Increasing Your Expertise as a Problem Solver Some Roles of Computers

David Moursund

COVER

Fonts: Janson 12 pt (text) Palindrome (heads)

About the Author

Dr. David Moursund has been teaching and writing in the field of computer technology in education since 1963. He was the chairman of the Department of Computer Science at the University of Oregon from 1969 to 1975. He is now a professor in the College of Education at the University of Oregon. Dr. Moursund has authored or coauthored more than 30 books and numerous articles on computer technology in education.

In 1979, he founded the International Council for Computers in Education (ICCE), which became the International Society for Technology in Education (ISTE) in 1989 when it merged with the International Association for Computing in Education. Dr. Moursund is currently the executive officer of ISTE.

Project Coordinator Jean Hall

Editor Kathleen Friestad

Production Leigh Williams, Corinne Tan

Book Design Leigh Williams

Cover Design Nancy Fields

© International Society for Technology in Education, Second Edition, 1996
 © David Moursund, Reprinted January 2005
 Phone: 541-346-4414

Administrative Office 1787 Agate Street Eugene, Oregon 97403-1923 Phone: 541-346-4414 Fax: 541-346-5890 Index

Order Desk: 800-336-5191 Order Fax: 541-302-3778

Customer Service Office 480 Charnelton Street Eugene, Oregon 97401-2626

America Online: ISTE CompuServe: 70014,2117 Gopher: iste-gopher.uoregon.edu Internet: iste@oregon.uoregon.edu

ISBN 1-56484-095-6

Increasing Your Expertise as a Problem Solver

Some Roles of Computers

Second Edition

David Moursund

©International Society for Technology in Education, 1996

From the Publisher

The International Society for Technology in Education (ISTE) promotes appropriate uses of technology to support and improve learning, teaching, and administration. As part of that mission, ISTE's goal is to provide individuals and organizations with high-quality and timely information, materials, and services that support technology in education.

Our Books and Courseware Department works with educators to develop and produce classroom-tested materials and resources that support ISTE's mission. We look for content that emphasizes the use of technology where it can make a difference—easing the teacher's job; saving time; motivating students; helping students with various learning styles, abilities, or backgrounds; and creating learning environments that are new and unique or that would be impossible without technology.

We develop products for students, classroom teachers, lab teachers, technology coordinators, and teacher educators, as well as for parents, administrators, policy makers, and visionaries. All of us face the challenge of keeping up with new technologies and the research about their educational applications while we are learning about and implementing appropriate applications in our teaching/learning situations. Please help us in our efforts to help you by providing feedback about this book and other ISTE products and by giving us your suggestions for further development.

Jean Hall, Director, Books and Courseware Department Phone: 541-302-3775; Internet: halljea@oregon.uoregon.edu Anita Best, Editor Phone: 541-346-2400; Internet: Anita_Best@ccmail.uoregon.edu

International Society for Technology in Education Books and Courseware Department 480 Charnelton Street Eugene, OR 97401-2626



Table of Contents

Introduction	1
The Human Mind	1
The Information Age	2
Cognitive Science	4
Expertise	4
Intended Audience	6
Activities and Self-Assessment	6
Chapter 1 • Introduction to Problem Solving	9
Some Examples of Problems	10
What is a Formal Problem?	11
Representations of a Problem	13
Roman Numeral Example	15
Representing Problems Using Computers	16
Poorly Defined Problems and Problem Posing	17
Difficult and Unsolvable Problems	18
Persistence and Motivation	19
Working Toward Increased Expertise	19
Activities and Self-Assessment	20

Chapter 2 • Overview of Resources in Problem Solving
Some General Categories of Problems
Five General Categories of Resources
Creative Intelligence25
Tools
Accumulated Knowledge26
Education and Training27
Time
A Rising Level of Expectation
Computer as a Broad-Based Resource
Activities and Self-Assessment
Chapter 3 • Intelligence as Resource
A Definition of Intelligence
Howard Gardner's Theory of Multiple Intelligences
Multiple Intelligences and Computers
Robert Sternberg's Theory of Intelligence
David Perkins' Theory of Intelligence
Final Remarks
Activities and Self-Assessment
Chapter 4 • Tools as Resource
A Calculator Example41
Concept Versus Process
Some Problems With Tools44
Computer as Tool45
Building-Block Resources47
Computer-Based Building-Block Problems
Chunking Using BBRs
Education for Conceptual Understanding50
Activities and Self-Assessment52

Chapter 5 • Accumulated Knowledge as Resource	55
Brocodural and Declarative Knowledge	
Domain Specificity (Expertise in a Specific Domain) 50	
Don't Doing the Wheel	
Dollarative and Presedural Knowledge	
Studies of Chass Emerts	
Studies of Chess Experts	
A Strategy is a General Plan	
A General Strategy for Problem Solving	
Understanding of Concepts and Underlying Theory	
What is the Right Balance?68	
Computers—A New Aid to Problem Solving68	
Activities and Self-Assessment70	
Chapter 6 • Education and Training as Resource	73
Roles of Education and Training in Intelligence73	
Assuming Responsibility for Your Own Learning74	
Transfer of Learning75	
Overview of Instructional Uses of Computers78	
Computer Science and Computer Engineering78	
Computer as a Personal and Group Productivity Tool79	
Computer-Assisted Instruction (CAI)79	
More About CAI80	
Activities and Self-Assessment81	
Chapter 7 • A Computer System	83
Computer Hardware84	
Computer Software85	
Hardware and Software—A Computer System87	
Human-Machine Interface88	
Artificial Intelligence89	
Expert Systems	
Agents	
Intelligent Digital Connectivity94	

The Future of Hardware, Software, and Connectivity95 Activities and Self-Assessment96
Chapter 8 • Personal Productivity Tools
Process Writing
Generic Computer Productivity Tools102
Goals in Learning Generic Tools105
Tools Can Change a Domain106
Special-Purpose Computer Tools107
Integrated Packages and Suites108
Computer Tools Can Create a New Domain109
Activities and Self-Assessment110
Chapter 9 • Computer Programming
Types of Computer Programming111
Some History of Programming Languages113
Data Structures and Control Structures114
Procedures and Procedural Thinking115
Object-Oriented Programming116
Software Engineering116
Transfer of Computer Programming Learning118
Transfer of Problem-Solving Skills From Programming120
Expertise in Computer Programming120
Activities and Self-Assessment121
Chapter 10 • Final Remarks
Summary of Key Ideas123
Activities and Self-Assessment126
References and Resources
Index



Introduction

This book will help you to learn more about your mind and computers, and how they can work together to solve problems. The emphasis is on the types of problems where a computer is a useful aid to solving the problems.

This introductory chapter briefly mentions a number of topics that are covered in the book. Subsequent chapters delve into these topics in more detail. The annotated bibliography, in the References and Resources section at the end lists the resource materials used to prepare this book.

The Human Mind

The human mind is marvelous. It is naturally curious. It is continually learning and adapting to changing situations. It receives and processes data from our multiple senses. It poses and solves a wide range of problems. It can develop a high level of expertise in solving many different kinds of problems.

But, as we all know, the human mind and sensory system have limitations. The senses operate over limited ranges of the spectrum. For example, there are sounds that you cannot hear because of their high frequency. There are frequencies of light that you cannot see. The memory components of your brain are far from perfect. You may be certain that you know a particular fact—such as the name of a person—but be unable to recall it. You learn some new material, pass a test over it, and then later forget most of what you learned.

The processing components of your brain occasionally make errors. On repetitious tasks, the processing components can become bored, which may contribute to making errors.

The human mind is good at learning overall concepts and ideas. It is not nearly so good at memorizing and recalling small details. Consider the difference between understanding arithmetic concepts and remembering detailed processes of doing arithmetic. When and why you multiply or divide two numbers is a different topic from how you multiply or divide two numbers. It takes most people hundreds of hours of study and practice to achieve a high level of skill in paper and pencil multiplication and division. Even after this considerable learning effort, errors are frequent.

Over recorded history, people have worked to develop aids for the human mind and sensory system. Some of these aids are designed to overcome physical limitations of the mind/body. Examples include eye glasses, hearing aids, telescopes, and microscopes. In this book, we will call these types of aids *physical artifacts*.

A physical artifact may be designed to help a person achieve normal sight or hearing. But, it also may be designed to provide capabilities that are far beyond what can be achieved without the physical artifacts. For example, telescopes and microscopes extend human visual capabilities.

Other tools people have developed are designed specifically to aid the human mind. Examples include reading, writing, arithmetic, and the abacus. In this book, we will call these *cognitive artifacts*.

The abacus, navigation tables, star charts, slide rule, calculator, and computer are all cognitive artifacts. Cognitive artifacts aid in the storage, processing, and use of information.

A computerized robot or a computerized piece of science laboratory equipment can be thought of as a combination of physical and cognitive artifacts. Robots and other computerized machinery possess some of the characteristics of each of the two categories of aids to the mind and body.

Physical and cognitive artifacts contribute to the way we think about and attempt to solve problems at work, at school, and at play. You make routine use of many of these artifacts in your everyday life.

The Information Age

Our current era is often called the Information Age. During the past few decades, the buying, processing, and selling of information has been a rapidly growing business. Computers have emerged as a routine aid to storing, processing, and using information.

It is commonly asserted that "knowledge is power." Toffler (1990) analyzes a variety of forms of power, such as agricultural power, manufacturing power, military power, and knowledge as power. Toffler's book focuses on worldwide changes that are based on the growth in knowledge as a commodity that is bought and sold. A person, company, or nation that learns to make effective use of this commodity makes gains in power. That is, they gain in their ability to solve problems and accomplish tasks.

The electronic digital computer is a unifying tool of the Information Age. The computer is an aid to the collection, storage, processing, and utilization of information. The computer is a mind tool. When mind and computer work together, they can solve problems and accomplish tasks that are far beyond what can be accomplished by the mind alone. The key hardware components of a computer are illustrated in Figure 0.1. This diagram is designed to suggest that a computer system consists of four major types of hardware devices: input, storage, processing, and output. The computer user in this system may be a person; however, it may also be another computer or some other type of machine.



Figure 0.1 Computer hardware and computer user.

A computer system is a combination of hardware and software—physical machinery and computer programs. Each of the hardware components of a computer system is apt to contain/use some computer software. Some of this software is built into the hardware so that it is not readily changed by a person using the computer system. However, a generalpurpose computer system is designed to make use of a wide range of software that can be input to the storage units. Such a computer system can be thought of as a machine that can be modified easily to fit the demands of particular types of users and problems.

Cognitive Science

The study of the human mind and human intelligence has a long history. In recent years, the field of cognitive science has developed and blossomed. A cognitive scientist may be: a computer scientist who is developing computer models of how the human mind works; a neural biologist who is working to understand how a collection of neurons can learn and can solve problems; a psychologist who is developing new learning theories; or a linguist working to understand how the human mind processes language.

Progress in cognitive science contributes to the development of new and/or improved cognitive artifacts. For example, perhaps you have heard of the idea of an "advance organizer." Many textbooks begin a new topic with a brief summary of how some of the ideas in this new topic relate to things that you might already know. This summary is designed to help provide a scaffold for the new ideas. Work in cognitive science has contributed to our understanding of how people process new ideas and construct new knowledge based on what they already know.

One cognitive scientist, Howard Gardner (1983), has pioneered a theory of multiple intelligences. He argues that each person has a number of different types of intelligences. For example, people have musical intelligence, linguistic intelligence, and logical-mathematical intelligence. Through appropriate training and experience, these various intelligences can be enhanced—a person can develop his/her potential. Howard Gardner's theory of multiple intelligences is one of the central focuses in this book.

David Perkins (1995) gives a careful and detailed presentation on intelligence quotient (IQ). His book explores different definitions of intelligence and different components of intelligence. He presents convincing arguments that IQ can be increased by appropriate education and experience. A number of Perkins' ideas are integrated into this book.

Most people are good at dealing with a wide range of problems that they encounter at work, at play, and in their everyday life. However, they would like to be even better at solving problems and accomplishing tasks. They would like to increase their level of expertise in various areas.

Expertise

People can get better at whatever they do. A person can get better at a sport, at a hobby or craft, or in an academic field. A person who is really good at something relative to his/her peers is considered an expert. A person's level of expertise can increase through learning and practice.

It is important to distinguish between having some level of expertise and being an expert. The word *expertise* does not mean any particular level of ability. For anything that you can do, you can imagine a scale of performance that runs from "very low expertise" to "very high expertise." When a person has a high level of expertise in some particular area, we call this person an expert. Bereiter and Scardamalia (1993) contains an excellent summary of research about expertise.



Figure 0.2 An "expertise" scale.

In our society, a high level of expertise is valued. An expert athlete may win medals and perhaps be a highly paid professional; an expert car mechanic may prosper in the car repair business and be highly respected by customers. Similarly, an expert craftsperson may produce a product that is highly sought after. Expert writers, film producers, and so on, may reach the top of their profession and impact millions of people.

Talk shows on radio and television provide an outlet for many different experts to demonstrate their expertise. There are talk shows on such varied topics as antiques, automobile repair, business investments, medicine, and psychotherapy. Some of these shows encourage people to phone in their questions. The range of knowledge and skills of the experts who host these talk shows, or who are guests on talk shows, is impressive.

Expertise may be in a narrow domain, such as eye surgery, or it may be in a broad domain, such as parenting or teaching. A high level of expertise is based on a combination of natural talent, education and training, access to appropriate tools and information, and years of hard work.

There is a difference between expertise and specialization. A person may concentrate his/her learning efforts in a very

narrow area to become a specialist in the area. However, the person may not be very good at performing in that area—at solving the problems that that area addresses. You may think of eye surgery as being a narrow specialization in medicine. However, an eye surgeon could have a low level of expertise in the field.

In any domain, a person's level of expertise may be high or low relative to contemporary standards. The key phrase here is *contemporary standards*. A person may have a high level of expertise relative to the standards of 100 years ago, but this might constitute a low level of expertise relative to today's standards. In most domains, there is a rising tide of expectations.

In many different fields, computers can contribute to a person's level of expertise. This is because computers are such a powerful aid to solving problems and accomplishing tasks. Computers also can foster the learning experience. This book explores a variety of ways in which appropriate use of computers can help you gain in expertise in domains that interest you.

Intended Audience

This book is intended for people who are interested in:

- growing in expertise in areas of their choice;
- learning more about their own thought processes and learning processes, especially as applied to problem solving;
- learning more about capabilities and limitations of computers as an aid to solving problems, accomplishing tasks, and learning in all domains.

This is a short book, but it contains many profound ideas. You will benefit most from this book if you read it at a leisurely pace, pausing frequently to reflect over the ideas that are being discussed. Relate these ideas to things that you already know and ideas that you have thought about in the past. Think about your own thinking. Such reflective cognition and metacognition is critical to obtain the full benefits of this book. Moreover, according to Perkins (1995), careful reflection is an important contributor to increasing your IQ.

Activities and Self-Assessment

At the conclusion of each chapter of this book is a brief section labeled Activities and Self-Assessment. The basic assumption here is that you are responsible for your own learning. However, having someone raise a few carefully selected questions can help you to reflect on your own learning. Sometimes it is helpful to have some suggestions for possible activities. Here are a few activities and questions based on ideas in this introduction:

- 1. What is currently your area of greatest expertise? How does this level of expertise compare to others who you know are really good in this area? What are you doing to increase your expertise in this area? Are computers a useful tool in this area of expertise?
- 2. Select a field of interest to you. Interview one or more people who have a high level of expertise in this field. Report on what you learn. Include a discussion of how you can tell that a person has a high level of expertise in this field.
- 3. What are your personal thoughts and feelings about IQ and how it relates to gaining and/or having a high level of expertise in various types of fields? You may want to reflect on "natural talent" versus "hard work" in achieving a high level of expertise.



Introduction to Problem Solving

In your daily life you routinely encounter and solve problems. You pose problems that you need or want to solve, you make use of available resources, and then you solve the problems. Some categories of resources include: the time and effort of yourself and others; tools; information; and money.

Some of the problems that you encounter and solve are quite simple. For example, you are getting dressed and you need to tie your shoes. For most people, this is a relatively simple task. Many people find that putting together a colorcoordinated outfit is a more challenging task.

People who do research on problem solving tend to differentiate between simple problems and more complex problems. Often they focus their research on how people learn to solve and actually solve complex problems (Frensch and Funke, 1995).

The computer is a resource—a versatile tool—that can help you solve some of the problems that you encounter. A computer is a very powerful general-purpose tool. Computers can solve or help solve many types of problems. There are also many ways in which a computer can enhance the effectiveness of the time and effort that you are willing to devote to solving a problem. Thus, it will prove to be well worth the time and effort you spend to learn how to make effective use of this tool.

Some Examples of Problems

In this book we will use the terms *accomplish a task* and *solve a problem* interchangeably. We begin with a simple exercise.

Think about some of the tasks you have accomplished recently—some of the problems you have solved. Make a written list of six such problems. For example, did you eat breakfast this morning? If so, you solved a breakfast problem. Did you pay some bills? If so, you solved a bill-paying problem. Did you use a telephone to talk to a person? If so, you accomplished a distance-communication task. Did you settle an argument with a friend? If so, you solved an interpersonal problem.

Clearly, some problems are more difficult than others. Think about the problems on your list. Which were really easy? Which were more difficult? Label your problems using a three-point scale of low, medium, and high difficulty. Think about what makes one problem more difficult than another. Think about how the same problem that may be easy for you may be difficult for another person, or vice versa. Give some examples that illustrate why this is the case.

Let's carry this thinking exercise one step further. For each problem on your list, make a note of the main resources used to solve the problem. Perhaps you solved some of the problems purely by the use of your mind and body. For other problems, you may have obtained help from friends. For still other problems you may have used tools, such as a telephone, calculator, or car.

Table 1.1 contains a sample of the type of list you may have created. By doing this exercise, you have reminded yourself of three facts:

1. You routinely encounter and solve a wide range of problems.

- 2. The problems that you solve in your everyday life vary in difficulty, and the level of difficulty varies from person to person.
- 3. As you solve problems, you make use of resources such as:
 - mental and physical resources of yourself and others;
 - tools;
 - information resources such as books and videos;
 - money, to acquire other resources.

Problem	Difficu lty	Main Resources
Breakfast	Easy	Contents of refrigerator, microwave oven
Transportation	Mediu m	Bus, my feet
Communication with a friend	Easy	Telephone, my voice
Written communication with an irate customer	Hard	Stored data on the transaction, careful thinking, advice from my boss, computer
Obtain needed information about a particular company	Mediu m	Several friends, reference books
Headache	Mediu m	Meditation, aspirin

Table 1.1 Problems recently encountered and solved.

What is a Formal Problem?

There is a substantial amount of research literature on problem solving. Many textbooks and popular press books discuss problem solving (Polya, 1957; Frensch and Funke, 1995; Peters, 1994).

Researchers and writers use a somewhat common set of vocabulary as they talk about problem solving. Problem solving consists of moving from a given initial situation to a desired goal situation. A different way of saying this is that problem solving is the process of designing and carrying out a set of steps to reach a goal.



Figure 1.1 Problem-solving process—from initial situation to final goal.

In this book, we will use a formal definition of the term *problem.* You (personally) have a problem if the following four conditions are satisfied:

- 1. You have a clearly defined **given** initial situation.
- 2. You have a clearly defined **goal** (a desired end situation).
- 3. You have a clearly defined set of **resources** that may be applicable in helping you move from the given initial situation to the desired goal situation. There may be specified limitations on resources, such as rules, regulations, and guidelines for what you are allowed to do in attempting to solve a particular problem.
- 4. You have some **ownership**—you are committed to using some of your own resources, such as your knowledge, skills, and energies, to achieve the desired final goal.

These four components of a well-defined problem are summarized by the four words: givens, goal, resources, and ownership.

People often get confused by the resources part of the definition. Resources do not tell you how to solve a problem.

Resources merely tell you what you are allowed to do and/or use in solving the problem. For example, you want to create an ad campaign to increase the sales of a set of products that your company produces. The campaign is to be nationwide, to be completed in three months, and not to exceed \$40,000 in cost. All of this fits under resources. You still have to figure out how to create the ad campaign.

This definition emphasizes that you, or some entity, have a problem. Problems do not exist in the abstract. They exist only when there is ownership. The owner might be a person, or it might be an organization, or a country. However, the emphasis in this book is on problems that you, personally, may encounter and want to solve.

There are many other definitions of problem. The book edited by Frensch and Funke (1995) begins with an analysis of a number of these definitions. The analysis is done from a point of view of which definitions seem most useful to researchers doing research on problem solving. The book itself focuses on a limited range of problems, called complex problems. Complex problem solving is defined to be the type of thinking that occurs to overcome barriers between a given state and a desired goal state by means of behavioral and/or cognitive, multi-step activities. The given state, goal state, and barriers are complex and may change dynamically during the problem-solving process. This definition fits well with many real world problems, such as those faced by high-level decision makers.

Representations of a Problem

There are many different ways to represent a problem. A problem can be represented mentally (in your own mind), orally, in writing, on a computer, and so on. Each type of representation has certain advantages and disadvantages.

From a personal or ownership point of view, you first become aware of a problem situation in your mind and body. You sense or feel that something is not the way that you want it to be. You form a mental representation, a mental model, of the problem. This mental model may include images, sounds, or feelings. You can carry on a conversation with yourself—inside your head—about the problem.

Frequently, a mental model representation of a problem may suffice for solving the problem. You can think about the problem, consider alternatives inside your mind, and decide on a course of action. You may reformulate the problem, deciding on a goal that seems more appropriate. You can then use your mind/body to carry out actions to solve the problem.

For example, your stomach is generating hunger pangs and you sense that you are hungry. You have an "I am hungry" problem. You begin to give conscious thought to the problem. You remember that you ate just two hours ago. You remember that you are trying to watch your weight. You consciously integrate the "I am hungry" information with these other pieces of information. Perhaps you decide that a drink of water is the appropriate course of action.

Mental representations of problems are essential. You create and use them whenever you work on a problem. But, problems can be represented in other ways; for example, you might represent a problem with spoken words and gestures. This could be useful if you are seeking the help of another person in dealing with a problem. The spoken words and gestures are an oral and body language model of the problem. Think about your level of expertise in oral communication. Do you know some people who are particularly good at oral communication? Can you think of ways to increase your level of expertise in this area?

You might represent a problem using pencil and paper. You could do this to communicate with another person or with yourself. Writing and drawing are powerful aids to memory. You probably keep an address book or address list of the names, addresses, and phone numbers of your friends. Perhaps it contains additional information, such as e-mail addresses, birthdays, names of your friends' children, and so on. You have learned that an address book is more reliable than your memory.

There are still other ways to represent problems. For example, the language and notation of mathematics are useful for representing and solving certain types of problems. For example: A particular type of carpet costs \$17.45 per square yard—how much will the carpeting cost for two connecting rooms? One room is 16 feet by 24 feet, and the other room is 12 feet by 14 feet.



Figure 1.2 Two rooms to be carpeted.

Conceptually, the problem is not too difficult. You can form a mental model of the two rooms. Each room will be covered with carpet costing \$17.45 per square yard. So, you need to figure out how many square yards are needed for each room. Multiplying the number of square yards in a room by \$17.45 gives the cost of the carpet for the room. Add the costs for the two rooms, and you are done.

Note that this is only one of the many possible ways to conceptualize this problem. You may well think of it in a different way.

The field of mathematics has produced the formula A = LW (Area equals Length times Width). It works for all rectangular shapes. Making use of the fact that there are three feet in a yard, the computation needed to solve this problem is:

Answer = $17.45 (16/3 \times 24/3) + 17.45 (12/3 + 14/3)$

Perhaps you can carry out this computation in your head. More likely, however, you will use pencil and paper, a calculator, or a computer.

There are two key ideas here. First, some problems that people want to solve can be represented mathematically. Second, once a problem is represented as a math problem, it still remains to be solved. Over the past few thousand years, mathematicians have accumulated a great deal of knowledge about mathematics. Thus, if you can represent a problem as a math problem, you may be able to take advantage of the work that mathematicians have done before. Cognitive artifacts, such as paper-and-pencil arithmetic, calculators, and computers, may be useful.

Roman Numeral Example

Each way of representing a problem has certain advantages and certain disadvantages. A problem may be very difficult when represented one way and very easy when represented another way. Thus, a person who is skilled at representing problems in a number of different ways is likely to be a better overall problem solver than a person who has only a few ways to represent problems. An interesting example of how problems may be represented in two different ways is provided by Roman and Arabic numerals. Each is an adequate representational system for counting to a modest level. It may be easier to learn to write/count I, II, III than it is to write/count 1, 2, 3. But Arabic numerals and the positional notation system are much better for dealing with large quantities. Arabic numerals are far superior for multiplication, division, and working with fractions.

