

Problem Solving for Information Processing



Maureen Sprankle

PROBLEM SOLVING FOR INFORMATION PROCESSING

Maureen Sprankle
College of the Redwoods



Upper Saddle River, New Jersey
Columbus, Ohio

Library of Congress Cataloging in Publication Data

Sprankle, Maureen.

Problem solving for information processing / Maureen Sprankle.

p. cm.

Includes index.

ISBN 0-13-025599-8

1. Application software. 2. End-user computing. 3. Problem solving—Data processing.

I. Title

QA76.76.A65 S73 2002

004—dc21

2001021569

Vice President and Editor in Chief: Stephen Helba

Assistant Vice President and Publisher: Charles E. Stewart, Jr.

Assistant Editor: Delia K. Uhrec

Production Editor: Alexandrina Benedicto Wolf

Production Coordinator: Carlisle Publishers Services

Design Coordinator: Robin G. Chukes

Cover Designer: Galen Ludwick

Cover image: Kevin McTigert

Production Manager: Matthew Ottenweller

This book was set in Times by Carlisle Communications, Ltd. It was printed and bound by The Banta Company. The cover was printed by The Lehigh Press, Inc.

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada, Inc. *Toronto*

Prentice-Hall Hispanoamerican, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Prentice-Hall Singapore Pte. Ltd.

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Copyright © 2002 by Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458. All rights reserved. Printed in the United States of America. This publication is protected by Copyright and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department.

Prentice
Hall

10 9 8 7 6 5 4 3 2 1
ISBN 0-13-025599-8

*This book is dedicated to my wonderful husband, Norm, for his years
of constant love and support.*

Preface

Applications software has become more complex and more sophisticated over the years. Today, students should think in terms of how to process information to solve a problem. Problems for the end user are many times complex and require more than one type of software to achieve an efficient solution. Information processing involves problem-solving techniques and preplanning before approaching the computer. This book presents problem-solving techniques for applications including word processing, graphics, presentation graphics, desktop publishing, spreadsheets, relational database management systems, and the integration and automation of these applications.

Students will explore problem solving using applications for a project involving one or more types of software. They will study a problem and learn the best types of software to use for developing components of the solution instead of limiting the solution to just one specific type of software. When students view a problem through the use of multiple packages, they are thinking in terms of *information processing*.

This book was written for the beginning to intermediate applications students and may be used in conjunction with another text, or as a textbook in problem solving in applications or information processing classes. It is generic in nature, as no one brand of applications is presented. The book is written to help students find the best solution to a problem, and not on how to accomplish a task using a specific brand of software. If students are able to understand the concepts of various types of software, they can transfer that knowledge to any specific brand.

The text presents several types of applications concepts, as well as guidelines for problem design. It also provides specialized planning guides for problem development that follow the six steps of problem solving presented in Chapter 1. There are examples and case studies for each type of software presented in designated chapters. These are for students to follow and complete on a computer, if appropriate to the class. Problems at the end of the chapter—developed for personal and business use—allow students to develop their own solutions.

Organization

The book is divided into six units. Each is separate and may be presented in class in a different order from that presented in the book. Unit One presents information on problem solving, types of applications, installing and learning new software, and commonalities among software. Unit Two presents design concepts, graphics concepts, presentation graphics concepts, and case studies. Unit Three presents concepts and case studies for word processing and desktop publishing. Unit Four presents concepts and case studies for spreadsheets, including a chapter on mathematical concepts and functions for use with the development of spreadsheets. Unit Five presents concepts and case studies for relational database management systems. Unit Six presents concepts of documents processing and integration of types of software, programming concepts for use with applications, and concepts of automating a project. Included in the appendices are the ASCII Code, copies of the planning guides, and Otto the Robot who helps students with beginning programming concepts.

