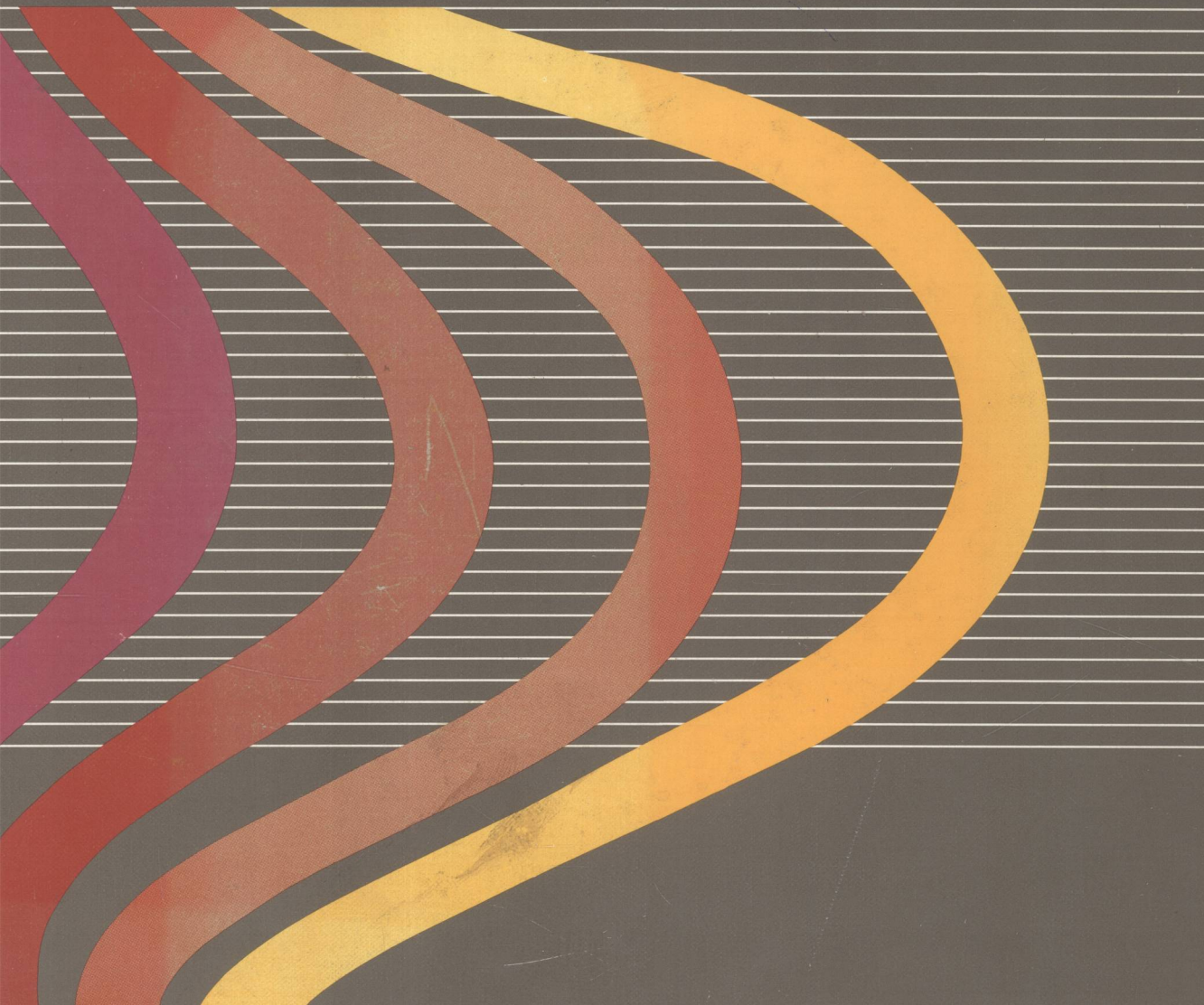
Computer Information Systems

An Introduction to Data Processing

Gerald M. Weinberg
Dennis P. Geller



Computer Information Systems

An Introduction to Data Processing

Gerald M. Weinberg

Dennis P. Geller
Babson College

Little, Brown and Company
Boston Toronto

Library of Congress Cataloging in Publication Data

Weinberg, Gerald M.
Computer information systems.