I II III IV V VI VII VIII	1 2 3 4 5 6 7 8
IX X XI XII XIII	9 10 11 12 13
MCMLXVIII divided by XXVIII	28 1,968
MMLXXII	2,072
times XXVII	x 27

Figure 1.3 Representing computational problems using Roman numerals and Arabic numerals.

This is a very important idea. Through many hundreds of years of experimentation—trial and error, careful thinking—our current base 10 Arabic numeral system was developed. Our base 10 Arabic numeral system is a cognitive artifact. When you learn it and use it, you are building on centuries of work. You did not have to reinvent this number system; it was taught to you when you were a child.

Fractions are another cognitive artifact. You may think that it is easy to determine that 1/2 + 1/6 = 2/3. However, this task was beyond the abilities of all but the most educated people 2,000 years ago. The contemporary standards for expertise in arithmetic computation have gone up considerably during the past 2,000 years! Interestingly, the advent of calculators and computers—and their increased use in schools—will likely lead to a decrease in paper-and-pencil computational skills but increased standards for correctness in doing computations. Contemporary standards change with changes in tools.

Representing Problems Using Computers

One particularly important feature of a mental model is that it is easily changed. You can "think" a change. This allows you to quickly consider a number of different alternatives, both in how you might solve a problem and in identifying what problem you really want to solve.

Other representations, such as through writing and mathematics, are useful because they are a supplement to your brain. Written representations of problems facilitate sharing with yourself and others over time and distance. However, a written model is not as easily changed as a mental model. The written word has a permanency that is desirable in some situations, but is a difficulty in others. You cannot merely "think" a change. Erasing is messy. And, if you happen to be writing with a ball-point pen, erasing is nearly impossible.

When a problem is represented with a computer, we call this a *computer model* or a computer representation of the problem. As you proceed in this book, you will explore a variety of computer models. You will see that for some problems, a computer model has some of the same characteristics as a mental model. Some computer models are easy to change and allow easy exploration of alternatives.

For example, consider a document that is represented as a word processor file. It may be easier to revise this document than a paper-and-pencil version of the document. A computer can assist in spell checking and can be used to produce a nicely formatted final product.

In the representation of problems, computers are useful in some cases and not at all useful in others. For example, a computer can easily present data in a variety of graphical formats, such as line graph, bar graph, or in the form of graphs of two- and three-dimensional mathematical functions.

But a computer may not be a good substitute for the doodling and similar types of graphical memory-mapping activities that many people use when attacking problems. Suppose that one's mental representation of a problem is in terms of analogy and metaphor. Research that delved into the inner workings of the minds of successful researchers and inventors suggests this is common and perhaps necessary. A computer may be of little use in manipulating such a mental representation.

Poorly Defined Problems and Problem Posing

Up to this point, we have used the term *problem* rather loosely. Many of the things that people call problems are actually poorly defined problem situations. In this case, one or more of the four components of a clearly defined problem are missing. For example, you turn on a television set and you view a brief news item about the homeless people in a large city and the starving children in an foreign nation. The announcer presents each news item as a major problem. But, are these really clearly defined problems?

You can ask yourself four questions:

- 1. Is there a clearly defined **given** initial situation? (Do I really know the facts?)
- 2. Is there a clearly defined **goal**? (Is it really clear to me how I would like things to be?)
- 3. Do I know what **resources** are available to me that I could use to help achieve the goal? In addition, are there rules, regulations, and guidelines that I need to know about as I work to solve this problem?
- 4. Do I have **ownership**—do I care enough to devote some of my own resources? (Am I willing to spend some time on achieving the goal?)

If you can answer "yes" to each of these questions, then you have a formal, clearly defined problem.

Often, your answer to one or more of the questions will be "no." Then, the last question is crucial. If you have ownership—if you really care about the situation—you may begin to think about it. You may decide on what you feel are appropriate statements of the givens and the goal. You may seek resources from others and make a commitment of your own resources. You may then proceed to attempt to solve the problem.

The process of creating a clearly defined problem is called *problem posing* or *problem clarification*. It usually proceeds in two phases. First, your mind/body senses or is made aware of a problem situation. You decide that the problem situation interests you—you have some ownership. Second, you begin to work on clarifying the givens, goal, and resources. Perhaps you consider alternative goals and sense which would contribute most to your ownership of the situation.

The result of the problem-posing process is a problem that is sufficiently defined so that you can begin to work on solving it. As you work on the problem, you will likely develop a still better understanding of it. You may redefine the goal and/or come to understand the goal better. You may come to understand the given initial situation better; indeed, you may decide to do some research to gain more information about it. Problem posing is an on-going process as you work to understand and solve a problem.

Problem posing is a very important idea. It is a particularly personal process, drawing on your full range of capabilities, knowledge, and interests. Often it can be hard work to convert a loosely defined problem situation into a clearly defined problem. Moreover, as you work to solve a problem, you may well decide that you want to change it into a different problem. If you are the one with ownership—if you have posed the problem—then you can modify the problem to fit your interests and needs.

Are you good at problem posing? Are you good at recognizing problem situations and converting them into clearly defined problems? What have you done during the past year to increase your level of expertise in problem posing?

Difficult and Unsolvable Problems

You know that some problems are more difficult than others. Also, you know that a particular type of problem may be quite difficult for you and quite easy for someone else. However, there is one more piece to this puzzle—some clearly defined problems cannot be solved because they have no solution. For example, you are presented with the following problem: "Find a four-letter word that contains all of the vowels." You know that this is an unsolvable problem because there are five vowels.

Here is a slightly more complex example. Suppose you want to solve the simple math problem: "Find two positive odd integers whose sum is an odd integer."

You might begin thinking about this problem by doing a little exploring. A few trials, such as 1 + 1 = 2 (even), 1 + 3 = 4 (even), 3 + 9 = 12 (even) and so on, might lead you to the conjecture that the sum of two positive odd integers is always an even integer. This could lead you to pose a new problem. The new problem would be: "Prove that the sum of two positive odd integers is an even integer." If you solve this new problem, you will have proven that the original problem has no solution.

Proving that a problem has no solution can, itself, be a very difficult task. Thus, one difficulty you face when you're working on a problem and not succeeding in solving it, is determining when to give up. You may give up because the available resources have been exhausted. You may give up because of a conviction that the problem is not solvable with the available resources. And, of course, you may give up upon becoming convinced that the problem is truly unsolvable.

The two examples used in this section are somewhat typical of textbook problems. They are trivial—they pale in significance relative to many real-world problems. Problems or problem situations, such as world peace, the homeless, the hungry, battered children, cancer, and so on, are far more difficult. Many real-world problems have the characteristic that persistent effort can contribute toward making progress on solving the problems, even though no final solution is reached.

Persistence and Motivation

Many real-world problems require a great deal of time and effort to solve. Some may not be solvable with the resources that are available. Some may take many years or many centuries to solve. Persistence is a common trait in successful problem solvers.

Your persistence in working on a problem may be determined by what motivates you. Think about intrinsic motivation and extrinsic motivation. In intrinsic motivation, your drive—your push to succeed—comes from within. You are working toward goals that you really want to accomplish. In extrinsic motivation, external factors are acting on you. They are telling you what to do and they are pressuring you to do it. The goals may be set by other people and may not be of any particular interest to you. You may be saying to yourself, "I am doing this to get a good grade. I have no interest in the problem."

Some people are able to have a great deal of persistence based on extrinsic motivation. However, the typical person is apt to have more persistence when driven by a strong intrinsic motivation. Intrinsic motivation and the ownership component in the definition of a problem are closely related.

Working Toward Increased Expertise

Think about some category of problems that you have become good at solving. Perhaps you are a really good housekeeper or a really good teacher. Perhaps you are really good at making friends and working with people. Maybe you are really good at performing music, solving math problems, or reading maps.

At some time in the past, you were just beginning to learn about these types of problems. Gradually your knowledge and skills grew. Your level of expertise in solving the problems increased.

As you look toward the future, do you intend to become still better at solving this category of problems? What are you doing to become more of an expert? Do you just leave it to chance, or are you actively and consciously engaged in increasing your level of expertise?

This book explores a number of ways to get better at problem solving. These suggestions can be applied in almost any problem-solving domain. The goal is to help you increase your level of expertise in whatever areas interest you. The assumption is that you have ownership—that you want to increase your level of expertise in various fields.

One factor in increasing expertise is obtaining appropriate feedback on what you are doing and how well you are doing it. You can provide feedback to yourself—through metacognition and reflective introspection. You can get feedback from a coach, a teacher, or a colleague. In certain types of problemsolving situations, you and a computer working together can provide you with useful feedback.

Another factor in increasing expertise is learning to make effective use of the tools that experts use. The computer is one such tool.

Activities and Self-Assessment

1. Many people benefit from keeping a journal as they work their way through a book such as this. In the journal, they reflect on ideas that occur to them as they read the book. For example, has it occurred to you that while you were in grade school, you may have spent a great deal of time learning to compete with a calculator or a computer? What are your feelings about this? As you think about problems and problem solving, do you feel adequate or inadequate? After reading Chapter 1, do you feel that you gained anything useful? If so, what? If not, why?

Start a journal. Make some entries in it each time you read a chapter or part of a chapter. From time to time you may want to go back to earlier entries and write in additional comments.

2. Think about some general problem-solving idea that you know of that is useful in lots of different settings. For example, if a problem is stressing you, perhaps you have

learned to relax, take a few deep breaths, and calm yourself. Where did you learn this idea? Could you help other people learn to do this?

- 3. Make a list of problems that you feel are very difficult to solve or are unsolvable. Make sure each is a clearly defined problem. For each, explain why you feel the problem is very difficult to solve or unsolvable.
- 4. There have been a large number of research studies on attitudes and how they affect learning, retention, problem solving, and so on. Generally, there is agreement that having a positive attitude is desirable and contributes to learning and the ability to solve problems. Discuss how these research findings relate to the ownership component of a clearly defined problem.
- 5. When you are forming a mental model of a problem, is your model mainly in terms of pictures, mainly in terms of sounds, or mainly kinesthetic? For example, do you visualize a problem in your "mind's eye"? There has been quite a bit of research on learning styles. Some people learn better visually, while others learn better orally. Do some introspection about how you think about or represent problems in your head. Think about your learning style. Write a brief report summarizing your findings.
- 6. This chapter introduces the idea that most of the things people call problems are actually ill-defined problem situations. It suggests that one key step in problem solving is to develop a clearly defined problem. Suppose you apply this idea to a number of different problemsolving areas. How will this help to increase your expertise as a problem solver?


Overview of Resources in Problem Solving

Chapter 1 provided a four-part definition of a formal problem:

- 1. givens;
- 2. goal;
- 3. resources;
- 4. ownership.

This chapter provides a general overview of resources. It lays the groundwork for specific consideration of the computer as a resource.

Some General Categories of Problems

If every problem that you encountered were completely unique and novel, problem solving would be really difficult. Fortunately, this is not the case. Many of the problems that you encounter are problems that you have seen in the past. Even many of the "new" problems that you encounter are problems that lots of other people have addressed before. The single most important idea in problem solving is building on the previous work that has been done by yourself and others. To help in this approach, problems are divided into categories. For example, there are accounting problems, music problems, personal relationship problems, and political problems. The problems in a particular category are somewhat related to each other. Thus, you can take an accounting course and learn quite a bit about a huge range of different accounting problems. You can learn the types of problems the field of accounting addresses and what aids exist to help solve these types of problems.

It is also possible to classify problems on the basis of general techniques or tools needed to solve them. For example, there are a large number of problems that fall into the database category. Examples of databases include a telephone book, a dictionary, and a personal address book. Many problems can be solved by "looking up" needed information. This has led to the development of database software. Computers are particularly useful in helping to solve database problems.

With database software and telecommunications, some problems that used to be quite complex are now quite simple. For example, an airline has hundreds of flights each day. It must sell tickets for each flight, perhaps for a year into the future. Agencies selling airline tickets are located throughout the world. This ticket sales problem is solved by use of computers, database software, and telecommunications.

Many different problems fall into a bookkeeping and accounting category. They range from simple personal budgeting to handling the payroll and fiscal planning of a multi billion-dollar enterprise. Spreadsheet software has proven to be a powerful tool in dealing with such problems.

A spreadsheet can be thought of as an automated accounting sheet that can contain both numbers and formulas. The computer can automatically apply the formulas to the numbers. For example, suppose that you have a spreadsheet model of the payroll for a company. A few changes occur, such as an increase in social security taxes and a cost of living wage increase for all workers. A few quick changes in the spreadsheet and the computer produces the new payroll. Both database and spreadsheet software are sufficiently general so that they facilitate the modeling (representation) of a wide range of problems. Database and spreadsheet representations of problems are very powerful aids to solving certain kinds of problems. If you need to solve problems that fall into these categories, it is well worth your time and effort to learn how to make use of these computer applications. We will explore these applications more in a later chapter.

Five General Categories of Resources

This book focuses on five general categories of resources: creative intelligence of people; tools; accumulated knowledge; education and training; and time. Each of the first four can be enhanced or improved. Roles of computers in enhancing these resources are discussed in the next four chapters of this book.

The fifth resource—time—is somewhat different. Your time is limited. Once time has passed, it cannot be regained or replaced. Thus, time is a resource that needs to be used carefully. There are many situations in which appropriate use of a computer can save you a lot of time. This is a unifying theme in the next four chapters.

Creative Intelligence

You are intelligent, and you are creative. That is, you can use your intelligence in novel (creative) ways to solve problems. In this book, we will use the term *creative intelligence* to represent the combination of creativity and intelligence.

It is possible to differentiate between intelligence and creativity. Thus, a person may score really well on a standard measure of intelligence (an intelligence quotient test) yet have a relatively low level of creativity. There has been a lot of research on creativity and how to increase creativity. Courses on creative problem solving are offered at many universities. Workshops on creative problem solving are readily available. They tend to be particularly popular in business and industry. Edward de Bono (1973–75, 1985, 1992) has written a number of books on this subject and is a popular workshop presenter. One of the key ideas stressed in the de Bono books is that if you are actively engaged in creative problem solving, you will get better at it. Another key idea is that creativity can be taught.

When you work to solve a problem, you bring your creative intelligence to the task. This intelligence is used: to understand the problem; to help reformulate a problem situation so it is a clearly defined problem; to modify the problem (pose a modified problem) based on information gained during the solution process; and to provide guidance in making effective use of other resources. The book (Perkins, 1995) contains an analysis of a number of ways that you can get better at such tasks.

Tools

In this book, we divide tools into two categories—physical artifacts (tools to supplement physical resources of a person) and cognitive artifacts (tools to supplement mental resources of a person). Some tools, such as computerized machinery, fall into both categories. A modern car contains a large number of microprocessors. An electron microscope can be thought of as a computerized microscope that makes use of a beam of electrons instead of light to illuminate the object being viewed.

It is evident that tools contribute to cumulative progress in helping people get better at problem solving. Once a useful tool is invented, it is relatively easy for other people to learn to use the tool. For example, you probably have made use of a microscope and a telescope. These tools have contributed greatly to a number of different fields of scientific knowledge.

We remember Alexander Graham Bell for his invention of the telephone. This invention has greatly changed the societies of the world. Nowadays, we think nothing of the fact that children learn to use a telephone even before they are old enough to go to school.

Some tools are general purpose, while others are designed to help solve a very narrow range of problems. As part of a general education, you learn to use a number of generalpurpose cognitive and physical artifacts. These become so commonplace to you that you don't even think about them. Thus, you may not even think of reading, writing, and arithmetic as cognitive artifacts—useful across every area of human intellectual endeavor.

A person who is working to develop a high level of expertise in a narrow area of specialization is apt to be learning to use some tools that are specific to that area. For example, the tools of a professional welder are quite a bit different than the tools of an eye surgeon.

Accumulated Knowledge

The amount of accumulated knowledge of the human race is huge and is continuing to grow rapidly. Various people have made estimates on how rapidly the accumulated knowledge is growing. Estimates range from a doubling every three years to a doubling every 10 years.

However, it is not clear what people mean when they say that the amount of accumulated knowledge is doubling every few years. Some people like to talk about a continuum that runs from data to information to knowledge.



Figure 2.1 The knowledge continuum.

For example, the instruments on a space ship that travels to Mars collect a great deal of *data* (often referred to as raw data) and transmit it back to Earth stations. This raw data is analyzed to give us *information* about temperature, wind velocity, size and location of hills and gullies, and so on. People analyze and synthesize this information to produce detailed *knowledge* about weather patterns on Mars and how they have shaped the planet's surface.

It is clear that the amount of accumulated data, information, and knowledge is all growing rapidly. It is increasing so rapidly that even experts in narrow specialization areas are hard pressed to remain at the forefront of their specialty areas. This information explosion contributes both to the number and nature of problems that people want to solve and to their ability to solve the problems that they pose.

There are many ways to store data, information, and knowledge. People carry a lot of it around in their heads. It may be in written form and stored on stone or clay tablets, or on paper. It may be stored on audio- and videotapes. It may be in photographs, paintings, or drawings.

And, of course, data, information, and knowledge can be stored in a computer. Computers are a key tool of the Information Age. A computer is both a storage and a processing device. It is the combination of storage and processing that make computers such powerful aids to problem solving.

Education and Training

Education and training are needed to learn to make use of tools. Some education and training is quite general, cutting across many disciplines. Other education and training is highly specific to a narrow area.

The basics of education that are emphasized in the early grades of school, such as reading, writing, arithmetic, speaking, and listening, tend to be quite general purpose. These basics of education are useful in addressing a wide range of problems. As one progresses further along in school, education and training begin to become more specialized. Through formal education, extending even to the postdoctoral level and through apprenticeship training that may take many years, one can gain a great deal of expertise in a specialized area.

The human mind is always learning. Information flows into the mind from the senses. It is processed—mostly at a subconscious level. Thus, it is appropriate to say that we are all lifelong learners. Much of this lifelong learning is informal, incidental, and nondirected. However, some of this learning is consciously directed. This consciously directed learning may occur at work, play, or school. Conscious, directed learning is essential to developing expertise and to increasing one's overall abilities as a problem solver. We will discuss this in more detail in later chapters.

Time

We have now listed four major types of resources that can be used in problem solving. Figure 2.2 shows each of these resources inside a circle. The diagram is designed to suggest the interaction of the various resources that are available to support a person doing problem solving. You might visualize yourself sitting at the top of a pyramid of resources that are available to support you in problem solving.



Figure 2.2 Four major resources in problem solving.

Finally, it is time to talk about time. The ownership component in the definition of a problem indicates that you are willing to devote some of your personal resources to accomplishing the goal. This may include putting in time thinking about the problem and time actually carrying out steps to solve the problem. It is useful to think about two different uses of time. First, there is the time spent before the problem is encountered. This time is spent in gaining general and specific knowledge and skills. It is time spent developing your mind and body. It is time spent honing an essential resource—namely, you!

Then there is the time actually devoted to solving a particular problem or accomplishing a particular task. This may be a few seconds, or it may be many years. It is important to remember that time spent solving a problem contributes to your overall knowledge, skills, and experience. It helps prepare you for the next time you encounter a somewhat similar problem.

Both the preparation time and the problem-solving time vary with the problem to be solved or task to be accomplished. However, the time needed can be decreased through using appropriate tools, education and training, access to accumulated knowledge, and so on. There are ways to save time when preparing yourself to solve problems and when actually solving problems.

A Rising Level of Expectation

In recent years, people in our society have come to expect continuing major breakthroughs in medicine, telecommunications, electronics, and other areas.

What has fueled this sustained pace of change? There is no evidence to suggest that people are inherently more creatively intelligent nowadays than they were 1,000 or 2,000 years ago. So, improvements in the resource that we have named creative intelligence do not explain what has transpired.

The next three categories of resources listed in the previous section have improved significantly over the years. More and better tools have been developed. The quantity, quality, and accessibility of stored information has improved. Education and training have responded to the changes in tools and in stored information. They have also responded to the changing needs of society and to the progress in the fields of learning and teaching. All of these changes can be thought of as people building on the previous work of other people. The net result is a rising level of expectation. The problems that people are expected to routinely solve become more complex. The expected level of expertise or performance rises. As a simple example, it took the genius of Isaac Newton and Gottfried Leibniz—two of the greatest mathematicians of their time—to invent calculus a little more than 300 years ago. Now, many students learn calculus while still in high school.

Of course, one can go back still further in time. Two thousand years ago, only a small percentage of the population could do simple arithmetic. A problem involving working with fractions could challenge a professional mathematician. Now, middle school and junior high school students routinely deal with such problems.

This rising tide of expectation is a challenge, especially to adults. You go to school, get a good education, and get a good job. Over the years, you improve in your expertise at handling the job. The question is, do you improve fast enough to keep up with the rising tide of expectation? If not, your overall level of expertise may actually be declining relative to contemporary standards.

Computer as a Broad-Based Resource

A computer is a versatile resource in problem solving. It is a resource that cuts across all academic disciplines. Moreover, it is a resource that is continuing to improve rapidly. Today's computers are a lot faster than computers of a few years ago. The cost-effectiveness of computers is growing rapidly. More and better software is being developed. The human-machine interface is being improved, making it easier for people to use computers. The term *human-machine interface* refers to the design and implementation of the way humans input information to the machine and the way humans receive information from the machine. The development of the mouse as a pointing and selecting device was a major improvement in the human-machine interface.

The first four main categories of problem-related resources listed in this chapter are each affected by computers. The next four chapters of this book explore these four categories of resources from a computer-oriented point of view. One of the underlying themes is time. Appropriate use of computers may save time as you prepare yourself to solve problems and may save time in the actual process of solving problems.

Activities and Self-Assessment

Have you been writing in your journal while working through the material in this chapter? Some of the following questions may help guide your journal-writing activities.

- 1. Do some introspection about your creative intelligence. What are some of your greatest strengths? What are some of your relative weaknesses? Have your strengths and weaknesses changed over time? How have your relative strengths and weaknesses affected you at work, at play, and at school?
- 2. Name some of the tools that you routinely use in solving the problems that you encounter. Analyze these tools from several points of view. For example, you might compare them in terms of how useful they are, how long it took you to learn to use the tools, their cost, their versatility, and so on.
- 3. Humans have accumulated a great deal of knowledge. Much of this knowledge is stored in libraries. Many of the problems that people encounter have already been successfully solved by other people. Discuss the capabilities and limitations of information retrieval—for example, accessing the accumulated knowledge in libraries—as an aid to problem solving. How is this being affected by the rapid trend toward computerization of libraries?
- 4. Analyze the quality of education that you have received in the past and are currently receiving. What are its strengths and weaknesses in terms of preparing you to effectively deal with the types of problems that people face in our Information Age society? Have you used your school time efficiently and effectively?

5. The business world is familiar with the idea of "just in time" delivery of materials to a manufacturing site. The idea is to avoid unnecessary costs of warehousing a large supply of parts. The idea of "just in time" education is new and interesting. It leads to a discussion of what you should learn well in advance of attempting to solve certain categories of problems, and what you should learn "just in time" when you encounter the problems. Discuss "just in time" education.



Intelligence as Resource

One resource that every person has is their own intelligence. Actually, it is much more accurate to speak of one's intelligences. Each person has varying levels of intelligence in different areas. For example, a person may have a high level of musical intelligence but a relatively low level of logical/mathematical intelligence. Howard Gardner argues that each person has at least seven distinct intelligences.

Robert Sternberg and David Perkins have both made significant contributions to our understanding of intelligence. Some of their ideas are covered in this chapter.

This chapter gives a definition of intelligence and then explores a variety of intelligences that people have. Many of the key ideas that are covered come from the work of Howard Gardner. After an initial exploration, the intelligences are analyzed from a computer-as-tool point of view.

A Definition of Intelligence

The study and measurement of intelligence have long histories. For example, Alfred Benet and Theodore Simon developed the first Intelligence Quotient (IQ) test in the early 1900s. Chances are, you have taken several IQ tests. Also, the chances are that you are not really sure what IQ is. It turns out that IQ is a complex concept. Researchers in this field argue with each other. There is no clear agreement as to what constitutes IQ or how to measure it. Three researchers who have written widely sold books about intelligence are Howard Gardner (1983, 1993), Robert Sternberg (1988), and David Perkins (1995). And, of course, there is an extensive and continually growing collection of research papers on the topic.

