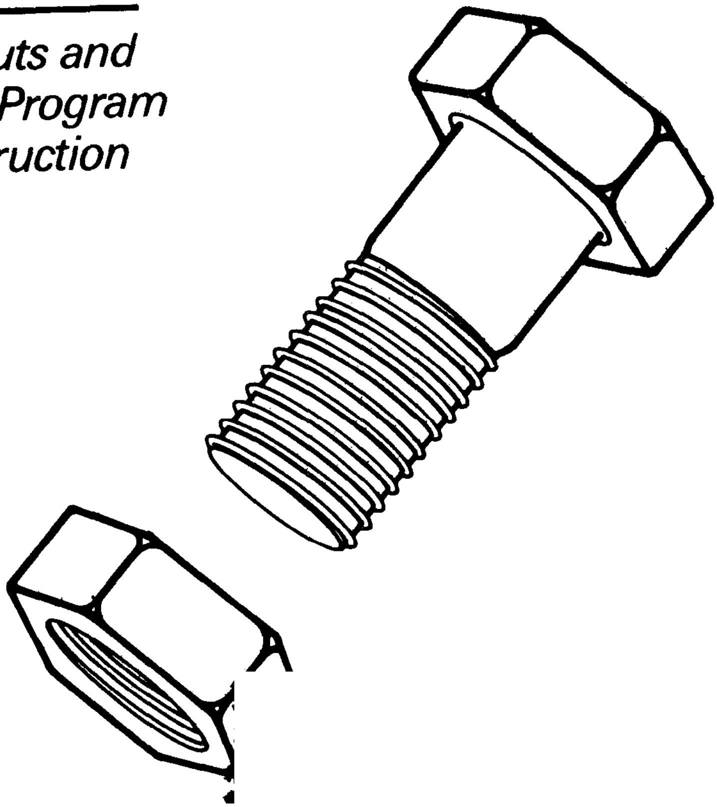


John DiElsi

Includes 5.0 & 5.5 Supplement

Turbo Pascal 6.0

The Nuts and Bolts of Program Construction



John DiElsi

McGraw-Hill, Inc.

New York St. Louis San Francisco Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

TURBO PASCAL 6.0: The Nuts and Bolts of Program Construction

5 6 7 8 9 0 **DOH DOH** 9 4 3

ISBN 0-07-557790-9

This book is printed on acid-free paper.

This book was set in Garamond type by Advanced Presentations.
The editor was David M. Shapiro;
production coordination was done by Linda Hauck and Simmon Factor.
The cover was designed by Seventeenth Street Studios.
Interior design and illlustrations were done by Advanced Presentations.
Development and production management were provided by Cole
and Associates.
R. R. Donnelley & Sons Company was printer and binder.

Library of Congress Cataloging-in-Publication Data

DiElsi, John J.
Turbo Pascal 6.0: the nuts and bolts of program construction /
John DiElsi.
p. cm.
ISBN 0-07-557790-9
1. Pascal (Computer program language) 2. Turbo Pascal (Computer program) I. Title. II. Title: Turbo Pascal 6.0
QA76.73.P35D54 1991
005.369—dc20 90-44654

Preface

This text is meant for an introductory course in structured programming using Turbo Pascal 6.0 and its integrated development environment (IDE) called the Programmer's Platform. Version 6.0, the most recent version of Turbo Pascal, is a significant improvement over the previous two versions, 5.0 and 5.5. This programming system offers tools that make it easier to learn proper programming skills and use them in creating well-written programs. Although other texts claim to be Turbo Pascal compatible, this one incorporates the programming tools provided by the IDE into the programming topics in a natural way, not as a special section in a chapter or an appendix.

The text also can be used for instruction in Microsoft's QuickPascal. All the program solutions written for Turbo Pascal compile and execute, *without modification,* in QuickPascal. Although QuickPascal and Turbo Pascal perform essentially the same functions, the programming environment for QuickPascal (menus, menu choices, windows, etc.) differs from the Turbo Pascal IDE.