1. Electronic data processing. I. Geller, Dennis P., 1927– . II. Title.
QA76.W437 1984 001.64 84–17172
ISBN 0–316–92849–6

Library of Congress Catalog Card No. 84–17172

ISBN 0-316-92849-6

9 8 7 6 5 4 3 2 1

HAL

Published simultaneously in Canada by Little, Brown & Company (Canada) Limited
Printed in the United States of America

Photo Credits

Chapter 1: p. 2 Courtesy of Grumman Data Systems Corporation; p. 5 George Bellerose, Stock, Boston; p. 9 Ellis Herwig, Stock, Boston; p. 13 Mikki Ansin, The Picture Cube; p. 14 Dave Schaefer, The Picture Cube; p. 16 Frank Siteman, The Picture Cube; p. 17 Peter Southwick, Stock, Boston; p. 19 Eric Roth, The Picture Cube; p. 21 (top) Dave Schaefer, The Picture Cube, (bottom) The Bettmann Archive; p. 22 The Computer Museum; p. 23 The Computer Museum.

Chapter 2: p. 32 Photo courtesy of Hewlett Packard Company; p. 40 Dave Schaefer, The Picture Cube; p. 45 Gabor Demjen, Stock, Boston; p. 50 Dore Gardner, 1984; p. 60 Courtesy of Wang Laboratories, Inc.

Chapter 3: p. 68 Prestel British Telecommunications; p. 71 Photo courtesy of Hewlett Packard Company; p. 74 Frank Siteman, The Picture Cube; p. 87 (slide rule) Harry Wilks, Stock, Boston, (abacus) The Bettmann Archive, (soroban) The Smithsonian Institution, Photo #58997A, (calculator) Photo courtesy of Hewlett Packard Company; p. 100 Read D. Brugger, The Picture Cube.

Chapter 4: p. 114 Grumman Data Systems; p. 117 United Technologies, Otis Elevator; p. 119 United Technologies, Otis Elevator; p. 127 Crown Copyright, The Charles Babbage Institute, University of Minnesota.

(Continued on p. 630)

Preface

Since 'tis Nature's law to change,
Constancy alone is strange
—John Wilmot

Why This Book?

It wasn't too long ago that microcomputers were something electrical engineers bought in kit form, the only computer game was about a ping-pong ball, the Apple Computer Company had just begun, and *Time* magazine generated controversy by naming "the computer" its Man of the Year. Today these events seem buried in prehistory, almost contemporary with the fall of Rome. Yet those of us who watched these events unfold aren't twenty centuries old. To people caught up in a revolution, things often seem to move quickly.

Within your lifetime, actually in just the past few years, computers have begun to change almost every sector of our civilization. We are all caught up in the unremitting change that characterizes the revolution—but that doesn't mean we must drift aimlessly in its current. To survive in a world of change we must learn to understand the forces of change—what gives rise to them, and what their effects are. We can't predict the future, but we can prepare ourselves for it.

Many books on information systems teach survival skills for the present. This is the kind of material whose relevance is easy to understand, but which may seem hollow once the student has reached the business world. It isn't enough to understand the way things were five years ago, or even to be able to work from that knowledge to understand the way things are today. We must be able to understand not only why things are the way they are, but also where they are going.

Our approach to this book is consonant with this message. As a wider range of students perceive the role of computers in their lives and opt to take courses that explain information processing concepts, the character of those courses—once designed with only computer science or MIS majors in mind—must undergo drastic change. We designed our book to assist both students and teachers in making that change.

We have designed a pathway through the subject matter that has proved equally effective in preparing computer science majors for specialized coursework and also in orienting non-majors to the bewildering technologies that are both enhancing and invading their lives. Over the years we have taught this

material to audiences as diverse as liberal arts undergraduates, MBA students, engineers, and prospective programmers. We have discovered three threads that can be woven together in a way that is relevant to the current and future needs of most students, while covering most of what one expects to find in an introductory course on information processing in business.