The following definition is a composite from various authors. It is designed to fit the needs of this book.

Intelligence is a combination of the ability to

- 1. **learn.** This includes all kinds of informal and formal learning via any combination of experience, education, and training.
- 2. **pose problems.** This includes recognizing problem situations and transforming them into more clearly defined problems.
- 3. **solve problems.** This includes solving problems, accomplishing tasks, and fashioning products.

Donald Norman is a cognitive scientist who has written extensively in the area of human-machine interfaces. His book (Norman, 1993) begins with a discussion of how tools (physical and cognitive artifacts) make us smart. In many areas, a person with appropriate training, experience, and tools can far outperform a person who lacks these aids.

The definition of intelligence used in this book is a very optimistic one. It says that each of us can become more intelligent. We can become more intelligent through study and practice, through access to appropriate tools, and through learning to make effective use of these tools (Perkins, 1995).

Howard Gardner's Theory of Multiple Intelligences

Many people think of intelligence as a number that results from taking an IQ test. But, researchers in the field of intelligence have long realized that people have a variety of different intelligences. A person may be good at learning languages and terrible at learning music—or vice versa. A single number cannot adequately represent the complex and diverse capabilities of a human being.

There is a substantial amount of literature as to what are the most important components of intelligence, how to measure them, and how to foster their development. The latter point may be the most important. According to the definition of intelligence used in this book, intelligence can be fostered—it can be enhanced. Thus, it is important to have a theory of intelligence that identifies components that can be fostered.

Howard Gardner has proposed a theory of multiple intelligences. He has identified seven components of intelligence (Gardner, 1983). He argues that these intelligences are relatively distinct from each other and that each person has some level of each of these seven intelligences. He admits to the possibility that there are more than seven intelligences. However, he has continued to support and believe in his initial list of seven for more than 10 years.

The following table lists seven intelligences identified by Howard Gardner. It provides some examples of the types of people who exhibit a high level of an intelligence. The seven intelligences are listed in alphabetical order.

Intelligen ce	Examples	Discussion
Bodily- kinesthetic	Dancers, athletes, surgeons, craftspeople	The ability to use one's physical body well.
Interperso nal	Sales people, teachers, clinicians, politicians, religious leaders	The ability to sense other's feelings and be in tune with others.
Intraperso nal	People who have good insight into themselves and make effective use of their other intelligences	Self-awareness. The ability to know your own body and mind.
Linguistic	Poets, writers, orators, communicators	The ability to communicate well, perhaps both orally and in writing, perhaps in several languages.
Logical- mathemati cal	Mathematicians, logicians	The ability to learn higher mathematics. The ability to handle complex logical arguments.
Musical	Musicians, composers	The ability to learn, perform, and compose music.
Spatial	Sailors navigating without modern navigational aids, surgeons, sculptors, painters	The ability to know where you are relative to fixed locations. The ability to accomplish tasks requiring three-dimensional visualization and placement of your hands or other parts of your body.

Table 3.1 Examples for each of the seven intelligences.

Do some introspection. For each of the seven intelligences in the Howard Gardner list, think about your own level of talents and performance. For each intelligence, decide if you have an area of expertise that makes substantial use of the intelligence. For example, perhaps you are good at music. If so, is music the basis of your vocation?

You will probably see that you have a reasonable level of ability to perform in each of the seven areas. Moreover, it is likely you will see that you have various levels of ability to perform in different areas. Finally, you may realize that you have devoted far more time and energy to developing some of your potentials than you have the others.

Multiple Intelligences and Computers

Intelligence	Computer-Related Ideas
Bodily-kinesthetic	Games involving eye-hand coordination and quick reflexes. Keyboarding skills. Rapid, accurate mousing. Carpel-tunnel syndrome.
Interpersonal	Teams of people working together, facilitated by groupware software. Networking with people located throughout the world. Role- playing games.
Intrapersonal	Metacognition during learning to make effective use of computer hardware and software. Understanding of one's other intelligences, learning strengths and weaknesses, transfer of learning strengths and weaknesses with respect to computer-related technology. Role-playing games.
Linguistic	Computer-based tools to aid in composition. New linguistic forms, such as hypermedia. Design as a key part of computer-based communication and desktop publishing. Computer programming languages.
Logical- mathematical	Computer programming. The representation and solution of logical and mathematical problems by means of a computer. Spreadsheets and databases.
Musical	Computer as music-writing instrument. Computer as music performer. Sound as an aid to communication in hypermedia applications.
Spatial	Paint, draw, and computer-assisted design/computer-assisted manufacturing (CAD/CAM) tools. Page-layout software. Dungeons and dragons types of games. Virtual realities.

Computers affect each of the components of intelligence in the Howard Gardner list. Table 3.2 shows a brief computeroriented analysis for each component. Table 3.2 How computers affect the seven intelligences.

The general idea suggested by Table 3.2 is that the computer is a useful performance-aid tool in each of Howard Gardner's seven areas of intelligence.

Of course, the computer can also be used as an aid to learning. That is, computer systems can be developed to help a person get better at using each of the seven intelligences given in the Howard Gardner list. Computer-assisted learning (computer-assisted instruction) is discussed in Chapter 6.

Robert Sternberg's Theory of Intelligence

As noted earlier in this chapter, different researchers have identified different components of intelligence. Sternberg (1988), for example, focuses on just three main components:

- 1. Practical intelligence—the ability to do well in informal and formal educational settings; adapting to and shaping one's environment; street smarts.
- 2. Experiential intelligence—the ability to deal with novel situations; the ability to effectively automate ways of dealing with novel situations so they are easily handled in the future; the ability to think in novel ways.
- 3. Componential intelligence—the ability to process information effectively. Includes metacognitive, executive, performance, and knowledge-acquisition components that help to steer cognitive processes.

Sternberg provides examples of people who are quite talented in one of these areas but not so talented in the other two. In that sense, his approach to the field of intelligence is somewhat like Howard Gardner's. However, you can see that Sternberg does not focus on specific components of intelligence that are aligned with various academic disciplines.

Sternberg strongly believes that intelligence can be increased by study and practice. Quite a bit of his research focuses on such endeavors.

David Perkins' Theory of Intelligence

Perkins (1995) examines a large number of research studies both on the measurement of IQ and of programs of study designed to increase IQ. He presents detailed arguments that IQ has three major components or dimensions.

- 1. Neural intelligence. This refers to the efficiency and precision of one's neurological system.
- 2. Experiential intelligence. This refers to one's accumulated knowledge and experience in different areas. It can be thought of as the accumulation of all of one's expertises.
- 3. Reflective intelligence. This refers to one's broad-based strategies for attacking problems, for learning, and for approaching intellectually challenging tasks. It includes attitudes that support persistence, systemization, and imagination. It includes self-monitoring and self-management.

There is substantial evidence to support the belief that a child's neural intelligence can be adversely affected by the mother's use of drugs such as alcohol and cocaine during pregnancy. Lead (such as from lead-based paint) can do severe neural damage to a person. Vitamins, or the lack thereof, can affect neural intelligence.

Moreover, there is general agreement that neural intelligence has a "use it or lose it" characteristic. It is clear that neural intelligence can be maintained and, indeed, increased, by use.

Experiential intelligence is based on years and years of accumulating knowledge and experience in both informal and formal learning environments. Such knowledge and experience can lead to a high level of expertise in one or more fields. People who live in "rich" learning environments have a significant intelligence advantage over people who grow up in less stimulating environments. Experiential intelligence can be increased by such environments.

Reflexive intelligence can be thought of as a control system that helps to make effective use of neural intelligence and experiential intelligence. A person can learn strategies that help to make more effective use of neural intelligence and experiential intelligence. The habits of mind included under reflexive intelligence can be learned and improved.

Previous chapters in this book have included an emphasis on metacognition and on reflexivity. Many of the journaling activities have focused on reflecting over learning, education, problem solving across different disciplines, and so on. All of these types of activities are important to increasing reflexive intelligence.

Final Remarks

This chapter lists a number of possible components of intelligence. Some of the ways of dividing up intelligence may seem more intuitively correct to you than others. It seems clear that even the experts do not agree with each other. An analysis of seven different theories of intelligence is given in Chapter 12 of Perkins (1995).

There has been a great deal of research about intelligence. The works of Gardner, Perkins, and Sternberg provide summaries of varied approaches to understanding intelligence. Shekerjian (1990) contains a number of case studies of people who have displayed high levels of intelligence. The literature provides substantial evidence that through study and practice, people can make appreciable gains in their ability to solve problems and accomplish tasks that require a high level of intellect.

Each of us is free to look at intelligence from our own point of view. You may want to argue that certain intelligences should be removed from the Howard Gardner list or that there are other candidates that should be added. You may like the Sternberg list better than the Gardner list. You may want to create your own list. You may agree with David Perkins that intelligence can be increased, or you may find that his arguments are not convincing to you.

Actually, the details given in the various definitions and analyses of components of intelligence may not be particularly important to you. Perhaps what is more important is that the various definitions can help as you work to understand your relative strengths and weaknesses. You can come to understand yourself better.

Also, these definitions can provide a basis for examining how computers may enhance your overall ability to solve problems. There is no doubt that through study, practice, and learning to make use of tools such as computers, you can get better at problem solving.

Activities and Self-Assessment

- 1. Analyze your personal strengths and weaknesses from the point of view of the seven intelligences identified by Howard Gardner. For each of these seven intelligences, analyze what you believe to be your "native" levels of intelligence and also your "developed" levels of intelligence.
- 2. Select a job situation, perhaps your current job or a career that you are planning. Analyze it from the point of view of which of Gardner's seven intelligences seems most important. Then select a second job situation or possible career that is substantially different from the first and repeat the analysis.
- 3. Do activities 1 and 2 from the point of view of the three components of intelligence proposed by Robert Sternberg. From your point of view, which of the two approaches (Gardner or Sternberg) seems to be most useful to you? Then repeat this activity for Perkins' three components of intelligence.
- 4. Add one or more types of intelligence to Howard Gardner's list of intelligences, to Robert Sternberg's list of components of intelligence, or to David Perkins' list of components of intelligence. Give arguments that support your additions. Analyze yourself from the point of view of the intelligences that you add to the list.
- 5. The definition of intelligence used in this chapter is different from a definition that says intelligence is something that you are born with—that your level of

intelligence is fixed at birth. Discuss "nature versus nurture and tools" issues in intelligence.

6. Select two or three intelligences from the Howard Gardner list that are most helpful in doing well in our current K–12 school system. Are these also the ones that are most helpful in doing well on the job or in other non-school settings?



Tools as Resource

A superbly trained runner can run a mile in under four minutes. With the aid of a bicycle, many people can traverse a mile in much less time than that. With a car, still less time is required. If the problem is to get from Point A to Point B rapidly, then running, using a bicycle, using a car, or using an airplane might all be alternatives to consider. The most appropriate tool will depend on a number of different conditions, including the distance between Point A and Point B.

Donald Norman (1993) argues that much of the "smartness" and overall performance capabilities of people is due to the tools that humans have invented. Humans have developed aids to their physical capabilities (physical artifacts) and aids to their mental capabilities (cognitive artifacts). The knowledge and skills to use these tools greatly enhances one's abilities to solve problems and create products.

This chapter discusses tools as a resource in problem solving. The main focus is on computer-as-tool.

A Calculator Example

The history of the abacus goes back at least 5,000 years. The abacus is such a good aid for doing arithmetic that its use is still taught in some countries. Moreover, the abacus provides a good model for learning and understanding number systems and arithmetic. Thus, bead frames are often used as a learning aid in elementary school classrooms.

The handheld electronic calculator is a relatively new aid to computation. A calculator can be thought of as a limitedpurpose computer. By 1980, handheld calculators were inexpensive enough to come into widespread use by the general public. The National Council of Teachers of Mathematics recommended their routine use in schools. However, parents, teachers, and the overall educational system have resisted this change in our educational system.

This section gives some insights into the nature of the arguments for and against the introduction of calculators into schools. The goal is to help you gain increased insight into both roles and controversy of making use of technology in education.

Since handheld calculators have become cheaply and readily available, people have argued about whether students should be allowed to use them. There is widespread agreement that students need to understand the concepts underlying the use of arithmetic to represent and solve problems. Thus, the argument is on what tools students should be allowed to use as they actually carry out the processes involved in arithmetic computation. Should students learn to do paper-and-pencil arithmetic before they are allowed to use calculators?

The process of paper-and-pencil multiplication or long division is rather complex, relative to the cognitive abilities (the innate, nature-provided mathematical/logical intelligence) that most people have. Our education system provides students with several hundred hours of instruction and practice in multiplication and long division. Still, many students achieve only a modest level of speed and accuracy at these tasks. That is, it takes a great deal of time and effort to develop a minimal amount of speed and accuracy at doing paper-and-pencil arithmetic operations.

That is not the case when a calculator (rather than paper and pencil) is the tool. It takes only a short amount of time to learn to use a calculator. In many cases, a calculator is much faster than paper-and-pencil calculation. And, for many students, a calculator is more accurate than paper-and-pencil calculation.

The previous paragraph contains three key ideas. The first two ideas have to do with time. First, there is the time needed to learn to use a tool to do arithmetic. It takes far more time to learn to do paper-and-pencil calculations than it does to learn to do calculator-assisted calculations. Second, there is the time needed to actually carry out calculations. In many cases, a calculator is faster than paper and pencil.

The third key idea is accuracy. The human mind is not particularly good at carrying out detailed processes that require great accuracy. We may set the passing mark at 75% on a paper-and-pencil computation test. We would surely set a much higher passing make if the student is using a calculator. That is, the calculator has changed contemporary standards for accuracy in doing arithmetic.

Concept Versus Process

In doing arithmetic, there is a substantial difference between the concept and the process. When do you divide two numbers? What is accomplished by dividing two numbers? What kinds of problems are solved by division? These are questions about the concept of division.

Notice that these concept questions are independent of the process being used to do the division. The process might be carried out mentally, using pencil and paper, using a slide rule, or using a calculator.

By and large, the human mind is much better at learning concepts than it is at learning processes and then developing speed and accuracy in carrying out the processes. That is one reason why people have developed tools, such as the calculator and computer, to help automate the carrying out of certain types of processes.

In any problem-solving situation, you are apt to have to deal with concepts and processes. At the concept level, you work to understand the problem and to pose a clearly defined problem. You work to represent the problem in a manner that may help in solving the problem. In conceptualizing and representing the problem, you make use of cognitive artifacts—for example, counting numbers and precise vocabulary. After you have appropriately understood and represented the problem, you decide on a course of action to follow in solving the problem. You may make use of tools as an aid to carrying out the processes needed to solve the problem.

One of the reasons that computers are so important in problem solving is that they are so fast and accurate at carrying out processes. This is causing educators to rethink the curriculum content in many areas. Math education leaders, for example, are now placing much more emphasis on teaching for conceptual understanding. The increased emphasis is on the higher-order thinking skills. Of course, instruction time also needs to be given to learning to use the tools effectively to carry out the processes.

We need not restrict ourselves to examples from mathematics. Consider using a word processor and spelling checker when doing writing. The work processor facilitates repeated revision of one's work. The spelling checker helps to decrease the number of misspelled words. The computer printer produces output that, typically, is far easier to read than the handwriting that most people can produce. These computer tools allow increased focus on the content of the writing.

Some Problems With Tools

The previous section discussed handheld calculators. Many people argue that students should master paper-and-pencil arithmetic before they are allowed to use calculators. Many others argue that there is little value in mastering paper-andpencil arithmetic and that students should be taught to use calculators. Such arguments can become quite heated!

Notice that in either case, there is an implicit assumption that it is appropriate to have students learn the cognitive artifact we call arithmetic. People for and against the use of calculators agree that students should learn to use arithmetic to represent and solve problems. People for and against the use of calculators agree that students need to understand the underlying concepts in math problems. The argument is about what cognitive artifacts should students use to carry out arithmetic computations.

This type of argument raises a number of important issues that need to be examined whenever one is deciding on what tools to learn to use.

- 1. Learning underlying concepts: To what extent does learning to use the tool contribute to learning underlying concepts about the types of problems with which the tool will be employed? To what extent does learning the underlying concepts contribute to increasing expertise in problem solving?
- 2. Time and other resources required to learn to make effective use of the tool: Is this an efficient use of the time and other resources of the learner and the teacher?
- 3. Contemporary standards of performance: Will use of the tool help in meeting contemporary standards of achievement and performance in solving problems using the tool? Is use of the tool appropriate or necessary to having a desired level of expertise?
- 4. Availability: Will the tool be available when it is needed?
- 5. Cost-effectiveness: Taking into consideration all costs and the desired levels of performance, is the tool costeffective relative to alternative tools?

These are not easy questions. Try answering them for the issue of learning paper-and-pencil arithmetic versus learning to use a handheld calculator. Keep in mind that a solar battery-powered handheld calculator now costs about \$4.

The same questions can be applied to other tools, such as a word processor with spelling checker and laser printer. This is a relatively expensive tool, and often it is not readily available. It takes quite a bit of training and practice to learn to make effective use of a word processor.

At the current time, most schools have decided that students should learn both paper-and-pencil writing and also writing in a word-processing environment. Moreover, students have to learn which of these aids to writing is more appropriate for themselves and for various types of writing tasks that they face. In this case, the computer has added to the amount that students have to learn.

Computer as Tool

The previous section raised a number of questions about learning to use tools as an aid to solving problems. All of these questions can be directed specifically toward computers.

Historically, most people thought of computers only as an aid to solving math problems. That line of thinking has gradually changed. Now, people recognize that computers are useful aids to problem solving in all academic disciplines.

Over the past 50 years, computers have gotten more and more cost-effective. Indeed, the cost of carrying out a given set of steps has gone down by a factor of a million or more since the first few general-purpose computers were constructed. This means that the computer is becoming a more costeffective aid in problem solving.

As with other tools, it is useful to think about the computer tool both from a concept and a process point of view. A computer is an information-processing machine. Some key concepts include:

- 1. A computer can be thought of as an automated symbolmanipulation machine. It can work with the same symbols that people use in writing and mathematics. It can also work with other symbol sets, such as musical notation.
- 2. A computer can rapidly and accurately carry out a detailed set of instructions that has been specified in a computer program. That is, computers are very good at carrying out processes.
- 3. A computer can store a great deal of raw data and information. This raw data and information can be used in solving problems.
- 4. A computer system can be designed to interact both with people and with machines—including other computers,

instruments that gather data, and automated factory machinery. All of this interaction can occur between people and machines that are widely dispersed and are connected by a telecommunications network.

The diagram given in Figure 4.1 represents one way of looking at the computer environment for problem solving.



Figure 4.1 Computer environment for problem solving.

Computer-assisted problem solving begins with recognition of a problem situation and posing a problem (A and B in Figure 4.1). The people in (A) work to represent the problem in a form that allows them to use a computer system (C) to aid in solving the problem. Typically, this representation process and overall solution process requires substantial interaction between A, B, and C. There are a large number of people (D) who have developed the hardware and software that will be used to help to solve the problem. When you use the computer as a tool, you are building on the work of these people.

Now, imagine for a moment that the hardware and software developers (D) could read your mind and anticipate your every need. For any problem that you wanted to use a computer to help solve, the hardware and software would exist to accommodate your exact needs. All you would have to do is interact with the hardware and software—feed in the data needed for the specific problems that you want to solve. Certainly that would cut down on the effort you would need to put forth learning to use a computer to solve problems and actually using a computer to solve problems.

Obviously, the hardware and software developers cannot read your mind and anticipate your every need. Moreover, you are not the only customer for their services. Commercially available hardware and software has been developed to fit the needs of the mass market. If the problems that you want to solve by using a computer are similar to the types of problems that lots of other people want to solve, then it is likely that the commercially available hardware and software will suit your needs. However, if the problems that you want to solve and the tasks that you want to accomplish are out of the ordinary, you may find that the commercially available hardware and software do not specifically fit your needs.

This places an increased learning burden on you. You have to learn to represent the problems that you want to solve in a form that fits the available hardware and software. You may have to spend a great deal of time and effort to learn to make effective use of the hardware and software.

If the problems that you want to solve are quite out of the ordinary, you may need to learn to write computer programs. Computer programming is discussed in Chapter 9.

Building-Block Resources

Think about a problem that you routinely encounter and solve. For example, perhaps you are a touch typist. Then, you routinely and easily solve the problem of typing words. You can compose at the keyboard, using your conscious thinking efforts to decide on the words you want to write. An automatic part of your brain and body accomplishes the task of getting the words typed.

Perhaps you are very good at shooting baskets in basketball. You routinely, easily, and without conscious

thought, solve the problem of tossing a basketball through the hoop.

The typing and basketball examples illustrate a very important idea about problem solving. If you frequently encounter the same problem, it may be worth quite a bit of study and practice time to develop the knowledge and skills needed to solve the problem easily. It may be possible to develop a high level of automaticity in solving the problem.

In this book, we will use the terminology *building-block resource* (BBR) to represent a problem that you are highly skilled in solving or a task that you are highly skilled in accomplishing. The terminology BBR is meant to suggest that such knowledge and skills can be used as building blocks in solving still more complex problems. If typing is a BBR for you, this can help you solve a writing problem. If shooting free throws is a BBR for you, this can help you to solve the problem of being a good basketball player.

It is evident that each person has their own set of BBRs. Take a look at the list given in Table 4.1. Which of the problems listed are BBRs for you? Add a few items to the list so that it contains a number of your own BBRs.

Fill in the column on the far right. If the item is a personal BBR, make an estimate of how many hours it took you to acquire your current level of knowledge and skills. If an item is not a personal BBR, make an estimate of how long it would take you to make it a personal BBR. The key concept here is that a person can acquire additional BBRs. However, it can take considerable time and effort to do so.

Possible BBR	Howard Gardner's 7 intelligences classification	Length of time to acquire this as a BBR
Touch type		
Ride a bicycle		

The middle column is discussed in the Activities and Self-Assessment section of this chapter.

Correctly spell when writing	
Converse in two languages	
Do accurate and rapid mental arithmetic	
Play chess well	
Throw a softball accurately	
Determine which colors complement each other	
Get along well with almost everybody that you meet	
Use the memory keys on a handheld calculator	
Accurately follow a recipe	
Easily sing in tune	
Easily find your way when in a new city	
Easily communicate in writing	
Really be in touch with your own feelings	

Table 4.1 Some possible BBRs.

It is evident that having a large repertoire of BBRs can be quite useful in solving problems and accomplishing tasks. Thus, quite a bit of formal education is designed to help students gain BBRs. However, it often takes a lot of time and effort to gain a reasonable level of expertise in an area. Thus, it can take a great deal of time and effort to build a wide range of BBRs. Time—not neural intelligence—is the limiting factor.

Computer-Based Building-Block Problems

Computers can carry out many tasks rapidly and accurately. For example, a computer can do a spell check. It can alphabetize a list. It can sort addresses by ZIP code. It can rotate a three-dimensional drawing of a house, showing the different perspectives. It can change the color of an object in a picture. It can calculate the energy efficiency of a building design.

Think of these tasks as BBRs. It is easy for a person to learn to use a computer to accomplish such tasks. By learning to use a computer, a person can quickly acquire a large number of BBRs. This certainly adds a new dimension to education and to developing expertise in solving problems.

Computer-based BBRs can be the basis for significant changes in our educational curriculum. Consider, for example, the time and effort it takes for a student to learn to represent a set of data using a pie chart (a circle graph). The student needs to know how to compute percentages, how to compute percentages of 360 degrees, and how to use a protractor. Moreover, consider how long it takes to actually accomplish such a task for a set of data. But, a modern word processor contains provisions for converting a table of data into a pie chart or into other graphical representations. This is accomplished as quickly and easily as such a word processor can spell check a paragraph or alphabetize a list. Both the time to learn to accomplish a pie charting task and the time to actually accomplish such a task are quite short.

Chunking Using BBRs

A person's short-term memory is quite limited in terms of the number of items of information that it can simultaneously deal with. It is often suggested that for a typical person, the number of pieces of information that short-term memory can deal with is about 7 ± 2 . Thus, a typical person can go to a phone book, look up a seven digit phone number, and remember it long enough to dial it.

However, it is evident that people can learn to actively think about far more complex situations than is suggested by the number 7 ± 2 . Paper and pencil are important aids to this short-term memory limitation. Another big help is *chunking*. I work at the University of Oregon. All phone numbers at the University of Oregon begin with the three digits 346. I live in Oregon. At the current time, all of Oregon, with the exception of the Portland area, has the long distance prefix 541. Thus, any University of Oregon phone number is (541) (346) followed by four digits. That is, it is two "chunks" and four individual digits. The two chunks are "State of Oregon prefix" and "University of Oregon prefix." The total, six pieces of information, is something that I can deal with.