To be productive, any language must conform to a set of rules governing its use. But this is not enough. You may be able to discern the meaning of an English sentence even though it is ungrammatical, but a computer cannot interpret the statements in a programming language unless they are "grammatically" correct. And though most people can write grammatically correct sentences in English or some other human language, very few are professional writers. In computer languages, the need to write programs that are not only "grammatically" correct but polished as well is even more important.

This text emphasizes the creation of understandable, reliable, and maintainable Pascal programs. It also provides the tools to do the job. Its informal writing style will help the reader understand important programming concepts while minimizing the jargon and technical language that can cause misinterpretation. The numerous examples provided are explained in detail. In addition, there are completely developed problem solutions that show how to apply these concepts in practical ways. This emphasis on detail—*the nuts and bolts of program construction*—is vital to the creation of well-designed, well-written solutions to practical problems. Turbo Pascal is a vehicle; the problem-solving skills are the goal.

Summary of Contents

Chapter 0 introduces those portions of the Turbo Pascal 6.0 programming environment that are needed to create, edit, and execute a program. It also includes a sample programming session that acquaints the reader with the Turbo Pascal integrated development environment.

A six-step method for designing and implementing solutions to programming problems is detailed in Chapter 1—and applied throughout the text.

Chapter 2 discusses string, character, Boolean, and numeric data types; program documentation; and the fundamental components common to all Pascal programs. These concepts form the basis for a sample program that ends the chapter.

Pascal statements that create output and accept input are the essence of Chapter 3. These include how the Turbo Pascal environment can display output on a monitor screen, print it, and store it in a text file as well as how it can accept input from a keyboard and a text file. The chapter also highlights how the Turbo Pascal environment assists in finding and correcting program errors.

Chapter 4 introduces the fundamentals of processing numeric data. It also develops a strategy for testing programs that can be used as part of the software development methodology presented in Chapter 1.

Using system-defined procedures as a guide, Chapter 5 illustrates how parameters pass information between programmer-defined procedures and stresses the importance of the modular independence that can result. Chapter 6 distinguishes between Pascal procedures and functions, and shows how to create programmer-defined functions using system-defined functions as models.

Chapters 7 and 8 present, respectively, selection structures and iteration structures that, along with the sequential processing in earlier chapters, complete the repertoire of basic structures from which all programs are constructed. In addition to introducing single-option and multiple-option selection structures, Chapter 7 shows how these structures can be utilized to create menu-driven software. In Chapter 8, both definite and indefinite loop (iteration) structures are discussed and used to introduce simulation.

Array and record structures as well as their application in searching and sorting operations are detailed in Chapters 9 and 10.

Chapter 11 introduces the fundamentals of creating, reading from, adding to, and changing a data file composed of record data types.

Chapter 12, the final chapter, gives a brief introduction to advanced programming topics, including enumerated data types, pointer data types, linked lists, recursion, and object-oriented programming.

Several appendices serve as handy reference tools for both the Turbo programming environment and the Pascal language. Answers to selected exercises and programming assignments also are included.

General Features

Each chapter opens with both a list of key terms introduced in that chapter and a set of learning objectives. And each chapter ends with (1) a "Store and Forward" section that summarizes the concepts presented in that chapter and links them to the new topics presented in the next chapter, (2) a "Snares and Pitfalls" section that details errors commonly encountered by new programmers, (3) exercises to test comprehension of concepts, and (4) programming assignments covering a wide range of interests, to test the ability to apply concepts in a practical way.

An easy-to-read style enhanced by simple notational conventions regarding the use of **boldface** and *italic* type make it easy to understand concepts and work independently. The many figures included enhance the detailed explanations of programming concepts. Examples and their interpretations appear side by side, to eliminate sifting through text to find those explanations. Style tips for producing well-documented programs are presented throughout in special highlighted sections. These emphasize the importance of establishing good programming habits at an early stage.

An instructor's manual providing guidelines for effective instruction and answers for selected text exercises and assignments not found in the text is available, as is a computer disk containing the solutions to all chapter application programs and selected programming assignments.