The first of these threads, not surprisingly, is *change*. The current generation of students is different from the ones that preceded it. For older generations, technological change was a special event, but for today's students it is a normal process. Yet both generations are surprisingly unaware of the mechanics of change.

We have attempted to demonstrate that current technology represents but a single frame in a moving picture. Rather than tell students what the world was like when we wrote this book, we have attempted to give them an understanding of the fundamental factors that lead to technological change in areas of hardware, software, and the patterns of computer use.

The second thread is the notion of a *system*. To some, the word "system" has an unpleasant mathematical flavor, and others have used the word so loosely that it sometimes seems devoid of meaning. To us, a middle ground is a firm and necessary foundation for building the complex concepts of this course. That middle ground is best expressed by the notion of a context for understanding. A computer or program is not created to be put on a pedestal and admired. It exists in a context defined by the uses it is put to and the needs it was designed to satisfy. It must work within this context with other computers, programs, programmers, users, customers, and often with others as well. As we expand our point of view from the entity itself to include those people and things it must work with, we begin to see a system with various parts, and learn that it is the system, rather than the individual program or piece of equipment, whose behavior must be made effective and efficient.

The majority of students will be users of computer systems. To be intelligent users, they will need to understand the systems they interact with. They will need analytical tools to study the complexities of their own needs in order to see them as composed of simpler, and hence more tractable, units, working in concert. They will also need the synthetic tools to frame solutions to small problems and use them to help build solutions to larger ones. Without introducing any specific instructional material about "systems thinking," we have adopted these approaches to explaining about information systems.

The third thread is *people*—more specifically the balance between the technological and human dimensions. Information systems are for and about people. The computers may be the most dazzling components, but the people and their needs were, are, and always will be primary. Students must learn that no technological solution that ignores people can possibly succeed. For every technology and direction we study, we have attempted to point out the potential human costs and benefits. We have provided material on the "human side," such as information about career paths. Rather than have a chapter

with a title like "People and Computers," we have incorporated the human element directly into material on the technical element. We thus can avoid giving the student the impression that human aspects of information systems can somehow be wrapped up in a tidy, separate package and disposed of as soon as technology improves.

Consequently, all three threads weave throughout the book's major sections. The result, we hope, is to prepare the student for a future we can't presently describe. A student who learns how to write a simple program in BASIC is hardly prepared for the use of fifth-generation languages; one who learns how to create an ISAM file will not be prepared for secondary storage based on genetically engineered organisms. Of course, these topics are important because they describe the present. But there are more general concepts that can be applied to each of these systems. Programming languages will change, but the principles of good programming, based equally on the technologies that support them and the people that use them, will not. Rules that specify when to use an ISAM file will be irrelevant with storage media other than disks, but the fundamental trade-offs on which these rules are based will still be important.

The Plan of the Book

In the first section we examine the critical concepts upon which the subsequent material is based. We begin with a detailed examination of the concepts of data and information. We examine the attributes of data that must ultimately be considered in any real information system—cost, reliability, utility, and so on. In the context of some simple applications we show how these attributes can be identified, sometimes quantified, and always traded off against each other. We then introduce three contrasting information systems in sufficient detail for the student to develop an appreciation for system diversity. We have discovered that many students find material on information systems confusing or mystifying because they do not have a background to which they can relate the concepts. By providing three concrete examples, we build a base of experience we can refer to in later chapters.

We then proceed to examine these three very different systems by looking for common features, discovering in the process that there are certain fundamental processes shared by all information systems. Before examining each of these fundamental processes in detail, we devote a chapter to introducing the central concept of programming.

A word about our programming language is in order. It is BASIC-like because many students now have some familiarity with microcomputers using BASIC. The programs we develop might well run without modification on certain full-featured BASIC systems, but this was not our goal. We wanted, rather, a language that was suitable for expressing algorithmic ideas in a structured and readable manner, without excessive distraction by detail. Dialects of BASIC vary widely from one machine to another, and we don't wish

to encourage fruitless classroom debate over which dialect is "better." Our dialect is close enough to most real dialects that students will have no trouble moving from ours to the one for whatever machine they favor. The Instructor's Manual contains guidelines for transforming programs in our dialect to a less-featured or differently featured BASIC.