One of the reasons that BBRs are so important is that they are chunks. When I am thinking about how to solve a complex problem, I can think in chunks that are BBRs for me. That relieves my short-term memory from having to deal with small details that would quickly overwhelm its capacity.

Education for Conceptual Understanding

This chapter began with a discussion of handheld calculators. The handheld calculator example was used to help point out that there is a difference between a concept (such as division) and a process (actually doing division). Educators and others are not in full agreement about how much educational emphasis should be placed on understanding concepts and how much emphasis should be placed on learning processes.

Interestingly, such disagreements tend to focus on relatively simple BBRs such as the four basic arithmetic operations. There is much less disagreement on more complex BBRs. For example, almost all handheld calculators have a square-root key. The square root of 9 is 3 because $3 \times 3 = 9$. (For mathematically inclined people, -3 is also a square root of 9.) An eight-digit calculator indicates that the square root of 2 is 1.4142135. It is possible to calculate the square root of a number by various paper-and-pencil processes. The process that used to be taught in high school algebra courses is more complex than long division. This has been dropped from the curriculum in most algebra courses—and replaced by having students use a calculator or a computer.
The concept of square root is relatively easy to learn. It is very easy to acquire the calculation of a square root as a BBR if one has a calculator available. But, this still leaves a person with no understanding of how it is possible for a calculator to determine the square root of a number. This raises the question of whether it is appropriate for students to acquire the square root BBR with no conceptual understanding of how a calculator is able to accomplish such a task.

This question has no simple answer. Moreover, the question is made more difficult as complex tools, such as the computer, come into routine use in education. Should a computer system be viewed as a "black box" that magically carries out assigned tasks? What level of understanding should students have of the software they are using? Is it alright to use software that rotates three-dimensional objects, but have no idea of the underlying mathematics used in representing and rotating three-dimensional objects?

To a reasonable extent, educators agree that students should understand the underlying concepts of both the problems that they are working to solve and the tools and processes being used to solve them. A reasonable level of understanding can be helpful in detecting and correcting errors, and in handling difficulties that arise as you use various aids to solve problems.

Let's take a specific example. A computer is an electrically powered machine, operating off of household current or batteries. The same holds true for a calculator, radio, television set, tape recorder, and CD player. One underlying concept for all of these tools is that they will not work without power. Thus, if such a device fails to work, perhaps the first thing to do is to check the on/off switch. Perhaps the second thing to do is to check to see if the device is plugged in and the electric current is working, or if the battery has a charge in it. By understanding the underlying theory of how such devices are powered, a person can detect and correct several common sources of malfunction.

An understanding of underlying concepts is very important in detecting and correcting errors or malfunctions. Let's take this example back to calculators. A first-grade student can learn what sequence of keys to press on a calculator when faced with a calculation such as 285.69 divided by 54.8. The student can memorize how to do this with a high level of accuracy. But, what level of understanding does the student have of decimal notation and of long division? Does the first grader understand what problems can be represented and solved by division? How does the student detect an error, such as not keying in the decimal point?

These same difficulties carry over to computers. It is possible for a person to acquire a large number of BBRs by using a computer. Some may be so intuitively obvious and transparent that a high level of underlying concepts seems inherent to the BBR. For example, you know a great deal about rotating objects and what they look like from different angles. You know a great deal about what objects look like on a television screen. Combining these areas of knowledge gives you good insight as to what to expect when you use a computer program to rotate objects in three dimensions.

Contrast this with a sophisticated mathematical technique such as the amortization of a loan. Unless you have had specific education in this area, the chances are that you do not understand the underlying concepts of the problem to be solved. Thus, you have no way of knowing if the mathematical procedure is applicable to a particular problem you want to solve or whether you have made a gross error in entering the problem into the computer.

As you learn more about computers, you will continually be faced with the issue of how much understanding you need. How much time and effort are you willing and able to devote to learning underlying concepts? To what extent are you comfortable and satisfied with using computer hardware and software as a black box? Does your previous training and experience help you to detect and correct errors that you make when using computer hardware and software? How important is it that you be able to detect errors?

Activities and Self-Assessment

1. Go back to the list of problems that you developed in the Activities and Self-Assessment section of Chapter 1. Think of this list in terms of the physical tools that you

would use to help solve the problems. Create a threecolumn table to summarize your analysis. Some sample entries are given in Table 4.2.

Problem	Tools	Discussion	
Balancing checkbook	Paper, pencil, calculator	I have trouble being systematic about recording the needed data as I write checks, make deposits, and make withdrawals.	
Light doesn't work	Step ladder, light bulb	I keep a supply of light bulbs in my closet. Once, the problem was a broken switch.	
Printed report to boss falls apart	Stapler, staples	It seems as if I am always running out of staples.	
Wet hair from rain	Hat	I keep a rolled up hat in my coat pocket.	

Table 4.2 Sample list of problems you may have come up with at theend of Chapter 1.

2. There are lots of problems where a computer is of little or no use. Go back to the set of problems that you wrote down in dealing with Table 1.1 at the beginning of Chapter 1. Add a few more problems to the list, with special emphasis on problems where you feel a computer might possibly be a useful resource. Then add a fourth column to the table as illustrated in Table 4.3. Classify each of your problems on a "Potential Computer Usefulness" scale. For example:



Figure 4.2 Scale showing potential computer usefulness.

Problem	Difficu lty	Main Resources	Computer Usefulness
Communicati on with a friend	Easy	Telephone, my voice	3—I know that the telephone companies use computers.
Written communicati on with an irate customer	Hard	Stored data on the transaction, careful thinking, advice from my boss, computer	4—Our customer transaction records are computerized. I write best when I use a word processor.

Table 4.3 Determining how useful a computer is with respect to a
problem.

- 3. Table 4.1 lists a number of possible BBRs. Add some items to this table. Fill in the middle column, analyzing how each of the potential BBRs relates to the seven intelligences in the Howard Gardner list. Also, analyze each potential BBR from the point of view of how much time it would likely take you to acquire a reasonable level of expertise. In this analysis, indicate how you might measure whether you had acquired the level of expertise that you have in mind.
- 4. How do you feel about learning to do paper-and-pencil arithmetic versus learning to use a calculator? Analyze your feeling from the point of view of understanding concepts and understanding processes. Some people consider a calculator to be a black box. For you, personally, are the algorithms that you have memorized for multiplication and long division actually black boxes? Extend your discussion to the use of a calculator to calculate square roots.

5. Take a look at the word processor that you use. Does it contain a graphing package? If so, explore use of this package. Discuss how easy it is to learn how to use and how easy it is to use. Compare and contrast this with by-hand graphing of data.



Accumulated Knowledge as Resource

You carry around a great deal of knowledge and experience in your mind/body. That is, you have a great deal of experiential intelligence. This has been accumulated over many years. You use much of this knowledge freely and easily—nearly effortlessly. For example, when you are talking, you formulate an idea in your head, translate it into words, and speak the words. When you want to use a fork to get a bite of food from a plate and transfer it to your mouth, little conscious effort is required.

As is suggested by Figure 5.1, problem solving can be viewed as an interaction among the humans working to solve a problem, accumulated knowledge of the human race, and the problem to be solved. (Remember, quite a bit of the accumulated knowledge needed for a particular problem may reside in the mind/body of the problem solver.)



Figure 5.1 Interacting facets of the problem-solving process.

Building on Previous Knowledge

You routinely solve many of the problems that you encounter by just using the knowledge you have accumulated throughout your lifetime. However, you are but one person. Collectively, the human race has accumulated a great deal of knowledge, far more than any one person can learn. Much of this accumulated knowledge is stored in people's heads; however, a great deal is stored in books, films, tapes, paintings, and other artifacts. Now, an increasing percentage of that knowledge is stored in computers.

Remember, a computer is both a storage device and a processing device. Thus, a computer is an excellent aid to building on the previous work of other people.

You have heard the expression, "Don't reinvent the wheel." Many of the problems that people want to solve can be solved by finding out how others have solved the same problem in the past, and then doing what they did. Computers are making it easier and easier to find out what others have done when faced with particular problems. The libraries and other information sources throughout the world are being computerized and networked together by the Information Superhighway.

Moreover, software (called groupware) has been developed that helps groups of people work together to solve problems. Progress in computerizing and networking the world is having a profound impact on problem solving.

Procedural and Declarative Knowledge

Your mind/body "knows" a great deal of information. As Figure 5.2 suggests, this can be divided into two main categories: procedural and declarative. Declarative knowledge can be divided into two categories: episodic and semantic. This section explains each of these types of human knowledge.



Figure 5.2 A model of human memory.

Procedural knowledge is knowledge about how to do things; it is often kinesthetic. Your mind/body has mastered many different physical procedures for accomplishing tasks. For example, when you were a young child, you did not know how to tie your shoes. Now, you tie your shoes without conscious effort. You have mastered this skill; it is stored in your mind/body.

Touch typing (touch keyboarding) provides another example of procedural knowledge. Your procedural knowledge can be thought of as a set of BBRs, immediately available to you in problem solving.

Declarative knowledge is concerned with the facts that you know. You have memorized a great many facts or pieces of information. For example, you know the alphabet, some telephone numbers, addresses, names of people, dates, and so on. Such declarative information is important to problem solving.

People often differentiate between procedural and declarative knowledge by calling the former "know how" and the latter "know what." Your "know how" and your "know what" work together as you solve problems and accomplish tasks.

Declarative knowledge can be divided into two categories. Episodic declarative knowledge is concrete. The knowledge is established when an episode or unusual event occurs. (Think of this as one trial learning.) Declarative knowledge is very personal, intimately tied to specific episodes or settings such as: one's first kiss on a date; one's wedding; or a particularly nice vacation event.

Semantic declarative knowledge is more abstract. It tends to be symbolic and context free. You probably do not remember when you first learned the alphabet, the counting numbers, or the chemical formula for water. Semantic declarative memories are relatively easy to create through study, and they are easy to forget.

Procedural knowledge and semantic declarative knowledge take time and effort to acquire. Moreover, a certain amount of forgetting occurs. The adage "Use it or lose it" tends to apply. Procedural knowledge and semantic declarative knowledge differ somewhat in this regard. You are probably skilled in memorizing pieces of information for a test. This is semantic declarative information. Once the test is over, you soon forget what you have memorized. If you find that you need this information a few years later, the task of memorizing it at that later date is not helped much by the earlier memorization process.

However, once you develop a reasonable level of procedural skill in a particular area, such as touch typing or piano playing, quite a bit of this procedural knowledge stays with you for a lifetime. Even if you don't type or play the piano for years, you can quickly regain a reasonable percentage of your original skill. A somewhat different way of thinking about this is that procedural knowledge BBRs tend to last a lifetime. Our educational system struggles with the question of what knowledge all students should acquire. For example, is there some set of declarative information that every student should be required to learn? Hirsch (1988), whose book was a national bestseller, argues that the answer is "yes" and lists 5,000 "essential names, phrases, dates, and concepts that every American needs to know." From a communication point of view, it is helpful that speakers and listeners share a common core of declarative knowledge. Hirsch argues that this common core of declarative knowledge is a type of cultural literacy.

However, many people are critical of such a simplistic approach to education. Part of the argument against the Hirsch approach is that it does not place enough emphasis on higher-order skills—thinking and problem solving using one's declarative knowledge. Instead, it fosters a back to basics approach to education in which the goal is to memorize a lot of material.

Domain Specificity (Expertise in a Specific Domain)

It is useful to talk about the domain of a particular field of expertise. Some domains are formal academic fields; thus anthropology, biology, chemistry, dentistry, economics, and so on are each a domain. However, there are many more domains that are not formal academic fields. For example, you may have considerable expertise in antique collecting, bird watching, card playing (poker or bridge), driving in congested traffic, and so on.

In this book, we use the word *domain* to refer to a coherent area in which one solves problems and accomplishes tasks. Through study and practice, one's level of expertise in the domain can be increased.

Researchers have carefully examined the nature of increasing one's expertise in a particular domain. A key idea is domain specificity, or domain-specific knowledge. The more knowledge you have about a domain, the better problem solver you tend to be on problems within that domain. One way to think about domain specificity is from an information-retrieval model of problem solving. A person studying a particular field gradually gains knowledge about each of the major problems within that field. The person gradually memorizes a large number of patterns that correspond to the frequently occurring problems or subproblems of the domain, and also memorizes what to do when each of these patterns is encountered. In essence, the person acquires BBRs. Thus, solving a problem often consists of recognizing that the problem is somewhat familiar, and then retrieving from one's mind the information needed to solve the particular problem.

Or, if the problem is not recognized as a familiar pattern, perhaps it breaks easily into pieces that are recognized. This is one strategy that is useful in many different domains. Each domain has problem-solving strategies that are quite specific to the domain. Through study and practice one learns some of these strategies and becomes skilled in using them.

Knowing a lot about a particular domain is a combination of knowledge, skills, and experience. It can include considerable formal knowledge (book learning), or it may consist entirely of informal knowledge (learn by doing, street smarts). In either case, a lot of time and effort is needed to develop a reasonable level of expertise. For the most part, this time and effort is spent gaining knowledge and skills that other people in the domain have already acquired in the past.

Learning is an individual process. When you are struggling to learn something, it doesn't help you very much to know that millions of people have had the same learning struggles in the past. Indeed, you may even find this frustrating. You may wonder why you have to spend so much time learning to do something that other people have already learned how to do.

Don't Reinvent the Wheel

One of the key ideas in problem solving is "Don't reinvent the wheel." In many situations it is possible to build on your own previous work and that of others. The following few sections of this chapter discuss three general categories of knowledge and skills that a person can acquire and build on: declarative and procedural knowledge; concepts and underlying theory; and strategies.

As the diagram in Figure 5.3 suggests, these three categories of knowledge and skills are closely related. In essence, both the concepts and theory category and the strategies category are components of the declarative and procedural knowledge category. However, from a problem-solving point of view, it is useful to discuss them separately.



Figure 5.3 Three general categories of knowledge and skills.

Problem solving makes use of all three categories of knowledge. The specific balance of knowledge that is needed for problems in a particular domain varies both with the domain and with the person working to solve a problem.

Declarative and Procedural Knowledge

Your mind/body has a great deal of procedural knowledge. You use it as you walk, talk, drive a car, ride a bicycle, write, keyboard, and so on. Procedural knowledge plays an important role in athletics. Athletes practice fundamentals over and over again throughout their careers. These fundamentals are the BBRs necessary to achieve a high level of athletic performance. In the remainder of this section we focus on declarative knowledge, especially semantic declarative knowledge. Much of this is based on "book" learning. Let's begin with a thought experiment. Suppose that you had memorized the contents of every book and journal that is currently in the United States Library of Congress (a very extensive library) and you could easily recall any of this memorized information whenever you liked. Would this memorized knowledge help you to be an expert problem solver across many different domains?

The basic question is: how important is memorization in solving problems and accomplishing tasks? Researchers on expertise have examined this question in some detail. The answer has several parts:

- 1. There are many recurring problems. Memorization of how to solve them can be useful. Moreover, these memorized solutions serve as BBRs. A large repertoire of BBRs can be useful in problem solving.
- 2. Real-world problems occur in varying contexts or environments. Often it is difficult to recognize that a particular context or environment is just a slight variation of something that you know. Thus, rote memorization is, for the most part, only useful in problem solving if one has good skills at transferring the memorized information into the actual problem-solving situation. Transfer of learning is discussed in the next chapter.
- 3. Many of the problems that people want to solve are unique—they have not occurred frequently enough in the past to warrant their inclusion in books, journal articles, and other reference materials. Thus, rote memory by itself, even with good transfer of learning skills, will not make you an expert at solving the full range of problems in a domain.

A number of different researchers have studied world-class experts. A world-class expert is really good—relative to the best in the world. We are used to the idea of world-class athletes. But, the concept of world class exists in all areas of human productivity and problem solving. There are worldclass economists, musicians, mathematicians, political leaders, writers, and so on.

The chances are that you are not a world-class expert in any particular domain of problem solving. Unfortunately, for the most part expertise researchers have not studied more ordinary people—people who are not world-class experts in anything—and their efforts to achieve their full potential level of expertise. Thus, you will want to interpret the information that follows in light of your own personal goals.

Researchers have examined world-class experts in diverse areas, such as business, chess, mathematics, music, and physics. Two ideas emerge. First, it takes many thousands of hours of hard work to achieve world-class expertise. Several researchers have indicated that it takes about 10 years of hard work, as well as suitable instruction and coaching. The second important idea is that an essential part of the expertise seems to be that of having committed to memory approximately 50,000 patterns and what to do when one of these patterns (or a slight variation thereof) is encountered. To a large extent, this memorized knowledge is a set of BBRs that is internalized at a subconscious level.

Similar studies in domains that depend quite a bit on procedural knowledge, such as sports, have yielded similar results. World-class gymnasts can be "produced" by careful selection of young children who seem to have a great deal of natural kinesthetic intelligence, and then putting these children through approximately 10 years of intense training.

In summary, research studies of world-class experts suggest that for any person to achieve their full potential level of expertise in some domain, they will need to work quite hard for many years and internalize a great deal of declarative and/or procedural knowledge.

The educational implications of this seem clear. Persistence is essential. It takes many years of hard work in a field to develop a high level of expertise. If you want to be good in a field—whether sports or academics—you must be willing to work hard at learning the field. This hard work needs to be continued over a long period of time, enough time to internalize many thousands of patterns and what to do when these patterns are encountered.

Of course, this creates a dilemma. Suppose you want to be good in a number of different fields? Suppose that you want to solve interdisciplinary problems? Suppose that you frequently encounter new problems that are outside the domains that you have studied? Life is not long enough to spend years becoming an expert in a large number of different domains. Time is a limited and limiting resource.

A partial answer to this dilemma is that each of us can work toward a balance of general breadth and specific depth. It is possible to acquire a useful level of knowledge and skills in a large variety of domains. If one of your goals is to acquire a high level of expertise in one or two domains, then the chances are that you will not develop as much breadth as people who don't focus so much attention on narrow domains.

A second answer lies in focusing on the reflective intelligence component of intelligence that Perkins (1995) discusses in his book. Reflective intelligence includes strategies that cut across many domains. It includes habits of mind and attitudes that are essential to attaining expertise in any domain. This will be discussed more in the next chapter.

In any event, the tools that humans have produced can be of immense help in developing a reasonable level of expertise in many different domains. Later in this chapter, we will focus on the computer-as-tool strategy in developing broad-based problem-solving skills.

Studies of Chess Experts

A number of studies of world-class experts have focused on chess players. Moreover, there has been a great deal of progress in developing computer programs that can play chess well. It is no longer unusual to have chess tournaments in which humans and computers compete against each other. The studies of human chess players and the work to develop computer programs that play chess well have led to considerable insight into problem solving. This section explores some of the ideas that have come out of this research. Some people have a knack for chess. This may be a combination of spatial and logical/mathematical intelligence. Through many years of hard work and conscious directed effort, they can develop this innate ability into a high level of chess expertise. It is important to emphasis the "hard work and conscious directed effort." Many people play a game such as chess or bridge for years, but don't improve appreciably after the first few years. They do not put in the hard work and conscious directed effort needed to continue their improvement.

One of the classical chess studies had both expert and novice chess players look for a few seconds at a collection of chess pieces arranged on a chess board and attempt to memorize their placement. If the pattern of pieces is a naturally occurring one from the middle of a reasonably wellplayed game, an expert chess player can accurately place more than 90% of the pieces after viewing the board for 10 seconds. A novice player accurately places less than 25% of the pieces under the same conditions. And, if the pieces are merely located at random on the board, both the expert and the novice score at less than 25%. That is, an expert chess player can quickly recognize naturally occurring chessboard positions but is no better than a novice at memorizing random board positions. This is taken as evidence that a mental pattern-matching process is occurring at a subconscious level.

Humans are very good at subconscious pattern matching. You do it when you recognize a familiar face, a familiar voice, or a familiar smell. To-date, humans are far better at such pattern-matching tasks than computer systems. This presents a major challenge to researchers in the field of artificial intelligence.

However, computers are far better at rote memory than humans. Thus, you might think that computers could be very good at chess through rote memory. Just program the computer to play through every possible game of chess and memorize every winning combination of moves.

However, that is impossible. It has been estimated that there are 10^{120} different possible games in chess. The number 1 followed by 120 zeros is a very large number! If a computer could memorize a trillion trillion possible board positions every second, it would still take far longer than the age of the universe for it to memorize even a tiny fraction of the possible board positions.

That does not mean that rote memorization is not important in chess. At the beginning of the game, the number of possible moves is limited and these have been studied extensively. Memorizing opening moves is a good way to get better at chess. The same holds for end-game situations.

Rote memory is useful in problem solving in every domain. But, even such a game as chess overwhelms a rote-memory approach to problem solving. Many of the problem-solving domains in the real world are far more complex than chess. And, there are thousands of different domains. Thus, rote memorization is of limited value in gaining increased expertise in problem solving.

Something else is needed in addition to memorization. Part of the answer is the learning of strategies and developing skill in their use. Another part of the answer lies in learning general concepts and underlying theory. These ideas are discussed in the sections that follow.

A Strategy is a General Plan

Figure 5.3, shown previously in this chapter, diagrams three general types of knowledge and skills useful in problem solving. Both strategies and concepts and theory can be thought of as special types of declarative and procedural knowledge.

This and the following section discuss strategies. A *strategy* can be thought of as a plan, a heuristic, a rule of thumb, a possible way to approach the solving of some type of problem. For example, perhaps one of the problems that you have to deal with is finding a parking place at work or at school. If so, probably you have developed a strategy—for example, a particular time of day when you look for a parking place or a particular search pattern. Your strategy may not always be successful, but you find it useful.

Every problem-solving domain has its own strategies. Research suggests:

- 1. There are few strategies that are powerful and applicable across all domains. That is, in problem solving, we face the issue of domain specificity. One needs to know a great deal about a particular domain to be good at solving problems within that domain.
- 2. The typical person has few explicit strategies in any particular domain. This suggests that if we help a person gain a few more domain-specific strategies, it might make a significant difference in overall problem-solving performance in that domain. It also suggests the value of helping students to learn strategies that cut across many different domains.

Do you know some general strategies that are useful in many different problem-solving domains? How about the idea of breaking big problems into smaller problems. This is called the *top-down* strategy. The idea is that it may be far easier to deal with a number of small problems than it is to deal with one large problem. Another strategy is to draw a picture, diagram, or other graphic aid to represent the problem. This is the *draw-a-picture* strategy.

You have lots of domain-specific strategies. Think about some of the strategies you have for making friends, for learning, for getting to work on time, for finding things that you have misplaced, and so on. Many of your strategies are so ingrained that you use them automatically—without conscious thought. You may even use them when they are ineffective.

The use of ineffective strategies is common. For example, how do you memorize a set of materials? Do you just read the materials over and over again? This is not a very effective strategy. There are many memorization strategies that are better. For example, a simple strategy is pausing to review. Other strategies include finding familiar chunks, identifying patterns, and building associations between what you are memorizing and things that are familiar to you.

Some learners are good at inventing strategies that are effective for themselves. Most learners can benefit greatly from some help in identifying and learning appropriate strategies. In general, a person who is good teacher in a particular domain is good at helping students recognize, learn, and fully internalize effective strategies in that domain. Often this requires that a student unlearn previously acquired strategies or habits.

A General Strategy for Problem Solving

Here is a general six-step strategy that you can follow in attempting to solve almost any problem. Note that there is no guarantee of success. However, this six-step strategy might get you started on a pathway to success.

- 1. Understand the problem. Among other things, this includes working toward having a clearly defined problem. You need an initial understanding of the Givens, Resources, and Goal. This requires knowledge of the domain of the problem.
- 2. Determine a plan of action. This is a thinking activity. What resources will you use, how will you use them, in what order will you use them? Are the resources adequate to the task?
- 3. Think carefully about possible consequences of carrying out your plan of action. Place major emphasis on trying to anticipate undesirable outcomes. What new problems will be created? You may decide to stop working on the problem or return to step 1 as a consequence of this thinking.
- 4. Carry out your plan of action. Do so in a thoughtful manner. This thinking may lead you to the conclusion that you need to return to one of the earlier steps. It is this reflective thinking that leads to increased expertise.
- 5. Check to see if the desired goal has been achieved by carrying out your plan of action. Then do one of the following:
 - If the problem has been solved, go to step 6.
 - If the problem has not been solved and you are willing to devote more time and energy to it, make use of the knowledge and experience you have gained as you return to step 1 or step 2.

