

Instructor's Manual
to Accompany

**Problem Solving and
Programming Concepts**

Ninth Edition

Maureen Sprankle, Professor Emeritus
College of the Redwoods

Jim Hubbard, HGC Technology LLC
Software Consultant

Prentice Hall
Upper Saddle River, New Jersey Columbus, Ohio

Contents

Unit 1: Introduction to Problem Solving and Programming	5
Chapter 1: General Problem Solving Concepts	7
Chapter 2: Beginning Problem Solving Concepts for the Computer	11
Chapter 3: Planning Your Solution	17
Supplemental Exercises Solutions.....	19
Unit 2: Logic Structures	21
Chapter 4: An Introduction to Programming Structure	23
Chapter 5: Problem Solving with the Sequential Logic Structure.....	25
Chapter 6: Problem Solving with Decisions	29
Chapter 7: Problem Solving with Loops	39
Unit 3: Data Structures	45
Chapter 8: Processing Arrays.....	47
Chapter 9: Sorting, Stacks, and Queues.....	53
Chapter 10: File Concepts.....	55
Chapter 11: Linked Lists.....	59
Chapter 12: Binary Trees	63
Unit 4: Database Management Systems	65
Chapter 13: Database Management Systems	67
Chapter 14: Relational Database Management Systems	69
Unit 5: Object Oriented Programming	71
Chapter 15: Concepts of Object Oriented Programming.....	73
Chapter 16: Object Oriented Program Design.....	75
Unit 6: Introduction to Game Development	77
Chapter 17: Introduction to Game Development Using OOP.....	79
Chapter 18: Introduction to Assembly Language.....	81
Unit 7: File Processing	83
Chapter 19: Sequential Access File Applications	85
Chapter 20: Sequential-Access File Updating.....	87
Appendix A: Otto the Robot	89

Unit 1: Introduction to Problem Solving and Programming

Unit 1 is designed to introduce the student to problem solving skills and basic programming concepts required for the rest of the book. These concepts include the steps used in problem solving and numerical concepts such as constants and variables, data types, operators, hierarchy of operations, expressions, and equations. Chapter 3 introduces the problem solving tools used in the book, which include

1. The Problem Analysis Chart (PAC)
2. The Interactivity Chart (Structure Chart)
3. The IPO Chart
4. The Algorithms
5. The Flowcharts

When the students have completed these three chapters, they are ready to learn about the four logic structures presented in Unit 2.

Some Teaching Suggestions

Beginning problem solving students learn problems solving skills faster and with less frustration when working with one or two other people. The old adage “two heads are better than one” really is true for these students. Working in groups about once a week helps to cement the concepts presented in class. A problem presented to the class and solved in a group allows the students to go through the thinking process with the help of other students. Students working in study groups outside of class also helps. Students who work together are, for the most part, the most successful in learning problem solving skills.

The “jigsaw” method of group learning also works very well with beginning problem solving students. This can be fun for the students, along with a change in the regular methods of learning. This method allows a student “expert” to teach a concept to a peer group. The steps to complete this exercise are:

1. Divide the lesson to be learned into categories or concepts—one category for each “Expert Group”.
2. Divide the class into as many groups as the are expert categories.
3. Give each group a category. These groups become the Expert Groups. The students learn their category as thoroughly as possible.
4. Peer groups are then formed by taking one person from each of the Expert Groups to form a Peer Group.
5. Each person in the Peer Group teaches the rest of the group about their category.
6. Each person then takes a test over the categories, and the Peer Group which has the highest average score wins.

Students differ in the way they look at designing a solution to a problem. Some have to look at the whole problem and then break the solution into parts or modules. Other students need to look at the individual types of modules before putting them together for the total solution. It is important to present a solution in both ways. One way to do this is to present the whole solution, break the whole into modules, discuss the modules, and then put the modules back together again.