Acknowledgments

I would like to thank those who reviewed the manuscript and offered suggestions and constructive comments. In particular, I thank Stephen Dawe, West Georgia Technical Institute; John P. Russo, Wentworth Institute of Technology; Thomas A. Duer, New Hampshire Community Technological College; Gouri Banerjee, Emmanuel College, MA; David P. Beach, Indiana State University; and David Harris, College of the Redwoods, CA.

In addition, I wish to thank assistant editor Delia Uherec and production editor Alex Wolf at Prentice Hall, and Kelli Jauron at Carlisle Publishers Services for their care and patience during the writing and production process.

I would also like to thank Kevin McTigert, a College of the Redwoods multimedia student, for creating the cover image.

Maureen Sprankle

Contents

UNIT ONE	GENERAL CONCEPTS OF PROBLEM SOLVING FOR INFORMATION PROCESSING	1
CHAPTER 1	INTRODUCTION TO PROBLEM SOLVING	3
	Six Steps to Problem Solving	4
	Problem Solving in Everyday Life	5
	Problem Solving with Computers	7
CHAPTER 2	INTRODUCTION TO COMPUTER SOFTWARE	11
	Computer Tools for the End User	12
	Basic Terminology	13
	Internet Browsers	13
	Word Processing	13
	Desktop Publishing	15
	Spreadsheets	15
	Presentation Graphics	17
	Relational Database Management Systems	17
	Multimedia	18
	Languages	18
	Support Software	18
	Defining Information (Document) Processing	19
	Installing a New Software Package	19
	Learning a New Software Package	20
CHAPTER 3	COMMONALITIES AND PROBLEMS WITH SOFTWARE	23
	File Management	23
	Compatibility Issues	31
	Common Features	33

CHAPTER 4 DESIGN CONCEPTS

41

Typography	42
Fonts	43
Leading	44
Size	44
Alignment	44
Letter Width	45
Font Enhancements	46
Horizontal Spacing	46
Font Types	47
Serif versus Sans Serif	47
Font Uses	47
Guidelines for Well-Designed Documents	48
Guidelines for the General Layout of a Document	48
Guidelines for Text	50
Guidelines for Adding Graphics	51
Guidelines for Adding Color	52
General Guidelines	52

CHAPTER 5 GRAPHICS CONCEPTS AND TOOLS

57

Files Management	58
Copyright Issues	58
Clipart and Photo Clips	59
Paint and Draw Software	60
Scanned Graphics	61
Digital Photography	61
Graphics Tools	62
Pointer Tool	62
Line and Shape Tools	62
Fill Tool	63
Group and Ungroup	63
Selection Tool in a Paint Software Package	63
Eraser Tool	64
Painting Tools	64
Eye Dropper Tool	64
Character Tools	64
Stacking and Layering	64
Manipulating Graphics	65
Aligning and Distributing	65
Sizing versus Cropping	66
Graphics Use	66

CHAPTER 6 PRESENTATION GRAPHICS

69

Elements of a Slide Show	70
Basic Features of a Presentation Graphics Program	71
Background	71

	Text	72	
	Graphics	73	
	Layouts	73	
	Charts	73	
	Video and Sound	74	
	Slide Sorter	74	
	Transitions	75	
	Animation	75	
	Interactively	75	
	Design Considerations	77	
	Problems with Using an Electronic Slide Show	78	
	Slide Show Planning Guide	80	
	Example Slide Show	85	
CHAPTER 7	PRESENTATION GRAPHICS CASE STUDIES		93
	Case Study 1: Speaker's Presentation	93	
	Case Study 2: Training Session	102	
	Case Study 3: Exhibit	112	
UNIT THREE	WORD PROCESSING AND DESKTOP PUBLISHING		121
CHAPTER 8	WORD-PROCESSING CONCEPTS AND TOOLS		123
	Features of a Word Processor	124	
	Entering Text	124	
	Formatting Text	126	
	Editing Text	130	
	Printing Text	131	
	Planning a Word-Processed Document	131	
	Document Planning Guide	131	
	Common Documents	137	
	Letters on Letterhead	137	
	Memos	137	
	Reports	140	
	Templates	144	
CHAPTER 9	CASE STUDIES FOR A WORD PROCESSOR		153
	The Style Sheet	153	
	Case Study 1: Training Manual	156	
	Case Study 2: Mail Merge	162	
	Case Study 3: Report with Tables and Columns	166	
CHAPTER 10	DESKTOP PUBLISHING CONCEPTS AND TOOLS		175
	Objects	176	
	Guides	177	
	Text Threading	178	
	Master Pages	180	
	Other Desktop Publishing Tools	180	