- Make a decision to stop working on the problem. This might be a temporary or a permanent decision. Keep in mind that the problem you are working on may not be solvable, or it may be beyond your current capabilities and resources.
- 6. Do a careful analysis of the steps you have carried out and the results you have achieved to see if you have created new, additional problems that need to be addressed. Reflect on what you have learned by solving the problem. Think about how your increased knowledge and skills can be used in other problem-solving situations. (Work to increase your reflective intelligence!)

This six-step strategy for problem solving is worth memorizing. Try using it with the problems that you encounter. Eventually it will become second nature. You will probably find that learning and using this strategy improves your overall ability to solve the types of problems you encounter in your everyday life.

Understanding of Concepts and Underlying Theory

Figure 5.3 given earlier in this chapter diagrams the three general types of knowledge and skills useful in problem solving. In this section, we will discuss concepts and theory.

Although concepts and theory are intertwined, it is often possible to have a useful understanding of a concept without knowing the underlying theory. Einstein's theory of relativity involves complex mathematics and physics. Only a modest number of people in the entire world have a good understanding of this underlying theory. But, you probably understand the concept that no object can move faster than the speed of light. Perhaps you understand the concept that the mass of an object increases as its velocity increases.

For a more mundane example, consider a computer model of a building, where the computer program allows you to view the building both from its outside and from its inside, from any angle. The concept is easy enough to understand. Such a computer program may be easy to use. However, it is a real challenge to develop such a computer program. This area of problem solving is part of the field called virtual realities.

In academic fields, the general concepts and underlying theory tend to be well developed. Within academic domains, a lot of research has focused on the importance of understanding underlying concepts and theory as an aid to problem solving. The results support the importance of understanding concepts and theory.

High-school geometry provides a good example. It is possible to memorize the proofs of a large number of theorems. Many people manage to pass a geometry course by using this approach. However, this approach is nearly useless when one is asked to deal with problems that they have not encountered before. In addition, research suggests that this approach is not of lasting value. The memorized proofs are soon forgotten and the student is left with little real knowledge of geometry that can be applied to problem solving in math, the other sciences, engineering, and so on.

Outside of academics, domains vary considerably in the nature and extent of the underlying concepts and theory. Often the underlying concepts are intuitive and may be difficult or perhaps impossible to put into words.

For example, a skilled craftsperson or artist produces a remarkable product. Can this person put into words the method of making a beautiful work of art? What are the underlying concepts or theories that make this product so outstanding? Can these concepts be stated as a set of rules that others can learn to follow?

A skilled soccer player has a remarkable sense of space, of being in the right place at the right time, and of making the right plays. This person has high levels of kinesthetic and spatial intelligence. Is there an underlying theory—the fundamental concepts of soccer—that this person has learned and follows? Can this theory be represented and taught in a manner to help others develop a high level of expertise in playing soccer?

What is the Right Balance?

Suppose that you have selected a domain in which you intend to develop an increased level of expertise. How much of your learning time and effort should you put into increasing your declarative and procedural knowledge? How much should you put into learning new strategies? How much should you put into learning concepts and underlying theory?

These questions do not have a simple answer. The answers vary both with the domain and with the person working to gain increased expertise in the domain. Thus, it is possible that two people will each have a high level of expertise in a domain, but will have different profiles of knowledge and skills in the domain. The time and effort required to gain the high level of expertise may vary considerably. This may be dependent on the learner's level of neural intelligence, especially as it relates to the specific domain. It may also be quite dependent on the person's experiential intelligence and reflexive intelligence.

This is a challenge to educators and to learners. A teacher or coach may have a great deal of general knowledge about how people learn and develop increased expertise in a domain. However, the teacher or coach cannot know your particular set of knowledge and skill attributes as well as you yourself can know them. Thus, knowledge of self (intrapersonal intelligence) is very important. Ultimately, you are responsible for your own learning.

Computers—A New Aid to Problem Solving

We conclude with a brief discussion of computers and how they relate to the other main themes of this chapter.

The original computers were designed to help do math calculations. Now, of course, computers are a versatile tool, useful in many different domains. In many domains, computers strongly affect the balance of knowledge, strategies, and concepts needed for a high level of performance. The later chapters of this book discuss these ideas in detail. Here is a brief overview.

- 1. In terms of declarative and procedural knowledge:
 - Computers are powerful aids to the storage and retrieval of information. In many cases, using a computer to store and retrieve information is an effective substitute for memorizing the information. You can "look it up" using a computer instead of looking in your memory.
 - Computers and computerized equipment can carry out many different procedures. In many cases, use of computerized or automated procedures is an effective substitute for mind/body procedural knowledge.
- 2. It has proven difficult to incorporate strategic knowledge into computers. However, progress has occurred in a number of domains. The field of computer science known as artificial intelligence addresses these types of issues.
- 3. For the most part, it has proven difficult to incorporate understanding of concepts and underlying theory into computers. To put it bluntly, computers are dumb! Even the best of the artificially intelligent computer systems have almost no understanding of or feeling for what they are doing. Thus, they are not able to detect colossal blunders.

This analysis suggests that there are some things that computers can do well, and there are some things that computers do poorly. To a reasonable extent, the relative strengths of computers correspond to relative weaknesses of people, and vice versa. It is difficult for a person to memorize a book verbatim; but a computer can memorize thousands of books. Humans are not good at repetitious tasks and are prone to error; computers are fast, accurate, and not bored by repetitious tasks.

Humans are good at understanding concepts and using these concepts to make decisions. Humans are good at learning strategies and then making appropriate use of these strategies as they work to solve problems. Humans are better than computers at dealing with concepts and strategies. As you increase your knowledge of using computers to solve problems, you will gradually develop a repertoire of strategies that will be particularly useful in a computer environment. One good example is the strategy of guess and check (sometimes called trial and error). Guess and check is useful in many domains and is often done without the use of a computer. You have a problem to solve. You make an educated guess at a solution. You then check to see if you have actually solved the problem. Perhaps you use this technique when working on a crossword puzzle or in putting together a picture puzzle.

There are many situations in which a computer is helpful in doing the work of generating a guess. Computer graphics, for example, can make it easy to experiment with a design. You can look at the results of your experiment to see if it solves the problem you had in mind. You and the computer working together become a powerful combination, able to quickly accomplish tasks that neither you nor the computer can do alone.

Another good example comes from the use of spreadsheets. You develop a spreadsheet model of a particular business situation. You can then ask "What if?" questions. A few keystrokes allow you to make changes in the business situation and then to examine the results. Such use of spreadsheets provides a powerful example of making effective use of the strengths of both people and computers.

A third type of example comes from situations in which the computer itself can generate guesses, either in a random or a systematic manner. A computer may be able to check tens of millions of different guesses in a relatively short period of time. This brings a new dimension to problem solving.

Activities and Self-Assessment

- 1. Select several different domains in which you have a reasonable level of expertise. For each, make a list of:
 - major types of procedural knowledge and declarative knowledge needed in the domain;

- useful strategies in solving the problems in this domain;
- key concepts and underlying theory needed for expertise.
- 2. Do some introspection on your relative strengths in declarative and procedural knowledge. Are you better at learning and using declarative knowledge or at gaining and using procedural knowledge? Cite some evidence to support your analysis.
- 3. Recall a relatively complex problem that you have solved. Analyze this problem from a point of view of the six-step general problem-solving strategy given earlier in this chapter. Which of the six steps did you follow? What did you do in this overall problem-solving process that might contribute to your experiential intelligence or reflective intelligence?
- 4. Make a list of things that you are sure you can do better than a computer. Make a list of things that you feel a computer or computerized equipment can do better than you. Does each individual list contain some common themes? That is, are there some general categories in which you feel you can outperform a computer and vice versa?
- 5. What is your domain of greatest expertise? What have you done over the years to increase your expertise in this domain? Make an estimate of how much time you have put in to achieve your current level of expertise in this domain. What approaches are you following to maintain your current level of expertise or to continue to improve it? What are your goals?
- 6. One useful technique in problem solving is building on the work of others. A person can become skilled at doing library research on what is known about a particular problem. These library research skills are specific to the domain of library science but are domain independent in the sense that they can be applied across a broad range of academic domains. Analyze your strengths and weaknesses in library research skills. How is this area being affected by computers? Are you comfortable using

a computer to do library research? Are your skills equally well developed in searching Internet sources of information?



Education and Training as Resource

In many problem-solving situations that you encounter, you are your main resource. You apply your knowledge and skills to solve a problem or accomplish a task of interest to you.

Much of the education and training needed to solve a problem occurs before you encounter the problem. However, it is often necessary to learn more about a new problem and ways to solve it. "Just in time" education and training are becoming increasingly important.

This chapter focuses on formal and informal education and training. The dividing line between education and training is not clear-cut. Thus, a College of Education trains teachers. The preservice teachers are both educated and trained in their program of study. The goal is to help students gain increased expertise as teachers. We will define *education and training* as instruction designed to enhance the capabilities of the mind and body.

Roles of Education and Training in Intelligence

Some definitions of intelligence seem to assume that it is a fixed and immutable characteristic of a person—fixed at birth. However, it is clear that the potentials that one is born with can be developed by appropriate education, training, diet, and experience. The definition of intelligence used in this book is based on what you can currently do and what you actually do—not on what potentials you were born with.

The definition of intelligence given by Perkins (1995) and discussed in Chapter 3 contains three components: neural intelligence; experiential intelligence; and reflective intelligence. The content of the current chapter relates to improving experiential and reflective intelligence.

Education and training are a lifelong experience. Your mind is naturally inquisitive and always learning. In formal learning situations such as schooling, the learning is directed by a curriculum, teachers, and coaches. In many informal learning situations, your environment serves as the curriculum and you may serve as your own teacher and coach.

You know a great deal about education and training because of the many years you have spent in our formal educational system and the many years you have lived in an informal learning environment. Thus, you have learned to learn both in formal and informal learning situations. That is not to say, however, that you have reached your full potential as a learner; you can certainly increase your level of expertise as a learner. You can develop better research skills, study skills, and learning skills. You can develop better habits of metacognition and other aspects of reflexiveness. You can improve your skills in transferring your new knowledge into the domains that you already know.

Assuming Responsibility for Your Own Learning

Reading a book such as this one requires considerable intelligence and education. Spend a minute browsing through the Index. Notice the range of vocabulary and ideas that the book covers. It takes a great deal of semantic declarative knowledge to be able to read and learn from a book like this. Now, browse through the References and Resources section. Notice how many different people have contributed ideas to this book. As you learn from this book, you are building on the work of a large number of researchers and writers.

You have a considerable level of expertise in learning from academic books and from other formal educational settings. How did you gain this expertise? Have you received specific instruction on how to learn or on how to study?

It turns out that many students have relatively poorly developed study skills. They have not learned many of the ideas given in a study skills book such as Gall, Gall, Jacobsen, and Bullock (1990). Thus, they do not make effective use of their study time. With some training and experience, you can probably improve your learning and retention rates by a significant percentage.

For example, have you developed good note-taking skills that you use both in classes and while reading? Are you good at asking yourself questions about the material you are studying and then answering the questions? Do you draw diagrams and pictures to represent key ideas? Do you think about the meaning of new material relative to what you already know? As you learn new ideas, do you share them with your colleagues and do you try them out at work, at play, and in your studies? Do you reflect over general ideas from what you are learning and how these ideas apply to other domains?

Do you use the same study skills in each course that you take? Suppose that you had to take a written test after you have finished reading and studying this book. What strategies would you use if you knew the test was going to be objective—true/false and multiple choice? What strategies would you follow if you knew the test was going to be short essay? What strategies would you follow if you knew the test was going to be open book and open notes? What strategies would you follow if you knew that the test was going to be a month later, but that you would not have any chance to study during that month?

You may find such talk about tests to be distressing. It places emphasis on passing tests rather than on learning.

What learning strategies would you use if you knew that you would not be tested at all?

You are mature enough to decide for yourself what you want to learn. You are mature enough to determine whether you are learning at a level to satisfy your own needs. The independent, self-sufficient, lifelong learner does not study just to pass objective or essay tests. Rather, the studying is internally (intrinsically) motivated.

Think back over the first few chapters of this book. Consider the problem situation of learning from this book. Do you have ownership? Can you name a number of ideas from the first few chapters that are personally meaningful to you? Are you already practicing them in a variety of settings? Can you name some "ah ha" insights that you have added to your episodic declarative memory as a consequence of studying the first few chapters?

The fundamental question is, are you making use of ideas from this book? Are you able to transfer your book learning into real-world problem solving?

Transfer of Learning

Transfer of learning deals with transferring one's knowledge and skills from one problem-solving situation to another.

Transfer of learning is commonplace and often done without conscious thought. For example, suppose that when you were a child and learning to tie your shoes, all of your shoes had brown, cotton shoe laces. You mastered tying brown, cotton shoe laces. The next year you got new shoes. The new shoes were a little bigger, and they had white, nylon shoe laces. The chances are that you had no trouble whatsoever in transferring your shoe-tying skills to the new larger shoes with the different shoe laces.

Transfer of learning lies at the heart of learning—and hence, of an educational system designed to help students learn. Here is a simplified plan applicable to your education.

- 1. Learn to solve the problems that you frequently encounter or expect to encounter frequently in the future. Make these personal BBRs. Develop a high level of skill that readily transfers to variations of these problems.
- 2. Gain a broad-based general education that includes declarative and procedural knowledge over a wide range of human endeavors. This will help you to relate the various problem situations that you encounter to the collective knowledge of the human race.
- 3. Learn to learn; know your capabilities and limitations as a learner. (Note how this relates to intrapersonal intelligence in Howard Gardner's list.)
- 4. Learn effective ways to make use of steps 1 through 3 in dealing with the nonroutine problems that you encounter and will encounter in the future. That is, gain general skills in the transfer of your learning. Work to improve your general problem-solving skills.

The goal of gaining general skills in the transfer of your learning is easier said than done. Researchers have worked to develop a general theory of transfer of learning—a theory that could help students get better at transfer. This has proven to be a difficult research problem.

At one time, it was common to talk about transfer of learning in terms of near and far transfer. This theory of transfer suggested that some problems and tasks were so nearly alike that transfer of learning occurred easily and naturally. This is called near transfer. Other problems and tasks required more concentrated effort and thinking for transfer to occur. This is called far transfer.

We know that near and far transfer occur. But, what is "near" or "far" varies with the person attempting to do the transfer. We know that far transfer does not readily occur. The difficulty with this theory of near and far transfer is that it does not provide a foundation or a plan for helping a person to get better at transfer.

In recent years, the low-road/high-road theory on transfer of learning, developed by Salomon & Perkins (1988), has proven to be more fruitful. Low-road transfer refers to developing some knowledge/skill to a high level of automaticity. It usually requires a great deal of practice in varying settings. It becomes a stimulus-response type of BBR. Experts in a field have a large repertoire of these BBRs that have been practiced to automaticity and are used without conscious thought.

On the other hand, high-road transfer involves cognitive understanding; purposeful and conscious analysis; mindfulness; and application of strategies that cut across disciplines. In high-road transfer, there is deliberate mindful abstraction of the idea that can transfer, and then conscious and deliberate application of the idea when faced by a problem where the idea may be useful.

For example, we have previously mentioned the strategy of breaking a big problem into smaller components; this is called the top-down strategy. You can learn the name and concept of this strategy. You can practice this strategy in many different domains. You can reflect on the strategy and how it fits you and your way of dealing with the problems you encounter.

Eventually, the strategy becomes part of your repertoire of approaches to problem solving. When you encounter a new problem that is not solved by low-road transfer, you begin to mentally run through your list of strategies useful in highroad transfer. You may decide that breaking the problem into smaller pieces would be an effective strategy to apply.

Two keys to high-road transfer are mindfulness and reflectiveness. View every problem-solving situation as an opportunity to learn. After solving a problem, reflect about what you have learned. Be mindful of ideas that are of potential use in solving other problems.

Of course, there are a wide range of problems that lie between those easily handled by low-road transfer and those that require the careful, conscious, well-reasoned, mindful approaches suggested by high-road transfer. The previous chapter discussed the many years of hard work required to gain a high level of expertise in a domain. To a large extent, this work results in moving many problems from the middle ground in the domain toward the low-road transfer end of the scale. More and more of the problems that you encounter in the domain are quickly and easily solved, almost without conscious thought and effort. An expert has a large number of BBRs within a domain of expertise.

Computers have added new dimensions to the transfer of learning. We have previously mentioned how computers can easily provide you with a greatly expanded collection of BBRs. Of course, it still takes quite a bit of practice before these computer-based BBRs become part of your low-road transfer repertoire. In many cases this is aided by proper design of the human-machine interface. If the software is appropriately designed, it is relatively easy to transfer your learning from one software environment to another.

Overview of Instructional Uses of Computers

Instructional uses of computers are often broken down into three main parts: 1) computer science and computer engineering; 2) computer-as-tool for personal and group productivity; and 3) computer-assisted instruction (CAI). Figure 6.1 shows this three-part division.



Figure 6.1 The three-part division of the instructional uses of computers.

Computer Science and Computer Engineering

The fields of computer science and computer engineering as well as related areas, such as information science and data processing, are well-established domains of study. These computer-related disciplines have existed for many years, and they continue to change rapidly. For example, the Association for Computing Machinery (ACM) is a large professional society that began in 1947. In 1968, the ACM Curriculum '68 specified details of a college undergraduate curriculum of study. These recommendations helped to shape undergraduate programs for studying computer science throughout the country. The ACM has produced major revisions to this initial curriculum several times since then.

At the two-year college, four-year college, and university levels, many computer science and data-processing programs of study have existed for 20 to 25 years and more. There are hundreds of associate- and bachelor-degree programs, and many doctoral programs in computer science and computer engineering. There are hundreds of research journals as well as a great many popular periodicals carrying computer-related articles.

Computer programming is a fundamental component of computer science and computer engineering. A computer program is a detailed step-by-step set of instructions that tells a computer how to accomplish a particular task. While a computer program may be short, some programs are millions of instructions in length. Large teams of programmers, working over a period of years, produce these long and complex programs.

Some people find it useful to draw an analogy between writing computer programs and writing in a natural language, such as English. With a modest amount of education and training, a person can learn to write simple computer programs. This is somewhat akin to being able to write simple paragraphs in English. However, it takes a great deal of education, training, experience, and natural talent to become qualified to play a major role in developing large and complex programs.
Chapter 9 discusses computer programming in more detail. It will help you to decide whether you need to learn a little or a lot about computer programming.

Computer as a Personal and Group Productivity Tool

Many computer tools are designed to be used by an individual to increase productivity. For example, a word processor, spreadsheet, database, and graphics software are all personal productivity tools.

An important trend is emerging in the use of personal productivity tools. The tools and the people using them are being networked together by local area and wide area computer networks. This connectivity is so important that special software, called groupware, has been developed just to enhance group productivity.

With groupware, a team of people who may be located throughout the world can be working together on the same project. They can all be looking at the same computer images, and each can make changes to the images. They can all draw upon the same databases of information. They can communicate with each other through the computer as they work together to solve a problem.

Chapter 8 discusses computer-as-tool in more detail. It looks at a variety of personal productivity tools, some of which may be particularly useful to you.

Computer-Assisted Instruction (CAI)

Computer-assisted instruction (CAI) involves using a computer to help deliver instruction. There are a variety of forms of CAI. Drill and practice tends to be based on behavioral learning theory. The computer presents drill material and provides feedback on the learner's responses. Typically, the goal is to increase the speed and accuracy of performance on materials that students have previously studied.

A second form of CAI is called a tutorial. It includes the presentation of materials to be learned. Many of the newer pieces of tutorial CAI software incorporate ideas from cognitive learning theory. The instruction may make use of text, graphics, sound, color, animation, and video. The interactions between the learner and the computer system may be complex.

A third general type of CAI is called a simulation. Probably you have heard of flight simulators and other types of simulators that have been developed to train airplane pilots, astronauts, ship pilots, and so on. A wide range of simulations have been developed for use in schools. For example, there are simulations for doing chemistry and physics experiments. There are historical and social studies simulations. There are medical simulations that immerse the learner in medical emergencies.

CAI simulations can immerse students in interactive learning environments in which they explore interesting topics and solve challenging problems. The emergence of virtual realities portends a fruitful future for CAI simulations. A virtual reality can be thought of as a simulation that is so lifelike that it causes the user to suspend disbelief when functioning in the simulation.

More About CAI

Learning is assisted and made more efficient by appropriate instruction and by appropriate feedback. That is why we have teachers, coaches, and trainers. An expert teacher or coach can make a significant difference in learning.

That is not to say that people cannot learn on their own. You do so all the time. You serve as your own teacher, coach, and trainer. Your learning may be assisted by books, audiotapes, videotapes, and other media.

CAI falls someplace between having expert humans as teachers and coaches and being your own teacher. Certain aspects of an expert teacher can be incorporated into CAI. The interactive nature of CAI means that you and the expertise in the CAI materials can work together to assist you in your learning.

Historically, early CAI tended to fall into one of two categories. Some of the very first CAI materials were

simulations developed by and/or for the military. These were expensive to develop and required the use of very expensive computer systems, but they were successful. Flight simulators provide an excellent example. The second category of CAI materials were relatively inexpensive drill-and-practice programs. These were based on behavioral learning theory and were often mundane. For example, many different drilland-practice programs were developed to help students learn to do paper-and-pencil arithmetic.

CAI has improved over the years. The improvement is based on continuing research on human learning as well as on the development of better hardware and software. For example, there are now many examples of tutorial software that incorporate both behavioral learning theories and cognitive learning theories. Better hardware and software have led to hypermedia CAI systems that are very impressive. As they are used, data is gathered on how well they work. Analysis and use of this data by the CAI developers leads to incremental improvements on a continuing basis. There is no doubt that CAI will continue to grow in importance as a component of our educational system.

There has been an extensive amount of research on CAI (Kulik, 1994). The results of using CAI vary with the quality of the instructional materials, the area being studied, and the learner. However, the research suggests that on average, students learn faster (perhaps 30% faster) in CAI environments, as compared to conventional instructional environments. Similar research suggests that on average, students learn as well or better in CAI environments, as compared to conventional environments. Because time is one of your most valuable resources for problem solving, the possibility of learning faster is very important.

More and more software packages include some built-in CAI such as a tutorial on how to use the software. Online Help features provide detailed instruction on specific features of the software. You can think of this as "just in time" instruction. These various forms of CAI are adding a new dimension to our formal and informal educational systems.

Activities and Self-Assessment

- 1. Discuss your knowledge of effective study skills. Place particular emphasis on whether you actually make use of the study skills that you know to be effective.
- 2. Research indicates that with experience we improve as problem solvers, but we seem to have difficulty transferring that knowledge to analogous problems in other domains. Give examples from your own life that tend to support or refute this research. Pay special attention to how well experts in one area (for example, research professors) function in other areas (for example, in teaching, in advising students, and so on).
- 3. Michael Jordan was one of the best basketball players in the world and at the peak of his career when he retired from professional basketball. He then tried professional baseball, where his level of success was much more modest. Later he resumed his basketball career and continued to perform at a very high level. Discuss transfer of learning in athletics.
- 4. Make a personal list of strategies that you use when you encounter a new, unfamiliar problem-solving situation. Relate your discussion to the idea of high-road transfer.
- 5. There has been substantial research on CAI. It can be summarized by "CAI works." Discuss the various experiences you have had with CAI. What are your personal feelings about appropriate roles of CAI in formal and informal education?



A Computer System

Earlier chapters in this book discuss how physical and cognitive artifacts can greatly enhance a person's ability to perform. This chapter focuses on a problem-solving environment that consists of hardware, software, and a computer user.



Figure 7.1 Interaction between computer user, hardware, and software.

In Figure 7.1, the computer user may be a person, but may also be a factory machine, a robot, or computerized instrumentation designed to gather data. However, the emphasis in this book is on a human interacting with computer hardware and software to solve problems and accomplish tasks.

The diagram in Figure 7.1 suggests an important concept. This human-machine problem-solving system can get better—become a more powerful resource—by improving the hardware, by improving the software, and/or by improving the human using the machine.

This chapter discusses the hardware and software components of Figure 7.1. The goal is to increase understanding of the computer as a resource in making effective use of your intelligence.