An instructor who wishes to teach programming in conjunction with this text may well wish to provide supplementary materials such as a manual for the language being used. Our language approach means that the book will cause little or no difficulty for the instructor who uses FORTRAN, COBOL, PL/I, Pascal, Ada, or any other non-BASIC language. Those instructors who do not wish to teach any more programming than the text provides will find that our material serves well to teach the principles of expressing algorithms and the skill of program reading.

In the second section, we examine in detail each of the fundamental information processes—input, processing, storage, and output. Without tying the material too tightly to any particular current technology, we cover the function of the hardware, the kinds of activities associated with the process, and the most important trade-offs in which this process participates. We also take a close look at the role of the operating system as the coordinator of these processes.

The final section addresses actual applications. We look at traditional applications such as payroll and telecommunications. We also look at applications that are still rather innovative, such as the use of artificial intelligence in business or computer assistance to the disabled. We pay particular attention to the use of computer tools in the knowledge industries, particularly office automation and programmer productivity, because this is the type of work many students will eventually do.

Through the use of a detailed case study we examine the processes of developing a business system, from its initial functional specification, through its completion and transfer to a maintenance group. We also examine the special role of software—how it can be acquired and how it should be evaluated. In these chapters the student has the opportunity to see careers in information processing in their actual context to assist them in choosing directions for further study.

Other Materials

We have provided each chapter with a full range of exercises. Some of these test the students' understanding of the basic concepts presented. Others go further, by asking them to apply these concepts in new ways or in larger contexts. The exercises provide the students with both the opportunity to exercise their new skills and the challenge to extend and evaluate them.

Some of the exercises suggest visits to real information systems, in small stores or large corporations. Realistically, we know that a group of students descending on the local supermarket might not be welcomed, but there is

valuable learning to be done outside of the classroom. If such visits were apportioned among various student teams that reported back to the class on their experiences, they would provide a wealth of additional experience and examples for the class as a whole to draw upon.

At the end of each chapter we have placed a "Last Word." While these are occasionally humorous, they all touch on important issues that space limitations prevented us from covering in depth. These Last Words can serve as the basis for lively discussions that integrate the material in the chapters with wider concerns. For example, in an excerpt that talks about the use of computers in the Watergate hearings we find out that the computer files were later shared with the chief defendants; this raises the issue of possible inequities in our legal system that may arise when only one side in a case has the funds to use computers to support its trial preparation.

We have provided an Instructor's Manual which suggests approaches to teaching the material. The Manual also contains additional material for classroom discussion or student projects, and provides answers for the exercises.

Acknowledgments

We wish to acknowledge gratefully the many people at Little, Brown who worked with us to help bring this book to fruition: Charles Durang, who brought us together; Ken Tennity and Tom Casson, who supervised the process; Tim Rosa and Audrey Handleman, who provided the essential comunications link; Victoria Keirnan and her colleagues whom we've never met, who produced a finished book from a heap of manuscript pages and scribbled notes; and, most of all, Lee Ripley, for helping us to shape our ideas into a readable book.

We also wish to acknowledge the support and suggestions from the many reviewers who examined this book during its various stages of creation: James D. Brainerd, Ferris State College; John Dineen, Middlesex College; Irv Englander, Bentley College; Robert Ferioli, Wang Laboratories; Richard Fleming, North Lake College; Fred Gruenberger, University of California—Northridge; Thomas Honeycutt, North Carolina State; Janice Lawrence, Texas A & M; Donald B. Medley, California Polytechnic Institute—Pomona; George Miller, North Seattle Community College; Steve Murtha, Tulsa Jr. College; Barbara E. Venz, Williams Woods College; and Ray Wright, Roosevelt University.

Appreciation is also due to Babson College for providing an environment in which this book could be tested.