Chapter 1: General Problem Solving Concepts

Students need to realize that problem solving skills are used in everyday life as well as with a computer. Since the students are familiar with everyday problems, the course should start by identifying and developing solutions to these problems using the six problem solving steps. These everyday problems deal with people since they have a broad and expandable knowledge base. Problem Two at the end of the chapter presents some problems which the student can develop a solution and then check with another student. Though the use of Otto the Robot in Appendix A, the students can relate these same problem solving steps using a limited knowledge base. Associated with the instructor's Manual you will find a simulator for OTTO the Robot. The solutions to these problems can be developed and tested in groups. This should lead to the fact that all work on a computer deals with a limited knowledge base.

Lecture Outline

- I. Introduction to class
- II. Six steps in problems solving:
 - A. Identify the problem
 - B. Understand the problem
 - C. Identify alternative ways to solve the problem
 - D. Select the best way to solve the problem
 - E. List instructions that enable you to solve the problems using the selected solution
 - F. Evaluate the solution
- III. Types of problems:
 - A. Algorithmic
 - B. Heuristic
- IV. Problem solving with other humans vs problem solving with computers
 - A. Terminology
 1. solution
 2. results
 3. program
 - B. Why do we as humans have problems with problem solving?
 - C. Use one of the tasks in problem one to illustrate problem solving with another English speaking human. The students should write the set of instructions and then test the instructions in a group situation.
 - D. Use Appendix A with Otto the Robot to narrow the vocabulary and move into problem solving with computers. Associated with the Instructor's Manual you will find a simulator for OTTO the Robot. You may want to use it to demonstrate how Otto works and/or you may want to have the students test some of the instructions and try out their solutions to see if they work. You will need to install the program on your computer and/or the student's computers.

Solutions to Questions and Problems

Questions:

1. See page 3-4 in text.
2. An Algorithmic solution is a series of direct actions taken to solve a problem.
3. Each student will have different answers. Some typical answers are:
 - a. balancing a checkbook
 - b. putting a bicycle together
 - c. baking a cake
4. A HEURISTIC solution to a problem requires knowledge, experience, trial and error.
5. Each student will have different answers. Some typical answers are:
 - a. raising a child
 - b. making money on the stock market
 - c. creating a compromise
6. Each student will have different answers. Some typical answers are:
 - a. Financial planning
 - b. Address book
 - c. Writing reports
7. See section on Problem Solving Steps.

Problems:

1. Each student's answer will be different according to the problem they selected. Correct the problem according to what should be contained in each step.
2. Each student's answer may be a modification of these solutions. The main thing to look for is the correlation between the knowledge base and the instructions. There are certain assumptions the student may make, such as the language spoken. These are excellent problems for students to develop in a group.
 - a. Make a cup of cocoa:
Knowledge Base: cup, milk, cocoa mix, spoon, microwave, how to pour liquid into a cup
Instructions:
 - 1.) open cocoa mix
 - 2.) pour cocoa mix into a cup
 - 3.) fill cup with milk to 1 inch of top
 - 4.) stir with spoon until dissolved
 - 5.) heat in microwave for 2 minutes
 - b. Sharpen a pencil.
Knowledge base: pencil, pencil sharpener, sharp point, turn a handle
Instructions:
 - 1.) go to a pencil sharpener
 - 2.) put unsharpened end of the pencil in the hole in the side or top of pencil sharpener
 - 3.) turn handle
 - 4.) check to see if the pencil has a sharp point every 5 turns
 - 5.) remove pencil and use
 - c. Walk from the classroom to the student lounge, your dorm, or the cafeteria.
This problem will have a different solution, depending upon the campus where the student attends classes.

d. Start a car.

Knowledge Base: car, key, door, seat, keyhole, telephone, mechanic,
Instructions:

- 1.) go to the car
- 2.) open the car door on the driver's side
- 3.) sit down in the driver's seat
- 4.) put the key in the starter keyhole
- 5.) turn the key until you hear the engine turning over
- 6.) if the engine starts put the car in gear and drive to your destination
- 7.) if the car does not start:
 - a.) check the gas to see if there is gas in the car.
 - b.) try again
 - c.) if the car still does not start, call a mechanic or a tow truck to tow the car to the mechanic.

e. Get a glass of water.