	Desktop Publishing Planning Guide	181	
	Common Desktop Publishing Documents	186	
	Business Cards	186	
	Letterhead	190	
	Flyers	190	
	Signs	192	
	Schedule	192	
CHAPTER 11	CASE STUDIES FOR A DESKTOP PUBLISHER		205
	Style Sheet	205	
	Case Study 1: Newsletter	208	
	Case Study 2: Brochure	208	
	Case Study 3: Booklet	216	
UNIT FOUR	SPREADSHEETS		233
CHAPTER 12	MATHEMATICAL CONCEPTS AND FUNCTIONS FOR SPREADSHEETS		235
	Spreadsheet Fundamentals	236	
	Constant and Variable Data	238	
	Data Types	239	
	Numerical Data	239	
	Text Data Type	241	
	Functions	241	
	Text Functions	242	
	Statistical Functions	245	
	Mathematical Functions	245	
	Lookup Functions	247	
	Logical Functions	248	
	Financial Functions	249	
	Date and Time Functions	250	
	Database Functions	250	
	Operators	250	
	Hierarchy of Operations	251	
	Formulas	252	
	Developing Formulas	253	
	Example 1: Setting Up a Numerical Expression	253	
	Example 2: Setting Up a Relational Expression as Part of a Formula	253	
	Example 3: Setting Up a Formula that Uses More Than One Relational Expression	254	
	Example 4: Evaluating a Mathematical Formula	254	
	Example 5: Evaluating a Formula Using an IF Function	255	
	Example 6: Evaluating a Formula Created in Example 3	256	
	Example 7: Developing a Logical Expression for a Given Problem	256	

CHAPTER 13	SPREADSHEET CONCEPTS	261
	Spreadsheet Terminology	262
	Spreadsheet Features	262
	Entering Data	264
	Editing Data	267
	Analyzing Information	270
	Printing the Worksheet	270
	Charts	271
	Spreadsheet Design	273
	Spreadsheet Planning Guide	275
	Some Common Formulas	283
	Sample Problems	284
	Sample Problem 1: Calculating a Student's GPA	284
	Sample Problem 2: Amortization Chart	284
	Sample Problem 3: Worksheet Template	284
CHAPTER 14	SPREADSHEET CASE STUDIES	291
	Case Study 1: Budget	291
	Case Study 2: Costing	292
	Case Study 3: Database Functions	304
UNIT FIVE	RELATIONAL DATABASE MANAGEMENT SYSTEMS	321
CHAPTER 15	CONCEPTS OF RELATIONAL DATABASE MANAGEMENT SYSTEMS	323
	RDBMS	324
	Tables	325
	Primary, Secondary, and Foreign Keys	326
	Normalizing Tables	328
	Schema	332
	Creating Tables	332
	Queries	334
	Form Design	335
	Interface Design	335
	Reports	336
	RDBMS Planning Guide	336
	Example RDBMS	336
CHAPTER 16	RELATIONAL DATABASE MANAGEMENT SYSTEMS CASE STUDIES	357
	Case Study 1: Tree Sales	357
	Case Study 2: Home Inventory	367
	Case Study 3: Equipment Rental Control	378