Finally, we want to acknowledge a special debt of gratitude to Terra Ziporyn for her early critical reading and guidance. And to Judy Cook, who was reader, writer, coordinator, and motivator for the entire project from start to finish, we want to say that without you, Judy, this book would certainly never have seen the light of day. We hope that the book is sufficient thanks, because otherwise we wouldn't know how to begin.

Brief Contents

Unit One Overview 1

Chapter 1 Why Study Information Processing? 3
Chapter 2 Examples of Business Information Systems 33
Chapter 3 The Fundamental Information Processes 69

Unit Two The Framework of Information Systems 113

Chapter 4 The Foundations of Automatic Control 115
Chapter 5 Introduction to Programming 145
Chapter 6 The Structure of Programs 183

Unit Three The Hidden Technology 225

Chapter 7 Data Storage 227
Chapter 8 Operating Systems 271
Chapter 9 Processing 315

Unit Four The Creation 351

Chapter 10 The Development of An Information System: From Definition to Design 353
Chapter 11 The Development of An Information System: From Implementation to Maintenance 399
Chapter 12 The Software Spectrum 443

Unit Five The Visible Technology 489

Chapter 13 The Electronic Office 491
Chapter 14 A Tour of an MIS Department 537
Chapter 15 How New Applications Are Born 585

Contents

Unit One Overview 1

Chapter 1 Why Study Information Processing? 3

Data and information: where they come from and what they are used for; how we can establish their cost and value; how they affect our lives.

The Many Uses of Facts 4
- How Big is the Flood of Facts? 4
- Data and Information 7
- Information as Part of a Bigger Picture 9
- Information Processing 11

The Cost and Value of Information 11
- Are the Costs Worth it? 12
- Putting a Value on Information 15
- Establishing Systems of Keeping Data 17

The Changing Need for Information 19
- An Example of Changed Information Needs 20
- The Emerging Information Industry 21
- What the Information Revolution Means to You 25

Summary 26
Key Terms 27
Last Word 28
Exercises 29

Chapter 2 Examples of Business Information Systems 33

Three different types of information systems that can be found in businesses; how they operate; how they are used; what their parts are.

Paying Employees 34
- Reports 34
- Management Information Reports 39
- Analyzing Systems of Reports 41

Checking Out at the Supermarket 45
- The Receipt 46
- The Universal Product Code 47
- The Scanning System 49
- Resistance to Change 50
- Checkout System Reports 51
- Checkout Versus Payroll 53

Preparing for Court 55
- A Mountain of Paper 55
- The Litigation Support System 56
- How Smart Is the Computer? 60

Contrasting the Three Systems 61
Summary 62
Last Word 63
Key Terms 65
Exercises 65

Chapter 3 The Fundamental Information Processes 69

The functions common to all information systems; how each performs its unique tasks; how they work together.

Functions, Components, and Systems 70
Output 71
Input 71
Storage 72
Information Processing 72
The Information System Model 73
The Input Function 77
The Payroll Input Black Box 77
Arranging the Data 79
Checking the Data 80
Processing 83
Processing and Storage 84
Putting Together Instructions for Processing 85
Other Processing Operations 90
Storage 90
Methods of Retrieving Information 91
Evaluating Storage and Retrieval Costs 93
Multiple Uses of Stored Data 96
The Data Base Concept 99
Output 99
Formatted Reports 100
Handling Errors 104
Interactive Output 105
Summary 106
Key Terms 107
Last Word 108
Exercises 109

Unit Two 113
The Framework of Information Systems

Chapter 4 115
The Foundations of Automatic Control

How information systems functions are separated into hardware and software; how data are represented in the hardware; how they are interpreted by the software.

Communication and Control 116
- The Management Control Level 116
- The User Level 117
- The Operator Level 118
- The Hardware Level 118
- Communication Across Levels 119

The Central Role of Programming 120
- What's Different About "Automatic"? 121
- Decision Making 122
- The Emergence of Programming 123
- Where Is the Program? 124

The Physical Foundations 126
- What Makes It a Computer? 126
- Why Computers Use Binary-Coded Information 128

How Data Are Represented for Automatic Processing 131
- Representing Numbers 131
- Representing Text 133
- Organizing Larger Data Units 135
- Addresses 138
- Interpreting the String of Bits 138

Summary 139
Key Terms 140
Exercises 140
Last Word 141

Chapter 5 145
Introduction to Programming

A language for instructing the computer; understanding how the program statements work and how they operate on data; finding errors and making changes.