Knowledge base: water, glass, faucet, kitchen, way to get to kitchen, cupboard
Instructions:

- 1.) go to kitchen
- 2.) get a glass from the cupboard
- 3.) go to the sink
- 4.) turn on the faucet
- 5.) put the glass under the flow of water
- 6.) fill the glass almost to the top
- 7.) take to water and the glass back to where you were and drink the water

g. Start your computer.

Knowledge base: computer hardware, operating system,
Each person's answer will be a little different due to the differences in computers and
operating systems

3. The knowledge base will differ from student to student, however, the following is the basics:

- a. balancing your check book: bank checking account, check book, deposits, withdrawals, checks, service charges, balance, basic mathematics skills
- b. driving a car: car, starting the car, use of the gears, directions, laws and regulations of the state or country in which you are driving
- c. repairing a car: engines, mechanics tools, problem solving skills, use of manuals, car parts, cars
- d. building a house: houses, building materials, contractors, building tools, site development, laws and regulations of the area in which the house will be built
- e. calculating the cost of using your car for a month: miles driven, insurance costs, original car cost, price of gas, miles per gallon, average repair costs
- f. Play a video game: rules of the game, use of the hardware, eye-hand coordination

4. Students should develop these as a group so they can discuss what is relevant and what is not.

5. a. take the first two numbers and choose the largest remembering the number
- b. take the third number and the largest from step a and select the largest, remembering this number
- c. the result from step b is the largest.

6. Each student should develop their own reasons. This is a good problem to discuss in class.

7. See Appendix A. Each student will develop their own approach, however, all approaches should follow the six steps of problem solving.
8. See Appendix A for solutions.
9. Students will have varying answers. This is also a good problem to discuss in class.

Chapter 2: Beginning Problem-Solving Concepts for the Computer

Chapter 2 introduces computer basics which all computer students need to know and understand. Included are such concepts as constants, variables, data types, functions, operators, operator hierarchy, expressions, and equations. This is one of the most important chapters in the book, since the information is needed to use computers, whether as a programmer or as an end user.

Lecture Outline

- I. Constants vs variables
 - A. Constant data, named and unnamed
 - B. Variable names and naming conventions
 - C. Rules for naming and using variables—see page 15
 - D. Value of a variable
- II. Data types
 - A. Data vs Information
 - B. Data set
 - C. Numerical data
 - 1. integers: whole numbers
 - 2. real: all numbers
 - D. Character data: all characters
 - 1. string: string of characters
 - 2. concatenation: combining of strings
 - E. Logical data: true or false
 - F. Rules for data type usage
 - 1. use correct data type
 - 2. data types cannot be mixed
 - 3. each data type has its own data set
 - 4. all calculations use numerical data types
- III. Data Storage
- IV. Functions: (a list of functions is included in the text)
 - A. Mathematical functions
 - B. String functions
 - C. Conversion functions
 - D. Statistical functions
 - E. Utility functions
 - F. Parameters: data required by a function
- V. Operators
 - A. Operand
 - B. Resultant
 - C. Types: a chart is included in the text
 - 1. Mathematical
 - 2. Relational
 - 3. Logical: a chart is included in the text
 - D. Hierarchy: a chart is included in the text
- VI. Expressions and Equations
 - A. Assignment statement: use of the equals sign
 - B. Evaluation of expressions and equations—examples are included in the text

Solutions to Questions and Problems

Questions

1.
 - a. Allows several programmers to work on the same program without having conflicts.
 - b. Allows programs to be easily read.
 - c. Allows code to be easily maintained.
 - d. Allows increase in software productivity
2. See page 15.
3. A CONSTANT is a specific value of any data type that does not change through out the processing of the data. A VARIABLE is a value that can be changed during the processing of the data. A variable has an assigned memory location associated with the variable name.
4.