Computer Hardware

Work on the development of electronic digital computers went on in England, Germany, and the United States before and during World War II. In England, a special-purpose computer proved useful in cracking German secret military codes. In Germany, development efforts were hampered by the war and by the low priority placed on the project by the military regime. In the United States, the world's first generalpurpose electronic digital computer, the ENIAC, became operational in December 1945.

By 1950, approximately 20 computers had been constructed on a worldwide basis. Each was a one-of-a-kind, hand-built machine. Then, the era of mass production of computers began.

Improvements in computer hardware came rapidly. By the mid-1950s, there were a number of different brands and models of computers on the market, most costing many hundreds of thousands of dollars. The IBM model 650 was a typical mid-priced machine. This was a vacuum tube machine. A user's manual for this machine discussed its "blinding speed" of about 5,000 arithmetic operations a second. Certainly that was fast compared to calculators that took several seconds to perform a multiplication or a division. The IBM model 650 could calculate the square root of a number in about one-sixth of a second. (That is only slightly faster than

the speed of the square root key on today's \$4 handheld, solar battery-powered calculator!)

Although the transistor was invented in 1947, it took more than 10 years before transistors began to be used in place of vacuum tubes in computers. A transistor is more reliable than a vacuum tube. It uses less power and generates less heat.

The very rapid progress in transistor technology has been the driving force in computer hardware ever since. The individual transistor gave way to the integrated circuit (often called a chip), the large-scale integrated circuit, and the very large-scale integrated circuit. Individual wire-wrapped magnetic cores that could store one bit of information gave way to transistorized memory. Now, a single memory chip, perhaps the size of your fingernail, can store more than 64 million bits of information. It is common to use an 8-bit code for characters. Thus, this chip can store more than 8 million individual characters. This is roughly the equivalent of 8 thick books of information.

Progress in chip technology led to the microcomputer. A medium-priced modern microcomputer may be 10,000 times as fast as the IBM model 650. It might also have 10,000 times the internal memory as well as a disk storage system. Magnetic-disk storage was not available on the early mainframe computers. In the early 1960s, a 5-megabyte disk storage system was considered nearly state of the art. Now, many laptop computers have 160-megabyte or still larger capacity disk drives.

Computer hardware is fast and reliable. Continued rapid progress is occurring in improving the cost-effectiveness of computers. Some of today's laptop computers, costing perhaps \$1,000 to \$3,000, are more powerful than the million-dollar mainframe computers of 20 years ago.

The key idea here is that as the cost-effectiveness of computer hardware grows, it significantly changes the nature of the types of problems that can be solved and the strategies that can be effectively employed. A graphical interface and extensive use of colorful computer graphics are now common on microcomputers. Nowadays, even young children routinely play with computer graphics facilities that are far better than those on the multimillion-dollar mainframe computers of 30 years ago.

By late 1995, approximately 38% of households in the United States had a general-purpose microcomputer; the worldwide-installed base of microcomputers exceeded 150 million machines. Worldwide sales of general-purpose microcomputers in 1995 was approximately 50 million machines. Microcomputers are of growing importance in business, manufacturing, government, and education throughout the world.

Computer Software

A computer program is a detailed step-by-step set of directions that a computer can carry out. Computer software is divided into three main categories.

 Systems software. This is the operating system, the software that interfaces directly with the hardware. Many millions of microcomputers make use of DOS (Disk Operating System) developed by Microsoft Corporation. This is an operating system specifically designed for IBM and IBM-compatible microcomputers with disk drives. A new Microsoft Corporation operating system named Windows '95 came into mass distribution in the fall of 1995.

The operating system for the Macintosh goes by the name "system" followed by a version number, as in System 6 and System 7.

- 2. Programming languages. Hundreds of these have been developed. Examples include FORTRAN, COBOL, BASIC, Logo, Pascal, C, and Ada. For each such programming language (source language) there are computer programs that translate from the source language into the machine language of some type of computer. For example, there are computer programs that translate from C into the machine language of IBM compatibles, Macintoshes, and so on.
- 3. Applications software. These are computer programs specifically designed to help solve some particular

category of problems such as word-processing, database, spreadsheet, and graphics programs. Perhaps you are familiar with programs that can solve a variety of math problems. There are thousands of different pieces of applications software that are commercially available.

Historically, these three categories of software were distinct. Now, however, the dividing lines are blurred.



Figure 7.2 The blurring dividing lines between categories of software.

One example of this blurring is that, increasingly, computer programming languages are being built into applications programs. This has been true for database software for quite awhile. Hypermedia software, such as *HyperCard* and *HyperStudio*, include built-in programming languages. Microsoft Corporation uses BASIC as the programming language in a number of its applications. We will discuss the problem-solving implications of this trend more in Chapter 9, which focuses on computer programming.

Hardware and Software—A Computer System

As indicated in the previous section, there are three general categories of computer software: operating systems, programming languages, and applications. Figure 7.3 presents a simplified view of how these types of software interact with

each other, with hardware, and with a computer user. An application program is written in a programming language. The programming language and operating system are designed so that they can interact in an appropriate manner. The hardware actually executes (carries out) the instructions given in the application program.



Figure 7.3 Computer hardware, software, and user.

Of course, it is possible for an ordinary person to make modifications to hardware, such as adding a new hard drive, modem, scanner, or printer. A person can make modifications and/or additions to an operating system. And, even young students can learn to write computer programs. Thus, a more accurate picture appears in Figure 7.4.



Figure 7.4 The computer user working directly with both hardware and software.

Figure 7.4 suggests that the computer-and-human system is complex. Indeed it is! Thus, considerable efforts are underway to simplify this situation.

Human-Machine Interface

The human-machine interface for a computer system is a combination of hardware and software. In the earliest history of computers, this interface consisted of toggle switches and plug-in wires that could be used to connect various components of a computer. In essence, computer programming consisted of rewiring the machine. Needless to say, it was not easy to learn to write these types of computer programs. Mistakes were frequent. Often the mistakes were difficult to detect and to correct.

Gradual progress occurred in improving the humanmachine interface. One major breakthrough was the realization that it was possible to store a computer program in the computer memory, rather than having to rewire the computer to carry out a particular sequence of steps. Other improvements in hardware included punched paper tape and punch cards. Programs and data were stored as patterns of holes in paper tape or cards.

Improvements in software included programming languages such as FORTRAN and COBOL. Initially, FORTRAN was designed to be used by people with graduate degrees in engineering, math, and physics. Gradually, it became apparent that college freshmen—and even high-school students—could learn to use this programming language. By the late 1950s, computer use had begun to creep into a few elementary and secondary schools.

Eventually, time-shared computing was developed. A timeshared computer system is designed so that it can serve a number of users—each with their own computer terminal—at the same time. They could be located many miles from the computer, but still interact with it just as easily as people who were in the building that housed the computer. This was a major improvement in the human-machine interface. It was the beginning of the computer networks that are so commonplace today.

Another major improvement was the development of the personal computer during the 1970s. People could have complete control of their own personal computers. They could customize the hardware and software to fit their own particular needs. Rapid improvements in the cost-effectiveness of microcomputer hardware have made possible the implementation of a variety of improvements in the humanmicrocomputer interface.

Quite a bit of the processing power in a modern microcomputer is used in the human-machine interface. The idea is to make computers easy to learn and easy to use. For example, a mouse is relatively easy to learn to use. It is easy to learn to recognize and remember the meaning of various icons (pictures) on a computer screen. The graphical user interface (GUI, pronounced "gooey") that makes use of icons and a mouse is far easier to learn to use and easier to use than the keyboard-based interface of all the early microcomputers.

Other components of a modern human-microcomputer interface make use of sound and color. Although voice and other sound output have been available for quite awhile, voice input is just now coming into widespread use. It seems clear that, eventually, voice input will replace keyboarding as the most common form of input that people use with computers.

One of the key ideas in designing a human-machine interface is to facilitate transfer of learning. For example, young children have considerable skill in recognizing pictures, in pointing, grasping, and so on. These skills form the rudiments of the GUI. Thus, even young children can easily learn to interact with a GUI—because of transfer of learning.

Much of the early commercial success of the Macintosh computer, developed by Apple Corporation, was due to a combination of its GUI and the consistency of human-user interface across many different pieces of applications software. The knowledge and skills gained in learning one piece of Macintosh software tend to transfer to learning to use other pieces of Macintosh software.

The design and implementation of good human-machine interfaces for a computer system adds a new dimension to the transfer of learning. If the interface is well designed, it is easy to learn to use. Moreover, a good interface is designed to accommodate the strengths and weaknesses of humans. It is fault tolerant. Humans often make mental slips—for example, saying one word when they mean another, or making a keyboarding error. In a poorly designed human-machine interface, such a tiny slip might destroy many hours of work. A superior interface is tolerant of such slips and makes it very difficult for the user to cause a disaster.

Artificial Intelligence

The early human-machine interfaces for computers relied almost completely on the intelligent performance of the human user. One way to think about improvements in the human-machine interface is in terms of the machine becoming "smarter" or more intelligent.

For example, you can think of a computer that can read and interpret instructions represented as holes in a punch card as being more intelligent than a machine that must be rewired to change what it can do. Even such a small step-up in machine intelligence constitutes a major improvement in the human-machine interface.

In terms of human-machine interface, computers are a lot more intelligent now than they were a few decades ago. Moreover, a whole domain of study and research called *artificial intelligence* (AI) has developed. In some parts of the world, this domain is called machine intelligence. The terms artificial intelligence and machine intelligence suggest that it is a different kind of intelligence—not a human type.

Many different research teams throughout the world are working on various aspects of AI. These researchers have experienced considerable success. For example, you probably make use of a spell checker when you write using a word processor. Current spell checkers are good at detecting errors and at suggesting corrections. Some are now able to learn the patterns of keyboarding and spelling errors that a particular user makes and, thus, improve their spell-checking performance. Spell checkers provide an example of some of the progress that has occurred in AI.

The field of AI has had a long and checkered history. Early researchers in the field of AI were confident that "soon" (typically they felt that it would take about 10 years from the time that they were presenting their findings) there would be computers more intelligent than humans. Such predictions have been made by a variety of leading researchers many different times over the last 30 years! For a summary of current research in various aspects of AI, refer to *Communications of the ACM* (March and July, 1994).

However, one of the things that these researchers have concluded is that humans easily and routinely accomplish creatively intelligent problem-solving tasks that are very difficult relative to the capabilities of the very best of the artificially intelligent computer systems. To put it bluntly, humans are far more creatively intelligent than today's best AI systems.

To get some insight into this, consider the following paragraph:

I want you too go two the store four me. Their, I want you to by to pounds of coffe. I need them rite away. So, as you come hone form the store, please hury.

If you run this paragraph through a spell checker, it will catch the misspelling of coffee and hurry. This leaves the paragraph:

> I want you too go two the store four me. Their, I want you to by to pounds of coffee. I need them rite away. So, as you come hone form the store, please hurry.

The grammar checker that is built into the word processor being used by the author of this book produced a suggestion for each sentence.

🛛 📰 📰 Grammar 🗐		
Sentence:		lanore
I want you too go two the store four me.	Ŷ	Change
	ন্ট	Next Sentence
Suggestions:		Ignore Rule
Consider to or two instead of too.	슌	Close
		Explain
	<u></u>	Options

Figure 7.5 Analysis of first sentence.

쇼	
÷	Next Sentence
	[Ignore Rule]
쇼	Close)
	Explain
<u>₹</u>	Options
	다 다 다

Figure 7.6 Analysis of second sentence.

🗌 🔤 Grammar 🔤		
Sentence:	_ (lanore
So, as you come hone form the store, please hurry.	요 '	
	Ţ	Next Sentence
Suggestions:		Ignore Rule
Consider stores instead of store.	Ŷ	Close
Consider pleases instead of please .		Explain
	£	Options

Figure 7.7 Analysis of third sentence.

As you can see, the grammar checker is not making suggestions based on a human-like understanding of the paragraph! Implementation of the suggested changes produces the following paragraph:

> I want you to go two the store four me. There, I want you to by to pounds of coffee. I need them rite away. So, as you come hone form the stores, please hurry.

It takes a high level of understanding of the English language to detect and correct the errors in this paragraph. It is difficult to develop computer programs that actually have the necessary level of human-type understanding.

This grammar-checking example illustrates some of the difficulties of developing voice input systems for a computer. Voice input is a problem that researchers in AI have been working on for many years. It has proven to be a very difficult problem. We now have commercially available voice input systems in which the computer can recognize spoken words and translate them into written text. However, in such voice input systems the computer has little or no understanding of the meaning of the words. Errors are frequent. The human user must detect and correct the errors if a high level of accuracy is desired.

The problem of natural-language translation is also very complex. Significant progress has occurred in computer translation from one natural language to another. However, the quality of such translations still leaves much to be desired because the computer programs have no innate understanding of the meaning of what is being translated.

Grammar-checking, voice input, and language translation are all closely related. They all illustrate problems that humans can learn to solve but that today's computers don't solve very well. Current artificially intelligent computer systems are not good at tasks that require human understanding. The best of current AI software has little understanding of what it means to be a human being.

This is a very important idea. While computers can do many things far better than humans, the opposite is also

true—humans can do many things far better than computers. If great speed, accuracy, and attention to the smallest details are needed, computers are apt to be the answer. If understanding of natural language and of human beings is needed, computers have a very long way to go.

Expert Systems

One component of AI that has shown significant progress is called *expert systems*. The basic idea is to develop a computer program, called an expert system, that has a significant level of expertise within a particular domain. For example, consider the domain of interpreting electrocardiographs (graphs of electrical signals from the heart). Software that can do this as well as human experts was developed well over a decade ago. It has come into widespread use.

Expert systems are rule-based systems. They are developed by a very careful search for the rules that a human expert (or a number of human experts) use in solving the problems in a narrow domain. The rules are stated in an if-then format. For example, in a medical diagnosis, test for Condition A. If it exists, apply Treatment Q. If it is unclear whether Condition A exists, test for Condition B, and so on.

One of the unforeseen results of research on expert systems is that in many narrow domains, it has been found that a high level of performance requires only a few hundred or perhaps a thousand rules. It is hard to believe that human expertise in some narrow areas of specialization can be captured by such a small set of rules.

Once an initial working version of an expert system has been developed for a particular problem domain, the expert system can be improved. This can be done by analyzing its performance. When it performs poorly—for example, by making an outright error—the rules leading to this error can be examined. The rules can be modified, or additional rules can be added. By this approach, the performance of an expert system can be substantially improved over a period of time. This type of improvement process can be thought of as humans "teaching" the computer to be more of an expert. There are now thousands of expert systems in everyday use, each with a high level of performance in a very narrow domain. Uses vary from medical diagnosis to prospecting for minerals to scientific research to processing loan applications at a bank. Such expert systems are useless when presented with problems outside of their narrow domains. They may (seemingly) produce answers to a problem, but the answers are totally unrelated to the problem. For example, you might interact with a medical diagnostic expert system, describing certain characteristics of your car to it. Your description might include talking about rust-colored spots and a feeling of sluggishness. The medical system might conclude that your car has measles and is constipated!

This is an important idea. When you view yourself as a problem solver, you know that you have strengths and weaknesses. You can judge the usefulness of your ideas in a particular problem domain. You are apt to be able to tell when you are making major mistakes. These statements do not hold true for computer-based expert systems. Today's expert systems are not self-aware, like a human is. The expert systems have no sense of their capabilities and limitations. They are not able to detect when they are making major errors.

It seems clear that expert systems bring a new dimension to problem solving. One way to think about an expert system is as a BBR. An expert system BBR, by itself, does not know when it should be used. It does not know when it makes a major mistake. But a human can learn both when to apply a BBR and how to interpret the results. The combination of human and machine working together can be a powerful aid to problem solving.

Expert systems have some strengths, and they have some weaknesses. With appropriate education and training, a human can learn to make use of the strengths of an expert system and overcome some of its weaknesses. The combination of human and machine working together to solve problems can be far more powerful than either one working alone.

Agents

In recent years, a great deal of research work in artificial intelligence has focused on developing agents—think of an agent as a computer-based intelligent, personal secretary—that can help enhance the human-machine interface.

An agent might be quite specialized. For example, you might have an agent that screens your incoming telephone calls. It would need to have the intelligence to decide how important a call was. The most important calls might be passed on directly to you, even when you are in an important meeting. Some calls would be rejected outright

As another example, suppose that you are particularly interested in European art of the 18th century. You might have an agent that "reads" all periodicals that are being published online. Its goal is to find every article that might relate to European art of the 18th century. You might want this agent to produce brief abstracts of the articles and to prioritize the articles in terms of likely relevance to your interests.

Such agents are being created by researchers in artificial intelligence. Eventually, you will make routine use of many of these agents.

Intelligent Digital Connectivity

Previous sections in this book have discussed networking groups of people, for example, by use of telecommunications systems and groupware software. Artificial intelligence may be incorporated into the computer network and the groupware. The overall concept of such a system is sometimes referred to as intelligent digital connectivity.

Intelligent digital connectivity is several things. It is digitized information—text, graphics, color, sound, and video. It is an Information Superhighway that links people and machines. It is computer systems that aid in storing, processing, and retrieving digitized information. It is intellectual technology that is being produced by researchers throughout the world. It is the progress that is occurring in the computer and cognitive sciences. It is artificial intelligence—for example, expert systems and intelligent agents—that is emerging as an important part of the humanmachine interface.

Intelligent digital connectivity provides the foundation for a new type of intelligence that is called *distributed intelligence*. Distributed intelligence is the combined intelligence of people, machines, and the networks connecting them.

It is interesting to think of the overall capabilities and intelligence of such a distributed intelligence system. Each person in the system brings in various types and levels of human intelligence. The individual computers being used have varying levels of machine intelligence. Also, each person may well have personalized his/her computer by modifications to the hardware and software. Next, there is the intelligence built into the networking software and the groupware. Finally, the whole system may provide access to large amounts of computing power and to large numbers of databases.

The "intelligence" of a distributed intelligence system varies with its components. This brings a new dimension to problem solving. Suppose that a problem to be solved or a task to be accomplished is clearly beyond the capabilities of one person. A distributed intelligence system is needed. But, each of the components has a cost, and there is a wide range of components (including people) to select from. How does one build a cost-effective distributed intelligence system to fit the needs of a particular problem?

The flexibility involved in creating and disbanding a distributed intelligence system can be contrasted with the flexibility in industrial manufacturing. A manufacturing facility tends to have limited flexibility. It is not easily remodeled (retooled) and/or moved to a different site. Thus, it is clear that the idea of a distributed intelligence system brings new challenges to business and industry management. Many of today's business leaders lack the knowledge, skills, and experience to effectively deal with the flexibility that distributed intelligence brings to solving problems and accomplishing tasks.

The Future of Hardware, Software, and Connectivity

The personal microcomputer is now about 20 years old. During this time the personal microcomputer has progressed from being a toy to becoming an indispensable tool for tens of millions of people. Microcomputers have made this progress because of sustained rapid improvement in both hardware and software. Hardware improvements include major increases in processor speed, high-speed memory, and disk memory. Software improvements include major improvements in the human-machine interface and the development of a wide range of personal and group productivity tools.

The pace of change shows no signs of lessening. During the next decade, the pace of change will be similar to what it was in the past decade. In terms of hardware, this has been an exponential rate of change. Thus, over a span of about four or five years, the best of the new microcomputers may have four times the speed and four times the memory of the machines that they are replacing.

As previously mentioned, quite a bit of the computing power of a microcomputer is used in the human-machine interface. Voice input is a natural, easy to learn, and easy to use type of interface. However, it requires both very sophisticated software and an immense amount of computing power. Because of the continuing progress in hardware and software, voice input will likely be commonplace in 10 years.

Connectivity is growing at a fast pace. You are familiar with cellular telephones, earth satellites that broadcast television directly to peoples' homes, and other types of wireless communication systems. You are also familiar with fiber optics that allow interactive television. The combination of the wireless and the hard-wired forms of connectivity are tying the world together into a worldwide computer network. Eventually, a computer workstation will be a communication center, allowing quick access to major sources of information and to people throughout the world.

Activities and Self-Assessment

- 1. Discuss why it is so difficult to write a computer program that can detect homonym errors in writing and automatically correct them. Give some examples that illustrate the difficulties.
- 2. Discuss why the language translation problem is so difficult. If you know two languages, give examples based on translating from one to the other. Otherwise, give examples of ambiguous meaning in the language that you speak.
- 3. Select a narrow problem-solving domain where you have a relatively high level of expertise. Give some examples of if-then rules that capture some of your expertise.
- 4. A handheld calculator can be thought of as a machine that has a high level of intelligence in a very narrow domain—a domain of doing four different types of calculations. A calculator does not know when to multiply or divide. Discuss the human-calculator system. What are the strengths and weaknesses of each component? How can the overall system be improved?
- 5. Go back to the definition of intelligence given in Chapter 3. Compare and contrast this definition of intelligence with current progress in AI (machine intelligence). Use this analysis as a starting point for creating a definition of intelligence for distributed intelligence systems.
- 6. Select a problem-solving domain of interest to you. Analyze it from the point of view of distributed intelligence systems. How could your unique interests, educations, and experience fill an effective, complementary role in a distributed intelligence system designed to work in that domain?



Personal Productivity Tools

Reading, writing, arithmetic, speaking, and listening are all considered basic skills. These skills are useful over a wide range of problems that people encounter. Now use of computer-based personal productivity tools is emerging as a new standard in education.

Many of the productivity tools are generic—that is, they are useful over many different domains. Examples include a word processor, database, and graphics. Others are highly specialized. Software for writing and publishing music falls into this category. This chapter focuses mainly on generic computer productivity tools.

Process Writing

Many of the fundamental ideas regarding personal productivity tools can be illustrated using a word processor. It is likely that you have used a word processor; thus, a number of the ideas given here will be familiar to you.

To begin, you know that there is a considerable difference between being able to use a writing tool—be it pencil and paper or a word processor—and being able to write. The tool, by itself, does not make you into a writer. Writing is a process designed to produce a document that communicates a message. Typically, the production of a written product goes through several steps that, collectively, are known as *process writing*.

- 1. Conception of ideas and development of these ideas. This may involve brainstorming, doodling, making brief notes, and a lot of thinking.
- 2. Development of an initial draft. This involves getting the conceptualized ideas into words.
- 3. Obtaining and making use of feedback. Feedback from oneself and others is used to produce revised versions of the initial draft. Often this involves repeated cycling back to step 1 and/or 2.
- 4. Polishing the final draft for publication. This includes final cleanup on spelling and grammar. Nowadays, it often includes formatting the materials in a professional manner using desktop publishing techniques.

Computers can play an important role in each of these four steps of process writing. While the first step may be primarily mental, there are a variety of pieces of software designed to aid in jotting down ideas and organizing these ideas. In addition, most modern word processors include an outliner. This makes it easy to get rough draft ideas into the machine and to reorganize them as needed.

People who have learned touch keyboarding can keyboard far faster than they can handwrite. Many people compose at the keyboard. This is particularly useful in process writing because the computer is such a useful aid for revision. It is often argued that the main value of a word processor in process writing is that it is an effective revision tool. Many writing instructors feel that the key to high quality writing is "revise, revise, revise."

Revision is an important idea in problem solving as well. There are many problem-solving situations in which one can develop a proposed solution and then get feedback from oneself and others on the quality of the proposed solution. The feedback is then used in doing revisions to the proposed solution. The feedback and revision cycle continues until a satisfactory solution is obtained.

Desktop publishing has become a major industry. All wordprocessing software contains provisions for producing a final document that is nicely laid out. Professional-level desktoppublishing software contains a wide range of aids to produce professional-looking final products. Such documents often make use of graphics and color. They may be laid out in columns, make use of a range of type styles and sizes, and be designed to help convey their messages.

A person can learn to keyboard in a hunt-and-peck mode with just a minute or so of instruction. Young children can learn such keyboarding more easily than they can learn to form letters using pencil and paper. Similarly, it takes only a few minutes of instruction to learn how to use a word processor in a hunt-and-peck mode. However, this low level of word-processor use is only a modest aid to productivity in writing. It is too slow and it does not take advantage of the powerful writing aids that are built into a modern word processor.

You need four things for a word processor to be a useful personal process writing aid:

1. Keyboarding skills. You need not be a touch typist, even though it is helpful. Many professional writers are not touch typists. They look at the keyboard and they use only a couple of fingers from each hand as they keyboard. However, they know where the keys are and they have considerable speed. The skills that they have developed are adequate to fit their needs.