Understanding Programs 146
- Programming Languages 146
- A Sample Program 147
- The Assignment Statement: LET 147
- Data Names 149

The Flow of Control 150
- Following the Program's Execution 150
- Tracing the Sequence 151
- Modifying the Program 153
- Programming Errors 154
- Output of Results 155

Processing Diverse Data Types 156
Decimal Numbers 156
Handling Text 158
Dressing Up the Program 162
From Paper Program to Perfect Printout 164
Entering the Program Text 164
The Translator 168
Problems in Program Execution 170
When and How Errors Are Found 171
Summary 172
Last Word 173
Key Terms 176
Exercises 176

Chapter 6 The Structure of Programs 183

The basic units from which computer programs are built; how large programs can be constructed from smaller ones; input and output; the value of a programming style.

How Algorithms Are Built 184
Algorithms and Programs 184
Giving Names to Algorithms 185
Moving Data 186
Computations 187
Portraying Algorithm Structure 187
Alternative Actions 188
Repetitions 189
Summary of Algorithm Building Blocks 190
How Programs Are Built 191
Using Subroutines 191
Implementing the Subprograms 193
Communicating with the Subroutine 194
Subroutines as Standard Building Blocks 195
How a Subroutine Is Built for Reuse 196
The Function Form of Subroutine 197
How Programs Communicate with the Outside World 198
Communicating with a User 199
Moving Data to and from Files 199
How Programs Control their Own Execution 202
A Spaghetti Program 202
A Ravioli Program 203
Programming Style 204
A Lasagna Program 205
Which is the Best Control Structure/Style? 206
How Programs Repeat Themselves 206
The Endless Loop 207
Controlling the Iteration by Counting 209
Condition-Controlled Loops 212
Controls By Count and Condition 214
Summary 216
Last Word 218
Key Terms 219
Exercises 219

Unit Three 225
The Hidden Technology

Chapter 7 227
Data Storage

The fundamental trade-offs needed to understand data storage; how data bases are designed and used; the different physical storage devices.

The ABC's of Storage 228
- Accessibility of Information 228
- Bulk, or Storage Capacity 229
- Cost Considerations 232
- Trading for Reliability 233

Data Bases: The User's Dream 233
- The Need for Flexibility 234
- A System of Files 235
- A Data Model 237
- Using the Data Model 239

Data Base Languages 242
- Programmer Languages 242
- End-User Languages 243

Storage Media: The Physical Reality 247
- The Ideal Medium 248
- Magnetic Tape 248
- Punched Cards 251
- Magnetic Disks 255
- Choosing a Storage Medium 258

Managing the Organization's Data Resources 258
- The Goals of Data Resource Management 259
- Managing Consistency 259
- Managing Data Inventories 260
- Jobs in Data Resource Management 263

Summary 264
Last Word 266
Key Terms 267
Exercises 267

Chapter 8 271
Operating Systems

How the operating system manages the computer's use of its resources and its communications with external storage; different types of file structure; how the management of memory and other devices can affect the user's view of an information system.

Handling Tape Files 272
- A Typical Function 272
- Merging Two Files 272
- Operating System Services 276
- Operating System Aids to Efficient Merging 278

Direct Access Files 282
- An Order Inquiry System 282
- The Direct-File Solution 283
- The Key-to-Address Transformation 285
- Controlling Access Time 287

Indexed Sequential Files 289
- The Physical Organization of Disks 291
- Physical Access to Disk Information 292
- ISAM Data Storage 293
- Modifying the ISAM File 295

Main Memory Management 298
- Overlay Programs 299
- Virtual Memory 300
- Improving Virtual Efficiency 301

Resource Management 303
Multiprogramming 303
Time Slicing 304
Time-sharing 305
Other Operating System Functions 307
Summary 308
Last Word 310
Key Terms 311
Exercises 311

Chapter 9 Processing 315

How the choice of algorithm can affect the performance of an information system; different algorithms for sorting and searching.