UNIT SIX	PROGRAMMING FOR APPLICATIONS	393
CHAPTER 17	INTEGRATING APPLICATIONS FOR INFORMATION PROCESSING	395
	Information Processing	396
	Suites	396
	Sharing Files	397
	Object Linking and Embedding	398
	Information Processing Planning Guide	399
	Example 1: Mail Merge	402
	Example 2: Sales Report	409
	Example 3: Document Processing	409
CHAPTER 18	PROGRAMMING CONCEPTS	423
	Types of Problems	424
	Variables and Constants Revisited	424
	Instructions, Algorithms, and Flowcharts	425
	Some Common Tasks	426
	Logic Structures	426
	Sequential Logic Structure	427
	Decision Logic Structure	428
	Repetition Logic Structure	430
	Case Logic Structure	435
CHAPTER 19	EVENT-DRIVEN OBJECT-ORIENTED PROGRAMMING	441
	Interface Design	441
	Event-Driven Object-Oriented Programming	443
	Events	443
	Interactively	443
	Designing an Event-Driven Object-Oriented Project	444
	An Example	451
CHAPTER 20	CONNECTIVITY	459
	Working in Groups	459
	Working with Others around the World	460
	Telecommuting	460
APPENDIX A	ASCII CODE	463
APPENDIX B	PLANNING GUIDES	467
APPENDIX C	OTTO THE ROBOT	506
Glossary		513
Index		521

UNIT ONE

GENERAL CONCEPTS OF PROBLEM SOLVING FOR INFORMATION PROCESSING

Chapter 1: Introduction to Problem Solving

Chapter 2: Introduction to Computer Software

Chapter 3: Commonalities and Problems with Software

Chapter 1

Introduction to Problem Solving

Overview

Types of Problems

Six Steps to Problem Solving

Problem Solving in Everyday Life

Problem Solving with Computers

Objectives

When you have finished this chapter, you should be able to:

1. Describe the types of problems that can be solved by a computer.
2. List and use the six problem-solving steps to find the best solution for a problem.
3. Create and describe a solution for a problem with a given knowledge base.

Problem solving is the act of selecting and setting up a method to find the best solution for a problem. The **solution** for a problem is the processes involved to solve the problem. People make decisions every day to solve problems that affect their lives. The problems may be as unimportant as what to watch on television or as important as choosing a new profession. If they make a bad decision, then they have wasted time and resources.

It is important to know how to make the best decisions. Problem solving using the computer requires the same thought processes as for determining solutions in everyday life. This book is written to help you understand how to solve such problems on a computer. To understand how to create a solution on a computer, we will first examine a problem in everyday life—since we are most familiar with everyday problems. Then we will discuss problem solving using a computer. If you can understand how to process a solution in your own mind, then you will be able to understand how to process the same solution on a computer.

The computer is a tool to help solve a problem, whether it is writing a letter or creating a database for a company. Preplanning a solution should save time in the long run. As with any project, it is better to plan the entire process first. For example, architects create plans before they build a house or an office building, and companies plan assembly lines before they start to construct a car.

The job of an engineer is to design and plan. When you solve a problem on a computer, you become the engineer or the architect who designs and plans the solution. The same thought process and care should take place when designing and planning a document, a slide show, a spreadsheet, or a database as when building a house, designing a toy, or purchasing a car. The prospect of designing and planning a computer solution may

sometimes seem tedious, but if done right, it should save time and frustrations when finishing the product on the computer.

Six Steps to Problem Solving

There are six basic steps used in problem solving. An understanding of each of the following steps is important to find the best solution for a problem.

Identify the problem

1. *Identify the problem.* The first step in problem solving is to identify the problem. It is hard to solve a problem if you cannot properly identify it. In the classroom situation, most problems have been identified for you and given to you in the form of written assignments or problems out of the book. However, when you are doing problem solving outside of the classroom, you should identify the problem before you start to solve it. To begin, write a short paragraph, sometimes called an abstract or a problem statement, to describe the problem.