To learn to make appropriate use of all fingers on both hands and to develop speeds in the 20-to-30-words-perminute range takes a considerable period of instruction and practice. For many people, it takes about 8 to 10 weeks, spending about an hour a day. A speed in the 20to-30-words-per-minute range is a good indicator that the learning is embedded in one's procedural memory. This skill is not lost if one does no keyboarding over an extended period of time, such as a long summer vacation. Most people can learn to keyboard at a speed that is two to three times their handwriting speed. In addition, most people find that their keyboarding results are far more legible than their handwriting.

- 2. Word-processing skills. For example, how do you do a cut and paste? How do you do a search and replace? How do you use a spell checker and a thesaurus? How do you create tables, alphabetize a list, or automate the production of an index and table of contents? The manual for a modern word processor may be many hundreds of pages in length.
- 3. Word processor-assisted writing skills. In essence, paper and pencil provide a linear writing environment where it is difficult to correct errors and even more difficult to make significant overall revisions to a document. Interchanging the order of two paragraphs requires recopying an entire page or more. The word-processing environment is different. It takes a lot of training and experience to unlearn some of the linear and restrictive writing habits that are required when working with pencil and paper, and to learn to take advantage of the power of a word processor.
- 4. Desktop publication knowledge and skills. Before the development of desktop publication, many people made a living in the design, layout, and typesetting of print materials. Both design and typesetting were skilled professions. Now, desktop publication tools have made the writer more and more responsible for design and "typesetting."

In summary, it takes a lot of education and training to learn to make effective use of a word processor. Most people find that the investment in time and effort is well worth it. Part of this education and training can occur as one is making use of the word-processor tool to accomplish tasks at work and in one's studies. There is certainly no need to fully master a word processor before beginning to use one as an aid to personal productivity. There is no need to fully master the design of effective written communications before beginning to use design techniques. At the same time, it is easy to see that there is a wide range of levels of expertise in using a word processor. The person who learns on the job—and who continues to learn as a lifelong activity—will continue to grow in expertise.

There are many other productivity tools that are useful over a wide range of domains. Each shares many characteristics with the word processor. Each can be learned at a modest level with minimum effort and each is open ended—that is, each provides the possibility of a lifetime of increasing expertise.

Generic Computer Productivity Tools

We use the term *generic tool* to describe a software tool that is applicable over a wide range of different disciplines. The word processor and desktop-publishing tools discussed in the previous section are examples of generic tools. To make effective use of a generic tool, you need to know both the tool and the domain of application. You already have a reasonable level of expertise in many different domains. Thus, as you learn to use one of these generic tools, you will find that it is relatively easy to apply the tool to your areas of expertise.

There are many software tools that might be considered generic. The following list has been arranged in alphabetical order.

- Computer-assisted design (CAD). Notice how this computer application relates to spatial intelligence in the Howard Gardner list of multiple intelligences. CAD software is used to do architectural and engineering drawings of products that are to be constructed. A CAD system can be used in the design of all sorts of products. Such software is used in place of the ruler, compass, protractor, and other tools formerly used by the draftsperson.
- Database. A database is an organized collection of information, often specific to one particular topic. A telephone book is a database of names, addresses, and telephone numbers. A computerized database is much easier to edit (add entries, make corrections, delete entries) than a printed database. A computerized

database is designed to make it easy to locate needed information. It is also designed to make it easy to sort information into a desired format or to prepare reports based on parts of the information. For example, a customer database could be used to produce a report on all customers located in Ohio or Pennsylvania who spent more than \$100 during the past three months.

- Desktop presentation (to accompany oral presentations). The overhead projector, filmstrip projector, movie projector, tape recorder, and video projector have gradually merged into a computer-based system. Material to be presented is stored on computer disk in digital form and edited using the computer. The presenter then uses the desktop-presentation system interactively when making the oral presentation.
- Desktop publication. A computer system is used to store, edit, design, and lay out the materials that are to be published in printed form. Output may be to a printer, to film used to make plates to go on a printing press, or directly to a printing press.
- Graphics (paint and draw programs). Notice how this computer application relates to spatial intelligence in the Howard Gardner list of multiple intelligences. A paint program has some of the characteristics of a set of painting tools, while a draw program has some of the characteristics of a set of drawing tools. Taken together, these tools can be used to accomplish a wide range of graphic artist tasks. Moreover, photographs can be digitized and then edited using computer graphics capabilities. Similarly, individual frames of video material can be edited using computer graphics facilities. The graphics that are produced can be used in a word-processing document, in desktop presentation, or in other types of computer applications.
- Graphing (for graphing data and functions). Numerical data is easily converted to a wide range of different types of graphs, such as bar graph, line graph, pie chart, and so on. Mathematical functions can be

represented graphically. For example, a threedimensional mathematical surface can be represented on the computer screen and then rotated to allow viewing from different perspectives.

- Groupware. Notice how this computer application relates to interpersonal intelligence in the Howard Gardner list of multiple intelligences. This software combines telecommunications with personal productivity tools. It is designed to facilitate a group of people from different locations in working jointly, both simultaneously and individually, on a computerbased project. Increasingly, groupware will include provisions for the users to talk to each other and see each other as they work together.
- Hypermedia. A hypermedia document is designed to be used interactively by a computer user. It may combine text, sound, graphics, color, and video in a nonlinear fashion. The nonlinearity and interactivity mean that "reading" a hypermedia document requires the use of a computer. Increasingly, our educational system is working to have students become "reading and writing" hypermedia literate.
- Math systems. Notice how this computer application relates to logical-mathematical intelligence in the Howard Gardner list of multiple intelligences. There are a number of comprehensive software packages that can solve a huge range of math problems. Such software can solve the types of problems that students struggle over in algebra, calculus, and other math courses. The use of such software in these courses leads to a drastic change in the nature of the courses. And, of course, it leads to a drastic change in the ability of students to actually solve the types of problems they are studying in the courses.
- Spreadsheet. Notice how this computer application relates to logical-mathematical intelligence in the Howard Gardner list of multiple intelligences. A spreadsheet is designed to aid in doing bookkeeping, accounting, and modeling of business problems. It can also be used in other computational situations in which one works with a table of numbers and

formulas. A key feature is that the computer system can automatically rework all of the computations represented in the table whenever you make a change to any of the numbers or formulas.

- Telecommunications (for communication between people and databases). Notice how this computer application relates to interpersonal intelligence in the Howard Gardner list of multiple intelligences. Telecommunications is the electronic link between people, computers, and other machines. This may be via a local area network, perhaps just connecting people, computers, and machines that are all in one building. It may also be a worldwide connection, using local and long distance telephone lines, satellites, microwave systems, and fiber optics. Intelligent, digital connectivity is having a major impact on the societies and people of our planet.
- Word processor. Notice how this computer application relates to linguistic intelligence in the Howard Gardner list of multiple intelligences. A word processor is software designed to aid in writing. A modern word processor contains a number of features, such as a spell checker, thesaurus, graphics, and graphing, that may be of use to a writer. There is no clear dividing line between a word processor and desktop-publishing software.

The common features of the tools in the generic tools list are that they are applicable over a wide range of different disciplines, and that they are designed so that an ordinary person can develop a useful level of skill in using each of the tools. Many of these tools are now incorporated into the curriculum of our K-12 educational system. The definition of *computer literacy* that is gradually becoming commonplace is that it means a functional working knowledge of a number of the generic tools, at a level consistent with one's overall education. By functional, we mean both a knowledge of the mechanics of how to use the tools and considerable knowledge and skills in using the tools to solve problems and accomplish tasks.

Although a generic computer-based personal productivity tool can be used in many different disciplines, each tool is

oriented towards representing and solving certain somewhat specific types of problems. For example, consider a database application package. It is specifically designed for the representation and solution of database problems. There is a great deal of knowledge about the representation and solution of database problems that is built into a database application package.

This is not to say that database problems cannot be represented using other application packages. For example, a database of information can be represented in a word processor or in a spreadsheet. However, neither a word processor nor a spreadsheet contains the full range of built-in aids to solving database problems.

There are many different generic tools. Part of the process of learning to make effective use of these tools is learning to classify problems according to the type of computer tools that may be most useful. This requires gaining some knowledge and experience with a wide range of these tools.

Goals in Learning Generic Tools

When learning a specific generic tool, there are two goals. First, there is the goal of learning to use the tool. Second, there is the goal of actually using the tool to represent and solve problems and accomplish tasks. Sometimes instruction in the use of a tool is completely or almost completely divorced from actually using the tool to solve problems. With the generic computer tools, this is a mistake. Each generic tool requires time and effort to learn. However, each can be used at a rudimentary level with only a modest amount of training. Thus, with just a little training you can learn to use database, graphics, spreadsheet, and telecommunications software at a level that is useful to you. An appropriate approach to this learning is illustrated in the following diagram.



Figure 8.1 Continually changing our role as learner and user.

The general idea is that you cycle back and forth from being a learner (1) to being a user of your learning (2). A good teacher or a good coach can help you perceive that you need to learn more about a tool. They can do this by presenting you with tasks that are not easily accomplished with your current level of knowledge of the tool. Alternatively and/or in addition, personal reflectivity can be a driving force. You can detect when you don't have the knowledge and skills to efficiently accomplish the tasks that you want to accomplish.

You have probably heard the expression, "Don't saw with a dull saw." This refers to the idea of having inadequate knowledge and skills in using the tools needed to solve a problem or accomplish a task. As you learn to make use of generic tools, you are apt to frequently encounter the dull-saw phenomenon. For example, this may be a situation in which you find yourself doing repetitious and tedious tasks—doing things that you feel the machine ought to be able to do for you. However, you have not learned enough to have the machine do more of the work. That is the time to sharpen your saw!

Tools Can Change a Domain

In traditional high-school geometry, students learn to use a compass, protractor, and straight edge to do a number of different geometric constructions. The domain of geometry and the use of these tools are interwoven. There is a 2,000-year-old history of linking plane geometry to these compass, protractor, and straight edge tools. For many centuries, the tools of geometry have been used in architectural and engineering drawing, and in other situations where precise drawing is necessary.

Now, computer tools have been developed that are far more powerful and versatile than the compass, protractor, and straight edge. Computer-assisted design (CAD) has become a common tool of the draftsperson and the engineer. The drawings done using CAD tools can be interfaced with computer-assisted manufacturing (CAM) software and machine tools. The combination of CAD/CAM has led to major changes in the design and manufacturing of products.

This raises a very important question. What should students be learning in school? If a domain (such as plane geometry) is highly dependent on a certain set of tools (compass, protractor, and straight edge), what should happen when better tools are developed? This question is similar to the question about using a calculator rather than learning to do pencil-and-paper arithmetic.

This type of question does not have a simple answer. Computer tools are expensive and not available to everybody. Part of the purpose of a plane geometry course is historical and cultural. However, our education system is gradually coming to grips with the use of four-function calculators, graphing calculators, generic software, and special-purpose software in the math curriculum. Geometry, and the entire math curriculum, will gradually be changed as these tools are more thoroughly integrated into the curriculum.

Special-Purpose Computer Tools

The generic computer tools are applicable across many different domains. A number of domains have developed

software tools that are highly specific to the needs of professionals in these domains. For example, accountants are faced with the task of auditing the accounting work that others have done. Software designed to assist in this auditing process is specific to the needs of accountants.

The domain of computer chip design also requires specialized software. Some of today's large-scale integrated chips contain millions of individual transistors and other components. Both the design and the manufacture of such chips are highly dependent on software written for that particular purpose.

Another example is provided in the domain of music. A variety of computer tools contain some built-in music capabilities. However, these are highly limited relative to the computer music tools that have been developed for use by professionals in this domain. Computer-based music synthesizers have become an everyday tool of many music composers and performers.

Still other examples can be found in each domain in which the computer has been developed as a powerful aid to personal productivity. In many cases, such as in accounting, the design of computer chips, and many areas of science, the computer tool and the domain are becoming inextricably interwoven. Students no longer study the domain without simultaneously studying the use of the computer tool as an aid to solving the problems of the domain.

Integrated Packages and Suites

Historically, each computer tool has tended to focus on the problems of a narrow domain. For example, the initial word processors did not include a spell checker, a choice of type faces, or a choice of type sizes. In essence, the initial word processors were merely typewriters with memory. This is—only a small part of the domain of effective written communication.

Gradually this changed. Developers of word-processing software have explored the question of what tools are useful to a writer trying to produce an effective written communication, and how to build these tools into a word processor. This has led to the inclusion of tools such as a spell checker and a thesaurus. It has also led to the inclusion of tools to graph data, produce graphics, and incorporate a variety of type faces.

Even this expansion of the features of a word processor has not been adequate to fully meet the needs of a writer. A writer may well need to work with accounting data; thus, the writer may want to use a spreadsheet and incorporate it into a report. A writer may need to work with extensive databases and incorporate a database report into a document. A writer may want to telecommunicate a document and/or incorporate information obtained through use of telecommunication facilities. A writer may want to make an oral presentation, drawing on parts of a written document.

These diverse needs for aiding effective written communication initially led to the development of integrated software packages. A single software package, such as *ClarisWorks* or *Microsoft Works*, contains a number of generic tools, all designed to work together. *ClarisWorks* and *Microsoft Works* are each available for both IBM-compatible and Macintosh computers. The typical integrated package, such as the two just mentioned, includes a word processor, database, spreadsheet, graphics, and telecommunications. Such integrated packages have proven very popular in small businesses and in education.

Recent years have seen the development of a competitor for integrated packages. A *suite* is a collection of individual tools that are designed to function well in an integrated fashion. A company that produces a word processor, spreadsheet, and database may design all of these to interact seamlessly and easily with each other. From the user's point of view, the suite of software performs just as if it were an integrated package. But, each component is very powerful—a professional-level tool.

It is evident that the trend toward the development of suites of software tools will continue. Eventually, one can expect that the full range of software products from a company will interact in a seamless fashion, just as if they were one single tool. But the tool will be very complex. The various components of the tool will be continually changing. The components will undergo continual improvement, and new components will be added. Needless to say, this presents an interesting challenge to the humans who need to use the suite of tools.

Computer Tools Can Create a New Domain

A hypermedia document can contain text, sound, graphics, color, and video. These various media can be combined in a nonlinear fashion to represent and help communicate a message. Also, most hypermedia software packages contain some sort of built-in computer programming language. (Computer programming is discussed in the next chapter.)

Typically, a hypermedia document is designed to be "read" using a computer. The user interacts with the document. The document is nonlinear and different readers of the document are apt to pursue different paths as they use the document. Thus, hypermedia is a new type of communication medium. While a hypermedia document can be printed on paper or viewed as a videotape, this loses the interactivity and nonlinearity that make hypermedia unique.

It is not easy to learn to communicate effectively in a hypermedia environment. First, there is the difficulty of learning to make effective use of the various media. Think about your current level of writing skills and how long it has taken you to achieve your current level of expertise. Then think about each of the other media that are combined in a hypermedia environment. Are you a skilled graphic artist? How are your video-production and editing skills? Do you know how to make effective use of color and sound in a computer document?

Finally, think about learning to develop messages that are designed to be read in an interactive, nonlinear fashion. These are different from the linear written documents and video that we have grown up with. It should be evident that it is a major challenge to develop a high level of expertise in creating hypermedia documents.
It seems clear that hypermedia is part of the communications wave of the future. Many schools are now working toward having all of their students become hypermedia literate, even at the elementary-school level.

Activities and Self-Assessment

- 1. Select a word processor you know how to use. Obviously, this word processor can be used to enter, edit, and print text. Make a list of other features in the word processor you have selected. How does each contribute to helping a person complete a writing task? Which of these features are most useful to you?
- 2. Examine the list of generic tools given in this chapter. If you know of some other computer tools that you feel belong in the list, add them to the list. Then analyze the list from a personal point of view. What is your current level of expertise in using each of the tools? What are your learning goals for each of these generic tools?
- 3. Select a domain that interests you. Analyze it from the point of view of usefulness of the generic computer tools listed in this chapter. How would the teaching and learning of this domain be affected by having these generic tools thoroughly integrated into the curriculum?
- 4. Select one of the generic computer tools that you routinely use. Analyze it from the point of view of how it increases your personal productivity. Cite evidence of ways in which it increases your personal productivity. You may find that you can also give good examples of how it decreases your personal productivity.
- 5. Again, select a domain of interest to you. Perhaps it is a domain that you are studying in school, or perhaps it is a domain in which you work. Analyze the domain from the point of view of specialized computer tools. Are there any specialized computer tools that are routinely used by experts in this domain? How is the teaching and learning of this domain being affected by computer tools that have been specifically developed for this domain?



Computer Programming

This chapter discusses some general ideas about computer programming. A computer program is a set of instructions that a computer can follow. The instructions are used to represent a problem and the steps needed to solve the problem.

A computer program may be designed to interact with a person, so that the person and a computer work together to solve a problem. Or a program may be designed to interact with a machine, such as an automated production facility or data-gathering instrumentation equipment.

Types of Computer Programming

A computer program is a detailed set of step-by-step instructions that a computer can interpret and carry out. That is, a computer program tells the computer hardware what to do. The process of creating a computer program is called programming. The terms computer program and computer software are used interchangeably.

Traditionally, computer programming has meant the process of designing and writing a computer program in a programming language, such as a machine language, an assembler language, or a higher level programming language (e.g., Ada, BASIC, C, C++). There are hundreds of different

higher-level programming languages designed to fit the specific needs of programmers working in different problem domains.

Over the years, the traditional definition of computer programming has gradually been modified. Here are three somewhat distinct levels or types of computer programming:

1. Command-level programming. Some examples of command-level programming include the following. You organize your computer desktop and tell the computer what to do when it starts up. You specify a number of preferences to your word-processing software. For example, you specify the format for the pages of the new document that is to be created when the word processor is started up. You specify what printer to use and a variety of printer options that the computer is to follow. These instructions to the computer may be given by a combination of keying in commands and making menu choices.

A word processor contains a number of provisions for issuing commands. For example, you can issue a command to spell check a document, to change the font that is being used, or to change the margin settings.

You do command-level programming when you make use of a productivity tool, such as a database or a spreadsheet, to represent and solve a problem. In a database or a spreadsheet you specify headings for various rows and columns to represent the data in the problem you want to solve. You specify formulas the computer is to use and the sorting operations that the computer is to carry out.

2. Productivity-tool programming. When using a productivity tool, such as a spreadsheet, database, or hypermedia, you may find that you want to accomplish tasks that cannot easily be accomplished just by the command-level programming options the productivity tool makes available to you. Many productivity tools contain a built-in programming language. It might be called a macro language. It might have a more specific name, such as dBASE or HyperTalk. It might be a

general-purpose programming language such as BASIC. Often, just a few lines of instructions in a built-in programming language can save a great deal of effort when using a productivity tool to solve a problem.

3. Programming using a general-purpose programming language. Ada, BASIC, and C++ are some of the many general-purpose programming languages that are not specific to any particular productivity tool. Generally, they are available in a variety of different computer platforms, such as IBM-compatible and Macintosh computers. These languages are often used to write sophisticated programs to be used by yourself and/or other people. Traditionally, computer programming has meant writing programs in such programming languages.

It is possible to use a computer without knowing much about computer programming. However, many computer users find that their expertise in using a computer to help solve problems is greatly increased by learning and understanding computer programming at one or more of these three levels.

Some History of Programming Languages

The first computers were designed to help solve math, science, and engineering types of problems. For the earliest computers, programming consisted of rewiring the machine. A computer was rewired to be able to solve a particular type of problem. Plug boards were used, much like the plug boards used in early telephone systems.

It was a major breakthrough when the idea of storing a computer program in the computer memory was developed. The program could then be changed by just changing the contents of the computer memory. Indeed, the instructions in a computer program could be designed to make changes to the computer program! This "stored program" change in the human-machine interface made it easy to store computer programs for reuse at a later date and for transporting them to other computer sites. The programs could be stored on punched paper tape, punch cards, or magnetic tape.

The mechanics of programming in the early days of computers were not simple because the human-machine interface was not well developed. Programming was done at the machine-language level. A machine is constructed to "understand" about 100 to 200 different numbered instructions, for example, 001, 002, 003, 004, and so on. The instructions are designed to work on the contents of individual computer memory locations, and each instruction accomplishes a small, specific task. For example, an instruction might change the sign of a number. A different instruction might compare two different memory locations to see if they are equal. Still another instruction might add the number 1 to a given memory location. The smallest error in indicating what instruction is to be done or what memory locations it is to affect can lead to completely wrong results. It is really easy to make a simple mistake, such as telling the computer to do instruction 187 when you really want the computer to do instruction 178. It is difficult to detect and correct errors when one is programming at the machine-language level.

Gradually the human-machine interface in computer programming was improved. A simple example was the development of mnemonics for the instructions and variable names for memory locations. Words, such as ADD, SUB, MUL, and DIV, are easier to remember than numeric codes for instructions. Variable names, such as LENGTH and WIDTH, are easier to work with than memory location addresses, such as memory location 21834 and memory location 02642. Computer programs were developed that translated the mnemonic instructions and the variable names into appropriate machine-language instructions and specific memory location addresses. The translating programs were called assemblers, and the languages themselves were called assembler languages or assembly languages.

During the early to mid-1950s, still better human-machine interfaces were developed to help programmers. These were called higher level programming languages. FORTRAN (standing for FORmula TRANslation) was developed over a span of time—from 1954 to 1957. This language was specifically developed for use by scientists and engineers. Many different versions of this language have been developed since then, and FORTRAN is still a widely used programming language in the science and engineering fields. FORTRAN is based on the language of algebra. A person who knows high school algebra usually finds that it is relatively easy to learn to use FORTRAN.

The FORTRAN programming language is representative of a key idea in the computer field. Not only can people develop better human-machine interfaces but they can also develop interfaces to suit the needs of specific domains. COBOL was developed for programmers working on business problems. BASIC was developed as a math tool for college students. The key idea is that a person who has considerable knowledge in a domain can build on this knowledge through learning a programming language specifically designed to help solve problems in that domain.

Over the years, more and more higher level programming languages have been developed. Now there are hundreds of different programming languages. Superficially, these generalpurpose, higher level programming languages seem to differ quite a bit from each other. However, in many ways, they share much in common. Some of the commonality is discussed in the following section.

Data Structures and Control Structures

A computer program has two main parts. First, it contains a representation of the problem, including the data that is to be input, stored, manipulated, and output. This is called a *data structure*. Second, it contains detailed instructions telling the computer exactly what to do with the problem representation and the data. This is called a *control structure*. Thus, a computer program consists of a combination of a data structure and a control structure.

The initial higher level programming languages, such as FORTRAN and COBOL, were oriented toward somewhat specific domains of problem solving. FORTRAN was designed to help a programmer create the types of data structures and control structures needed to solve math, science, and engineering problems. COBOL was designed to help the programmer create the types of data structures and control structures needed to solve business problems. As use of computers spread to other fields, people began to develop software to fit the specific needs of problem solvers in these fields. This has helped contribute to the current situation in which there are a number of general-purpose programming languages and there are a number of productivity tools with built-in programming languages.

A productivity tool is often designed to make it easy to represent certain kinds of problems. For example, a graphic artist needs to represent figures that contain lines, rectangles, circles, and other geometric shapes. Thus, a productivity tool designed for graphic artists contains good provisions for representing these types of figures. The graphic artist may want to view a figure from different perspectives. Thus, a productivity tool designed for graphic artists contains good provisions for manipulating a figure to show it from different perspectives.

The data structures and control structures needed by a graphic artist are certainly different from those needed by a musician. The musician uses musical notation to represent music. A musician needs to hear the music in addition to viewing its score. The musician needs easy provisions for manipulating and combining the sound from a number of different musical instruments. Thus, a productivity tool designed for musicians contains provisions to make it easy to accomplish all of these tasks.

Procedures and Procedural Thinking

Earlier parts of this book discussed BBRs and chunking. The use of chunking in problem solving plays a central role in the field of computer programming.

All programming languages support a particular type of chunking or BBRs that computer scientists call a *procedure*. A procedure is a self-contained set of programming language instructions that can be treated as a unit. A procedure is often called a subprogram or a subroutine. A computer program is usually designed as a collection of procedures that have been appropriately linked together to solve a problem or accomplish a task. Computer programmers are taught to think in terms of cognitively manageable units of computer code. Remember, the human mind can deal with a concept as a unit or a chunk far more easily than it can deal with the details of a concept. A cognitively manageable unit of computer code focuses on a single concept or task. A cognitively manageable chunk of code needs to be short enough and simple enough so that it can be easily tested and debugged (have its errors removed).

Thus, computer programmers learn to think in terms of developing cognitively manageable procedures. They learn to solve problems by using such procedures. This overall process is called procedural thinking. It is one of the most important ideas in computer programming. It is such an important idea that some definitions of computer literacy include the specification that to be computer literate, a person has to understand and make use of procedural thinking.