Why Processing Efficiency Is Important 316
Price/Performance Economics 316
Satisfying the Growing Need for Power 317
Nonlinear Processes 320
Sorting Algorithms 323
Measuring Performance 323
Implementing and Analyzing a Selection Sort 326
Other Order n^2 Sorts 329
Faster Sorting Algorithms 330
Partition-Merge Sorting 330
Quicksort Algorithms 334
Searching Algorithms 335
Two Axioms on Efficient Searching 335
Handling Searches that Fail 337
The Search for Faster Searching 340
Summary 344
Last Word 345
Key Terms 346
Exercises 346

Unit Four The Creation 351

Chapter 10 The Development of an Information System: From Definition to Design 353

A case study of the early stages of information system development: what the needs are; what must the information system do to satisfy them; is it possible?

Overview of the System Life Cycle 355
Starting the Project 355
Design 357
Implementation 358
Phased Design and Implementation 359
Handling Change 359
Installation 360
The Mature System 360
Reevaluation 361
Requirements Definition 362
The Table of Contents 362
Level of Detail 363
The Statement of Purpose 364
The Time Horizon 364
Who Has the Answers? 365

What Are the Questions? 366
Transmitting the Report 367
Functional Specifications 372
Input Specification 372
Specifying the Processes 376
Resource Estimates 379
The User Interface 379
The Feasibility Plan 382
The Development Plan 383
Planning for Control 384
High-Level Design 385
Orchestrating the Plan 388
Tools for Planning 389
Evaluating the Plan 391
Summary 391
Last Word 392
Key Terms 394
Exercises 395

Chapter 11 The Development of an Information System: From Implementation to Maintenance 399

A case study of the later stages of information system development: how an implementation team is organized; detailed specification of the functions; building and testing the system; operating and maintaining it. Jobs in system development.

The Project Team 400
The Chief Programmer Team 400
The Chief Programmer 401
The Team Members 402
The Project Library 405
The Program Librarian 405
Detailed Design 406
The Definition Process 406
The Variety of Design Methods 408
Pseudocode 409
Flowcharting 409
Structure Charts 411
Allocating the Design Work 416
Testing the Implementation 417
Why Testing Is Necessary 418
Why Reading Is Necessary 419
Unit Testing 421
Integration Testing 422
Use Testing 423
Operation and Maintenance 424
Types of Maintenance 424
The Maintenance Challenge 425
Reviewing for Maintenance 427
The Operations Challenge 430
The Operations Review 433
The Training Challenge 434
Summary 435
Last Word 436
Key Terms 439
Exercises 439

Chapter 12 The Software Spectrum 443

Different types of software for different tasks and users, from operating systems to individual application programs; what's involved in building a software product; things to consider when choosing software.

Buying Software 444
- Sources of Software 444
- The Amateur Program 446
- The Professional Program 447
- The Program Product 449
- From Program to Product 450
- The Program System 452
- From Program System to Program System Product 453
- Custom Software 453
- How to Choose 456

Application-Independent Software 457
- Operating Systems 458
- Language Processors 463
- Utilities 465
- Programming Aids 466

Application-Dependent Software 467
- Software Packages 467
- Report Utilities 471
- Managing Data 477
- The Economics of Application Software 481

Summary 482
Last Word 484
Key Terms 485
Exercises 485

Unit Five The Visible Technology 489

Chapter 13 The Electronic Office 491

Information systems for knowledge workers; how software is changing the way we work with documents; software to support other office tasks.

Document Preparation 492
- Advantages of Computer Preparation of Documents 492
- Text Editing 494
- Creating a Routine Report 496
- Editing the Template 497
- Effects of Productivity 501

Document Maintenance 501
- Letters from Standard Parts 502
- The Importance of Format 502
- How Formatting Is Accomplished 504
- The Payoff from Automatic Formatting 505
- Spelling Checkers 507
- Smart Editors 509
- Grammar Checkers 509