Course of Study

Chapter 0, which introduces the integrated development environment, can be either omitted or covered independently if desired. To emphasize structure and modularization techniques, procedures are introduced early and used throughout the text. The creation and use of text files presented in Chapter 3 is optional; sections 3.6 through 3.8 can be omitted without affecting the study of topics appearing later in the text. Chapter 11 ("Data File Techniques") assumes no knowledge of text files in Chapter 3 and is self-contained. Beyond that, the book is organized so each chapter builds on concepts developed in the previous chapters. Chapter 12 ("An Introduction to Advanced Pascal Structures and Object-Oriented Programming") introduces topics likely to be found in a second course in programming and should be covered if time permits.

Acknowledgments

Many things contributed to the creation of this text, not the least of which was the understanding and moral support offered by both my family and my friends. Even though, at times, "the book" seemed to be more important than they, it was not. I would also like to thank Professor Joseph Bergin, Pace

University; Professor Roger Bielefeld, Cleveland State University; Professor George Converse, Southern Oregon State College; Associate Professor Maria C. Kolatis, County College of Morris (New Jersey); and Dr. Barbara Owens, St. Edwards University, for taking the time to suggest improvements, many of which have been incorporated; the copy editor Elliot Simon, for his valuable assistance in clarifying content; my colleagues at Mercy College, Dobbs Ferry, New York, for their encouragement; and Alan Held, Ginette Voldman, and Jim Hendricks for their assistance in providing many of the solutions. Special thanks are due to Barbara Pickard, Annette Gooch, Linda Hauck, Simmon Factor, and Brete Harrison at Cole and Associates, and David Shapiro at McGraw-Hill, all of whom helped produce the final product despite changes in publishers and five versions of Turbo Pascal. Their guidance and support were greatly appreciated.

John DiElsi

Table of Contents

Chapter 2: Components of a Pascal Program

Chapter 3: Fundamental Input and Output

Chapter 4: Processing Numeric Data and Testing Solutions

Chapter 5: Procedures

Chapter 8: Iteration Structures

Chapter 9: Arrays

Chapter 10: Records, Searching, and Sorting

Chapter 11: Data File Techniques

CHAPTER 0

The Turbo Pascal 6.0 System

Key Terms

About
bug
Close
Compile
compiler
cursor
Debug
desktop
dialog box
disk drive
DOS shell
Edit
File
hardware
hot key
input device
integrated development environment (IDE)
main memory
menu bar
monitor
New
object code
Open
operating system
Options
Output
output device
Print
processor
program
Programmer's Platform
Restore line
Run
Save
Save as
scroll
Search
secondary storage
software
source code
status line
System (≡)
TPTOUR
Trace into
User screen
window
Window

Objectives

- To introduce hardware and software terminology and fundamentals
- To present the essential components of the Turbo Pascal Programmer's Platform
- To provide hands-on experience for the programming process through a step-by-step exercise

0.1 Introduction

Mention the term *computer* and you're likely to get a wide range of responses, from awe to disgust. Some people think computers can do everything, while others think they should do nothing.

A computer is nothing more than a sophisticated tool, a collection of devices—*hardware*—that can be instructed to perform a variety of tasks. It is difficult to talk to a computer on its own terms, since all it basically understands is a series of on/off states, a combination of switches whose values are either on or off. To make the task of communicating with a computer easier and more efficient, we have to write a set of instructions—a *program*—in a high-level, or Englishlike, language and have it translated into a low-level language, one the computer understands. The translator, called a *compiler,* takes the program's instructions (*source code*) and puts them in a form the computer can execute (*object code*). Turbo Pascal is a language presented in an *integrated development environment (IDE),* called the *Programmer's Platform,* that simplifies the creation, correction, and production of programs.

This chapter introduces the fundamental concepts common to all computer systems. It summarizes the basic operation of the Turbo Pascal Programmer's Platform and provides a step-by-step guide for creating and running a sample program. More detailed explanation of the IDE can be found in the Turbo Pascal User's Guide that accompanies the software.

0.2 Hardware and Software Fundamentals

Every computer system consists of both a set of devices (the hardware) and instructions (*software*) for using that hardware. The major hardware components of a computer system include a *processor, main memory,* one or more *disk drives,* a printer, a *monitor,* and a keyboard. Figure 0.1 shows these components and how they're interrelated.