a. CompanyName	string
b. Item	string
c. InventoryNo	string
d. Quantity	numeric--integer
e. Price	numeric--real
f. Address	string
g. OrderDate	date
h. Reorder	numeric--integer
i. Obsolete	logical
5.
 - a. numeric, real--no quotes, decimals
 - b. string--quotes
 - c. logical-- no quotes
 - d. character--quotes
 - e. string--quotes
 - f. string--quotes
 - g. numeric, integer--no quotes, no decimal point
 - h. numeric, real--no quotes, scientific notation
 - i. numeric
6. A function is a small set of instructions built into a computer language or application that enables the user to complete a frequently repeated task with fewer keystrokes than would be required without the function.
7. To increase productivity
8. The / symbol represents real division that results in a number and decimal parts. The \ symbol represents modulo division that results in the whole number only--and integer-- and no decimal parts. The MOD represents the whole number remainder of a division.
 $9 / 4 = 2.25$
 $9 \setminus 4 = 2$
 $9 \text{ MOD } 4 = 1$
9.

a. 9	l. True
b. 5.0	m. 5.0
c. True	n. False
d. 2	o. True
e. True	p. 0
f. 25	q. True
g. False	r. 64.0
h. False	s. 6
i. 0	t. False
j. False	u. False
k. 10 .0	

10. See Hierarchy table (Table 2.8).

11. a. numeric / numeric
b. numeric, character or string / logical
c. logical / logical
d. logical / logical
e. numeric, character or string / logical

12. An expression is part of another instruction, which may or may not be an equation. An equation stores the value of an expression.

13. a. An EQUAL SIGN means replaced by in an equation. b. Yes, as a relational operator it means equal to.

Problems

1. a. 8.5
b. - 1.5
c. 13.5
d. 3
e. 3

2. a. $X = Y + 3 * Z - (Z + Y) / (Z - 3)$
b. $X = 5 * Y + (3 * Z - 1) / (4 * (3 * Z + 1) - Y)$
c. $X = (X - Y) ^ 2$

3. a. Could be either--depending on the problem. Since there is only one variable on the left hand side this could be either an assignment instruction or an expression to be used in another instruction.
b. Relational operator because there is an operator on the left hand side of the equals sign.

4. a. Area = Length * Width
B. Area = 2 * Height * (Length + Width)
C. Area = 2 * Height * (Length + Width) - WindowWidth1 * WindowHeight1 - WindowWidth2 * WindowHeight2 --DoorWidth * DoorHeight
D. Miles = Feet / 5280
E. Percent = (Ending - Beginning) / Beginning * 100
F. Average = (N1 + N2 + N3 + N4 + N5) / 5
G. SalesPrice = OriginalPrice * (1-Discount)

5. a. 24
b. 8
c. 10
d. 6

6. TotalMins = (NoonOut - MorningIn) + (NightOut - NoonIn)
Hours = TotalMins \ 60
Minutes = TotalMins MOD 60

7. Days = Date2 - Date1

8. PaidDate - BuyingDate < 10

9. FullTime = (Week1 + Week2) / 80 * 100

10. Average = (Test1+Test2+Test3+Test4+Test5)/5

11. Time = .5*Distance/30*60 + .5*Distance/65*60
Time = 30 * (Distance/30 +Distance /65)

- 12.a. R = False
- B. R = True
- C. R = True
- D. R = False
- E. R = True

13. a.

A	B	A OR B
T	T	T
T	F	T
F	T	T
F	F	F

A	B	NOT A OR B
T	T	F
T	F	F
F	T	F
F	F	T

c.

A	B	C	R
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F

d.

A	B	C	R
T	T	T	F
T	T	F	F
T	F	T	F
T	T	T	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	T

e.

A	B	C	R
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	T
F	F	F	T

- 14. a. R = T
- b. R = T

- 15. B = (A AND NOT C) AND D
 - A = Employed > 6 (Months)
 - B = Bonus (Logical Data)
 - C = NegativeReports (Logical Data)
 - D = Earnings > \$5000 (During The Fiscal Year)

- 16. DriversLicense And (Check <= 50 Or Card)

What's wrong with this?

- 1. a. space
 - b. should be an underline not a negative sign.
 - c. should be an underline not a division sign.
 - d. should better define the variable
 - e. space
 - f. should better define the variable
 - g. too long
-
- 2. a. 5
 - b. 2
-
- 3. a. not the same data type
 - b. cannot calculate with characters
 - c. not the same data type
 - d. not the same data type--
 - both c and d also must use logical operators with logical data types.