Understand the problem

2. *Understand the problem.* Before continuing toward the solution, try to understand what is involved in the problem. This includes understanding the audience and the knowledge base of the problem requirements, the developer, and the audience. If you are creating a spreadsheet on the cost of building a house, then you must first have as part of your knowledge base a working knowledge of what the cost of building a house entails. If you are setting up a solution for another person, then you must know the knowledge base of that person. Different terminology and graphics might be used depending on this knowledge base. For example, you would use a more detailed set of instructions with different terminology to tell someone how to find a restaurant in your city if he has a limited knowledge of the city than if he knows the city well. When you work with a computer, its knowledge base is very limited, very exact, and requires you to solve the problem using the knowledge base of the software that you are using. This knowledge base differs among software packages.

Identify alternative solutions

3. *Identify valid alternative methods to solve the problem.* The list of alternatives should be as complete as possible. You may want to talk to other people to find other solutions than those you have identified. Alternative solutions must be acceptable ones. For example, you could go from Denver to Los Angeles by way of New York; however, this probably would not be an acceptable solution, unless you also needed to go to New York. The more alternative solutions you have, the better your chance of selecting the best solution.

Select best solution

4. *Select the best method to solve the problem from the list of alternative solutions.* In this step you need to identify and evaluate the pros and cons of each possible solution before selecting the best one. To do this, you must first select criteria for the evaluation. These criteria will serve as the guidelines for evaluating each solution in order to find the best one.

Plan the solution

5. *Plan the solution.* Your plan should include the requirements of the problem as well as any instructions, a list of things to do, sketches, and so on. The plan should stay within the knowledge base of the individual, the nature of the audience, and the capabilities and limitations of the computer as defined in step 2. These restrictions can be very limiting especially when working with computers. Planning guides to help plan solutions to problems are included in later chapters.

6. *Evaluate the solution.* To evaluate or test a solution means to check its result to verify if it is correct, if it meets the requirements specified in the problem statement, and if it satisfies the needs of the client. A set of criteria is needed as a guide to evaluate the solution, and should be part of your plan. For example, when a person needs a piece of furniture to sleep on, buying a cot may be a solution, but it may not be satisfactory or meet the requirements of the problem statement. The criteria for evaluation might include the following: is the piece of furniture comfortable, is it within a budget, and will it fit the decor of the room? If the result of the solution for a problem is either incorrect or unsatisfactory, then the problem solver must revise the solution or start the process again.

If any one of these six steps is not completed well, the results may be less than desired.

Problem Solving in Everyday Life

People solve problems daily at home, at work, or wherever they go. Problems at home may include what to cook for dinner, what movie to see, what car to buy, or how to sell a house. At work, the problems may involve dealing with fellow employees, work policies, management, or customers. The better the decisions an employee can make, the more valuable that person is to the company. In each case, the six steps to problem solving should be followed. Most people use them without even knowing it.

For example, take the problem of what to do this evening.

1. *Identify the problem:* How do you and your friends or family want to spend the evening?
2. *Understand the problem:* Even with this problem, the knowledge base of you and your friends should be considered. Select only the solutions that everyone involved would know how to do. You probably would not select, as a possible solution, playing a game of chess if you or any of your friends do not know how to play chess or if there are more than two of you, since only two can play.
3. *Identify alternative solutions:*
 - a. watch television
 - b. invite other friends over
 - c. play games
 - d. go to the movies
 - e. rent a video
 - f. play miniature golf
 - g. go to an amusement park
 - h. go to a party

The list is complete only when you and your friends have exhausted all possible alternatives.
4. *Select the best solution:*
 - a. Select the criteria for evaluation of the solutions, such as a maximum amount of money, distance, or time involved.
 - b. Eliminate unacceptable alternatives, such as those that cost too much or do not interest one or more of the individuals involved.
 - c. Specify the pros and cons of each of the remaining alternatives.
 - d. Evaluate the pros and cons in terms of the criteria set up in step a. Select the best solution according to these criteria.