The processor performs all the calculations and makes all the decisions. It receives data and instructions from main memory and stores results there. *Secondary storage devices,* such as disk drives, store information that is not immediately required by the processor. They hold data and instructions, which are transferred to and from main memory as needed. The printer and monitor are *output devices* that receive results from main memory; the keyboard is an *input device* that enters data into main memory. Since secondary storage devices send data to and receive data from main memory, they also can be considered input and output devices.

The most powerful hardware in the world is useless without software to tell it what to do, and the most important piece of software is the operating system. An *operating system* is a collection of programs that control all the resources in a computer system, from hardware through programming

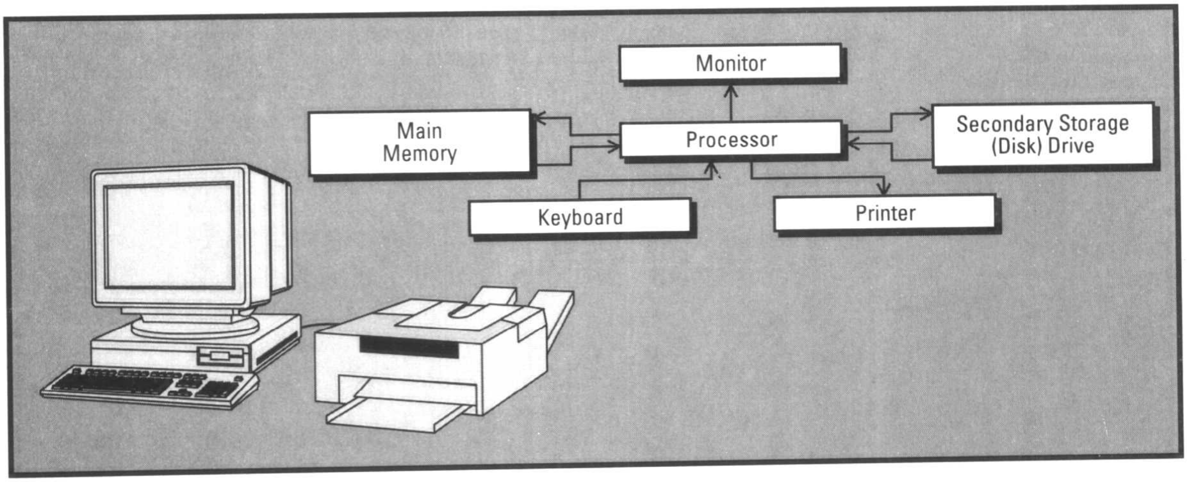


Figure 0.1
Components of a Computer System

languages and service programs. Turbo Pascal 6.0 is a programming language that was designed to work with IBM or IBM-compatible microcomputer systems with either fixed (hard) or removable (floppy) disks, using the MS-DOS or PC-DOS operating system. Although there is also a version available for Macintosh microcomputers, that version will not be discussed here.

If the Programmer's Platform is not already installed on your computer, the Turbo Pascal User's Guide provides detailed instructions.

0.3 The Turbo Pascal Programmer's Platform

The Turbo Pascal Programmer's Platform is an integrated development environment (IDE) that consists of a text editor for entering and editing program instructions, an integrated debugger to help locate errors (*bugs*) in the program, and a compiler to translate the instructions into machine-readable form. Pull-down menus, windows, and Help facilities simplify its use.

The main screen appears immediately after you load Turbo Pascal (see Figure 0.2). The main screen is divided into three sections: the menu bar at the top, the desktop (middle portion), and the status line at the bottom. The *menu bar* gives access to the commands in the IDE. If you press function key F10 to get to the menu bar, one of the ten menu choices will be highlighted. (All menu titles and menu choices in this text are displayed in **boldface** type.) To move from one menu item to another, you can either use the left- and right-arrow keys on the keyboard or press the first letter in the menu name (for example, F for **File**). If a menu item is highlighted, you can display the menu associated with that choice by pressing the down-arrow key or the Enter key. You can also display a menu in one step, by holding