Object-Oriented Programming

The early programming languages provided rather limited aids to the programmer working to represent a problem. Two major types of advances have occurred. One type of progress is seen in productivity tools, such as a word processor, database, or spreadsheet. An entire application program is designed for the representation of a particular type of problem. A word processor, for example, is designed to make it easy to represent the types of information that writers must deal with. A spreadsheet is designed to make it easy to represent certain types of business problems.

A second major breakthrough was the development of object-oriented programming. Some examples of objects that a programmer might want to include in a program could be a pull-down menu, a button, and a scrolling field. Each of these objects varies in physical size, placement on the screen, and how it is to interact with the computer user. An objectoriented programming language contains a number of built-in objects. With a few keystrokes, the programmer designates the specifics of an object that possesses an underlying computer programming code of possibly thousands of lines in length. In an object-oriented programming language, the programmer can also easily create new objects. Initially, object-oriented programming was considered to be an esoteric subject. We now have reached the stage where a number of programming languages and computer applications contain object-oriented provisions that are easy to use. For example, in an engineering drawing program there are good provisions to create objects consisting of combinations of circles, rectangles, and other geometric shapes. These objects can easily be sized, rotated, and positioned as needed when undertaking a drawing task.

Now, object-oriented programming has come to the forefront, proving to be an important key to building on the previous work of other people. If an object is appropriately defined in a programming language, it can be used by any programmer who has need for such an object.

Software Engineering

As computers started to become commercially available during the early 1950s, many colleges and universities developed programs of study in this field. Many of the early programs of study focused on computer programming. Thus, a student might take a beginning and advanced course in assembler-language programming, a beginning and advanced course in FORTRAN programming, and a beginning and advanced course in COBOL programming. The focus of the instruction was on the specific tool, rather than on problemsolving ideas that cut across different programming languages and domains. Now, such a curriculum is considered archaic.

Gradually it became clear that all programming languages share a lot in common. Moreover, it became clear that a number of problem solving and programming ideas are applicable to every computer programming task. A field of study and research called *software engineering* began to develop. Software engineering includes computer programming, but it also includes the study of data structures, control structures, the mathematics of solving problems via computer, and a number of different programming practices and ideas that are independent of any particular programming language. For example, in many different problems it is necessary to sort a table of data into alphabetical or numerical order. There are many different ways to do this. Some are much faster than others, and this partly depends on the circumstances of the particular problem. Thus, the study of sorting algorithms is a standard topic in software engineering.

In the early history of programming, computer programs tended to be relatively short. One person could fully understand the problem to be solved, write and test the software, and correct errors that were later found by users of the software. This gradually changed. It is estimated that it took the equivalent of 20 person-years of effort to develop the FORTRAN programming language and the program to translate from it into machine language for one particular machine. A team of some of the best programmers in the world worked on this project for several years.

Two important ideas emerged as programmers struggled to cope with large and complex projects. First, a large program should be constructed from small, self-contained modules (subprograms) that can be independently written and tested. A subprogram needs to be short enough and simple enough so that a programmer can fully understand it and take responsibility for that piece of the overall program.

Second, the idea of structured programming was developed. Computer programming became more of a systematic science. Programmers learned to think of a program as having structure, with the pieces of this structure being created, tested, and assembled in a systematic manner.

A careful analysis of computer programs reveals that every program can be built from three basic types of structures or constructs: sequence, selection, and repetition.

- Sequence means putting together pieces or chunks of program code so that the machine carries them out in a linear (sequential) order.
- Selection is a branching process—the machine is to carry out only one of several different chunks of program code, depending on particular circumstances. For example, if the height of a person is more than 180 cm and the weight of the person is less than 70 kg, then one chunk of code is to be executed. If the height is more than 180 cm and the weight is greater than or

equal to 70 kg, a different chunk of code is to be executed.

• Repetition refers to carrying out the instructions in a chunk of code over and over again, either some specified number of times or until some particular condition is satisfied.

Software engineering is now a well-developed discipline. It is both embedded in computer and information science coursework and is taught in separate courses.

Transfer of Computer Programming Learning

Does skill in one higher-level programming language transfer to learning and using a different higher-level programming language? This will be discussed in the current section.

Transfer of computer-programming learning can also be examined for transfer to noncomputer problem-solving environments. This is covered in the next section.

A computer program consists of data structures and control structures that are designed to solve some type of problem. A skilled computer programmer is adept at creating the data structures and control structures for solving problems in one or more domains. This requires three types of expertise:

- 1. Expertise in understanding and solving problems in the domain.
- 2. Expertise in the representation of problems and their data, and using the types of data structures that are available in a computer programming environment.
- 3. Expertise in the design and implementation of a set of instructions that manipulate the problem representation and the data associated with the problem to solve the problem.

The first of these three types of expertise is domain specific. It requires a great deal of knowledge about the specific domain. Research suggests that there is not too much transfer of learning of such domain-specific knowledge from one domain to another. Of course, an exception occurs when two domains contain a considerable overlap. For example, theoretical physics overlaps strongly with mathematics. Thus, a skilled mathematician may be able to make a major contribution toward solving problems in theoretical physics.

Considerable transfer of learning occurs in the second and third types of expertise in the preceding list. This transfer occurs as one moves among different hardware and software environments. It occurs as one moves among various productivity tools and among various programming languages. The transfer can be both low-road (automaticity) and highroad (reflectivity) transfer.

For example, suppose that you have learned a programming language that is object oriented and contains a scrolling field as one of the built-in objects. In this programming language, as you conceptualize the writing of a program to solve a problem, you are mindfully aware that a scrolling field is available as a BBR. You may make use of this BBR often, as you deal with a wide range of problems.

Now, suppose that you need to program in a different programming language that is not object oriented and does not contain a scrolling field as a built-in object. If the types of problems that you want to solve require a variety of scrolling fields, your immediate response to the situation will be to find or to write a procedure (a subprogram) that creates and manipulates scrolling fields. That is, you will add this BBR to the programming-language environment in which you are working.

It takes a great deal of training and experience to learn the many and varied ways to represent problems in a computerprogramming environment. The actual facilities that are built into different programming languages for the representation of problems vary from language to language. But, once you have mastered the basic ideas of problem representation, most of what you have learned will transfer from language to language.

A structured programming language encourages a type of modularity (a type of structure) in the program design. Structured programming is an idea that cuts across all programming languages. Thus, once you master the concept of structured programming, you can easily adjust your ideas to any programming environment.

To summarize, software engineering can be thought of as the science of computer programming. Computer programming involves representing a problem and representing the steps needed to solve the problem. The details of both the representation of a problem and of the steps to solve a problem will vary among different programming languages. A problem may be very easy to solve in one language and difficult to solve in another, or vice versa. However, the general underlying concepts transfer from language to language. This transfer is enhanced if the learner strives for learning that will transfer. Both the low-road and the high-road mechanisms are applicable.

Transfer of Problem-Solving Skills From Programming

Computer programming can be thought of as a type of problem solving in which the main resources are a computer and the computer programmers. You might think that the problem-solving skills needed in this environment easily transfer to solving problems in other environments that do not depend on computer programming.

There have been many research studies on the transfer of problem-solving skills from computer programming to other environments. In the typical study, some students (the experimental group) learn computer programming while other students (the control group) spend their time on some other learning tasks. The groups are matched on the basis of their initial overall skills in problem solving that do not involve the use of computers. After the "treatment," both groups are tested for their general noncomputer problem-solving skills.

The early researchers were puzzled by the rather consistent results indicating that little or no transfer of problem-solving learning surfaced in such studies. Salomon and Perkins (1988) and Perkins (1995) summarize the early research literature in this field. They describe a low-road/high-road transfer theory. They analyze the research studies on the basis of whether the treatment was powerful enough to lead to low-road and/or high-road transfer. Their conclusion is that the treatment in these types of studies has seldom been adequate to lead to either low-road or high-road transfer.

Based on an analysis of the research literature and their own studies, Salomon and Perkins conclude that there will be transfer of learning in problem solving if computer programming is taught in a manner that helps transfer to occur. That is, transfer of learning will occur if the instruction meets the conditions that are needed for either low-road transfer or high-road transfer.

Expertise in Computer Programming

It is easy to develop a modest skill level in one, two, or even all three of the levels of computer programming discussed near the beginning of this chapter. With appropriate instruction and practice, elementary school students can do this. Even a modest level of programming competence can contribute significantly to your overall level of expertise in using computers in many different domains.

Many studies have been designed to determine the characteristics or types of native skills that help a person become a good computer programmer. As might be expected, the logical-mathematical category of intelligence tends to correlate highly with success as a computer programmer. However, musical intelligence also tends to correlate highly. And, of course, persistence and attention to detail are critical.

Nowadays, a considerable amount of computer programming is done by teams of people. Interpersonal skills and communication skills are important in this type of programming environment.

By definition, the experts in any domain are able to outperform people who are merely competent or who are novices in the field. Computer programming is a domain in which people with a high level of expertise really stand out relative to people who are merely competent in the domain. The expert can undertake and accomplish tasks that are well beyond the capabilities of a person who is a competent programmer. This helps explain the high salaries that such experts can obtain.

As with any domain, it is possible to learn computer programming on your own. However, software engineering is now a well-developed field of study. Many different computerscience programs of study include software engineering coursework. Such coursework can be very helpful in developing expertise as a programmer.

Activities and Self-Assessment

- 1. Analyze your current levels of programming expertise in each of the three types of computer programming discussed in this chapter. What are your strengths, weaknesses, and goals? What are you doing to achieve your goals?
- 2. Consider one productivity tool, such as a word processor, database, or spreadsheet, that you know how to use. Describe the types of problems that this productivity tool is designed to represent. Now select some other type of problem domain. How hard or how easy is it to represent the problems of this domain in the productivity tool that you have discussed? For example, is it easy or hard to represent business problems in a word processor and then use the word processor to help solve these problems? Can a spreadsheet be used to solve database problems, or vice versa?
- 3. An integrated package contains a number of different productivity tools. Each productivity tool is designed to make it easy to represent and to solve certain categories of problems. Select an integrated package that you are familiar with. Make a list of problem domains in which the problems are easily represented in this integrated package. Make a list of problem domains that are not easily represented in this integrated package.



Final Remarks

The focus of this book is on gaining increased expertise in problem-solving domains that interest you. In many domains, the computer is now a routine aid to problem solving; that is, it has become a cost-effective and versatile resource. Thus, you are faced with the problem of deciding what you will learn about using computers.

Summary of Key Ideas

This book has covered a number of key ideas about problem solving and the roles of computers in problem solving. Here are a few of them:

- 1. Your time is a limiting factor.
 - You can spend learning-time before you encounter a problem. Here the idea is learning to learn, acquiring a broad base of knowledge, and laying the foundation for "just in time" learning.
 - You can spend learning-time when you encounter a problem. This is "just in time" learning.
 - You can learn as you solve problems. Experts tend to expand their horizons in this manner. They deliberately seek out difficult problems and use them

as a vehicle to expand their expertise. They are reflective and mindful. This is an effective way to combine problem-solving time and learning-time.

- 2. You have many and varied talents and potential. Some of these are summarized in the theory of multiple intelligences developed by Howard Gardner. Robert Sternberg provides a somewhat different definition of intelligence, and David Perkins provides still another analysis of intelligence. All of these definitions support the idea that you can improve your ability to perform in an intelligent manner. However, it takes time, hard work, and persistence to develop your talents.
- 3. As you learn, pay attention to transfer of learning. The low-road/high-road theory of transfer may provide you with a useful starting point. Metacognition, mindfulness, and reflexiveness are essential. Intrinsic motivation and ownership of the learning task can be a big help; thus, you may want to concentrate your learning efforts in domains that really interest you.
- 4. A computer system is a rapidly changing and rapidly improving tool. This rapid pace of change will continue for many more years. Time invested in learning to use a computer as an aid to problem solving will be repaid over and over again in the future. You will be investing this time in a tool that will become more and more powerful.
- 5. In many ways, we are just at the beginning of the Information Age. More and more information is being digitized. We have more and more connectivity, such as having access to the Information Superhighway. We have more and more computing power that can be used to process digitized information to help solve problems and accomplish tasks. This suggests that if you are not already a comfortable user of computers and the Information Superhighway, you should give careful consideration to becoming one.
- 6. Most people are far behind the frontiers of state-of-theart usage of computer tools in their domains of expertise. Their pace of learning is slower than the pace of change of computer technology within their domains.

That is, most people are falling further behind. This suggests that a paradigm shift is needed; knowledge workers need to spend more of their time learning and helping their colleagues learn. This also suggests that in domains where the computer is an especially useful tool, many young students can quickly surpass their elders.

7. Most real-world problems are interdisciplinary. That is, they cut across traditional disciplines and traditional domains. This creates an interesting difficulty for students. By dint of many years of hard work, you can develop a high level of expertise in one or two domains. You will then be good at solving a wide range of problems that occur in these domains. However, you will be quite restricted in solving most problems, since most problems cut across a number of different disciplines.

Alternatively, you can elect to pursue the domain of interdisciplinary problem solving. An expert in this domain is good at facilitating the assembly and use of the range of resources needed to solve problems that cut across traditional domains and traditional disciplines. This domain of expertise requires a broad range of knowledge, ability to learn quickly, good communication skills, and good ability to work with a wide range of people. And, of course, the computer is one of the tools that helps glue these many and varied abilities together.

- 8. The term "intelligent digital connectivity" summarizes ideas from personal and group productivity tools, the Information Superhighway, the digitization of many different kinds of information, and the computer as a powerful tool. Intelligent digital connectivity is changing the way that business and industry work (Peters, 1994). It is making it possible for people located throughout the world to work together on a specific problem. Teams are quickly assembled, they accomplish their task, and are then disbanded. This "just in time" assembling of human resources is the wave of the future.
- 9. Intelligent digital connectivity has spawned the idea of distributed intelligence. A system consisting of people, computers with artificial intelligence, intelligent networks, and groupware is a powerful aid to solving

problems and accomplishing tasks. Each of the components of such a system is improving over time—some much more rapidly than others. For example, the human component improves through education and experience in working in such a distributed intelligence problem-solving environment. Such progress is slow relative to the rate of improvements in the speed and power of computer hardware and computer networks.

Businesses throughout the world are increasing their awareness of the ideas in the preceding list. Increasingly, a business identifies a problem, assembles a team of people to work on the problem, and then disburses the team when the problem has been solved. Team members may be located throughout the world, connected by the Information Superhighway. Some of the team members may be regular employees of the business, but others may be independent contractors who have a high level of expertise in the domain of the problem. Individual members of the team may have to learn a great deal about the specific problem in a short time.

This model of problem solving is one in which there is worldwide competition for expertise. It is a model of worldwide competition for the most precious resource—the time and energy of creatively intelligent experts. You can improve your competitive position by learning to make effective use of the ideas covered in this book.

Activities and Self-Assessment

- 1. Analyze your strengths and weaknesses as a domainspecific problem solver and as an interdisciplinary problem solver. As you continue to pursue education and a career, are you focusing on one specific domain? If so, does this domain fit well with your strengths? Alternatively, if your focus is interdisciplinary, in what ways are you building on your specific strengths and interests?
- 2. Analyze your current and future job potential from the viewpoint of continued rapid progress in worldwide

intelligent digital connectivity and in distributed intelligence.

3. Think about some of the ideas that you have encountered in this book. Make a list of some of the ideas that seem most relevant to you. Pay particular attention to your reflexiveness, high-road transfer of learning, and distributed intelligence. Analyze your list. What are you doing to make use of the ideas on your list?



References and Resources

Bereiter, C., & Scardamalia, M. (1993). *Surpassing ourselves: An inquiry into the nature and implications of expertise*. Chicago and La Salle, IL: Open Court.

A seminal book on expertise. It is aimed at educators and education in general, but it also discusses some of the roles of computers in expertise. Not easy reading, but it is well worth whatever effort it takes.

Communications of the ACM (1994, March).

The Association for Computing Machinery is a very large professional society consisting of professionals in the computer field. *Communications of the ACM* is written for people who have a solid background in the domain of computer and information science. However, the March 1994 issue contains a number of summary, overview articles on artificial intelligence that have been written for people who are not specialists in this area. They provide an excellent overview of the problems of this domain and the progress that has been made in addressing these problems.

Communications of the ACM (1994, July).

See the comments about the ACM in the preceding annotation. The July 1994 issue of *Communications of the ACM* contains a

number of excellent summary, overview articles about intelligent agents.

Costa, A. L., (Ed.). (1991). *Developing minds: A resource book for teaching thinking*. Alexandria, VA: Association for Supervision and Curriculum Development.

This resource book produced by the Association for Supervision and Curriculum Development provides an excellent overview of what is known about teaching thinking skills. It contains a wide range of articles written by people who are experts in this domain.

de Bono, E. (1973–75). *CoRT thinking.* Blandford, Dorset, England: Direct Education Services Limited.

The Cognitive Research Trust (CoRT) thinking program is a program of study designed to increase thinking and problemsolving skills. This program has been the subject of quite a bit of research and has been implemented in a number of different places. It is an example of a type of program that has proven effective in increasing general problem-solving skills.

de Bono, E. (1985). *De Bono's thinking course*. NY: Facts on File, Inc.

Edward de Bono is a world-class expert in teaching thinking skills. He is a prolific author and inspirational workshop presenter. His books give practical, down-to-earth ways for improving thinking skills. Some of his books are aimed at business people and others are aimed at educators. His books are widely sold, so you are apt to find one or more of them in any major bookstore.

de Bono, E. (1992). *Serious creativity: Using the power of lateral thinking to create new ideas.* NY: Harper Business. See the comments under de Bono, E. (1985).

Frederiksen, N. (1984). Implications of cognitive theory for instruction in problem solving. *Review of Educational Research*, *54*, 363–407.

This is a summary and analysis of the research literature on problem solving. It is a good starting point if you want to explore the research literature. The article contains an extensive bibliography.

Frensch, P. and Funke, J., (Eds.). (1995). *Complex problem solving: The European perspective.* Hillsdale, NJ: Lawrence Erlbaum Associates.

This book provides an excellent overview of the current research on complex problem solving. The main emphasis is on research being done in Europe. However, there is an excellent chapter written by Robert Sternberg that compares and contrasts problem-solving research in the United States with problem-solving research in Europe. In recent years, much of the problem-solving research in the United States has focused on specific domains in which one can acquire a great deal of expertise. Examples include chess, electronics, lawyer's reasoning, physics, and writing. Research in Europe tends to focus on more general problems, such as managing the resources of a city. Europeans make use of complex computer simulations of the problem-solving environments to be studied.

Gall, M., Gall, J., Jacobsen, D., & Bullock, T. (1990). *Tools for learning: A guide to teaching study skills*. Alexandria, VA: Association for Supervision and Curriculum Development.

The Association for Supervision and Curriculum Development is a large professional society. It often commissions books that are based on the latest research in a domain and that are also strongly focused on how to implement the underlying ideas in the classroom. This book is written for educators who want to improve their own learning skills and the learning skills of their students. There is considerable emphasis on the cognitive learning theories and on ways you can improve your study skills.

Gardner, H. (1982). *Art, mind, and brain: A cognitive approach to creativity.* NY: Basic Books.

Howard Gardner is a cognitive psychologist and cognitive scientist. He is a prolific author and recognized for his research and writing in a number of areas of education. The three books listed here are representative of his work. The book focuses on creative intelligence. Gardner's books are widely sold, so you are apt to find copies of some of them in major bookstores.

Gardner, H. (1983). *Frames of mind: The theory of multiple intelligences*. NY: Basic Books. (A 1993 edition is now available; it is the same as the 1983 edition, except that it contains additional material in the preface.)

The 1983 book was written for a somewhat narrow, technical audience. The book has proven immensely popular, as have the general ideas contained in the book. This book, and the following one on the list, provide an excellent introduction to the theory of multiple intelligences and some applications of the theory.

Gardner, H. (1993). *Multiple intelligences: The theory in practice.*

NY: Basic Books.

This book expands on ideas originally presented in *Frames of Mind*. It provides a variety of examples of applications of the original theory. There is considerable emphasis on applications to education.

Hirsch, E. D. (1988). *Cultural literacy: What every American needs to know.* NY: Vintage Books.

Effective communication between people depends not only on having a common language but also on having some common knowledge and experiences. Hirsch's research work has focused on identifying a common core of topics and ideas—a type of cultural literacy—that contributes to the ability of people, in the United States, to communicate effectively with each other. One way to think about the content of his book is from the point of view of helping to solve the communication problem among people in the United States. If each person has a common core knowledge, such as the core proposed by Hirsch, this will help each individual solve the communication problem. Kulik, J.A. (1994). Meta-analytic studies of findings on computer-based instruction. In E. Baker and H. O'Neill (Eds.), *Technology assessment in education and training* (pp. 9–33). Hillsdale, NJ: Lawrence Erlbaum Associates.

James Kulik has been doing metastudies on computer-based learning (computer-assisted instruction) for many years. A number of his studies have been funded by the National Science foundation and other federal agencies. This specific article is an analysis of the metastudies that he and others have carried out. That is, it is a meta-metastudy. It presents convincing evidence that CAI works. The article also contains an extensive bibliography, so it provides an excellent starting point for a person who wants to explore the literature on CAI.

Moursund, D., & Yoder, S. (1990). *LogoWriter for educators: A problem solving approach*. Eugene, OR: International Society for Technology in Education.

Moursund and Yoder (also, Yoder and Moursund) are prolific authors in the field of technology in education. Each of their books listed in the References and Resources section focuses on a computer tool and underlying aspects of using the tool to solve problems. A number of other books that they have authored individually and jointly are available through the International Society for Technology in Education.

Moursund, D., & Yoder, S. (1993). *Problem solving and communication in a HyperCard environment*. Eugene, OR: International Society for Technology in Education.

The *HyperCard* software first became available in 1988. It can be used to create hypermedia documents and to make use of them on the Macintosh computer. This book contains a strong focus on problem solving as it helps you learn to use *HyperCard*.

Norman, D. (1990). *The design of everyday things*. NY: Doubleday.

Donald Norman is a cognitive scientist and a prolific, witty author. He has a high level of expertise in the domain of human-machine interfaces. He is interested in both noncomputer and computer-based human-machine interfaces. This book provides an excellent introduction to the design of such everyday tools as doors, drawers, and stoves. He gives many examples of poor human-tool designs.

Norman, D. (1993). *Things that make us smart: Defending human attributes in the age of machines.* Reading, MA: Addison-Wesley.

This publication provides a superb discussion of roles of technology in enhancing our intellectual capabilities. Norman emphasizes that poorly designed machines can make us feel dumb and prevent us from making effective use of our intelligence. See Norman (1990).

Perkins, D. (1995). *Outsmarting IQ: The emerging science of learnable intelligence*. NY: The Free Press.

This book provides a careful analysis of possible definitions of intelligence and how IQ is measured. Three different but closely related components of intelligence are explored: neural intelligence, experiential intelligence, and reflexive intelligence. Arguments are presented to support the contention that all three components of IQ can change. In particular, appropriately designed education can increase experiential and reflexive IQ. This book also has a major focus on transfer of learning, with particular emphasis on the highroad/low-road theory of transfer developed by Perkins and Salomon in 1987.

Peters, T. (1994). *The Tom Peters seminar: Crazy times call for crazy organizations.* NY: Vintage Books.

Tom Peters is a "guru" in business consulting. The reference listed here focuses on how the information and communication technologies are changing the world of business. The book is fast paced and loaded with important, challenging ideas about how business is and/or should be facing the issue of how to solve problems and create products in today's and tomorrow's world. The book is somewhat zany and irreverent, but makes many points that are important to problem solvers of today and tomorrow. Polya, G. (1957). *How to solve it: A new aspect of mathematical method* (2nd ed.). Princeton, NJ: Princeton University Press.

This book by George Polya is considered to be a classic in the field of learning and teaching problem solving. The emphasis is on strategies and metastrategies that are applicable over a wide range of math problems. A number of the strategies that are discussed are applicable in areas outside of mathematics and thus contribute to transfer of learning to other fields. Examples include breaking a problem into subproblems and relating a problem to other problems encountered in the past.

Salomon, G., & Perkins, D. (1988, September). Teaching for transfer. *Educational Leadership*, 22–32.

Salomon and Perkins have developed the high-road/low-road theory of transfer of learning. The article listed here provides a good overview of the domain of transfer of learning and how to teach transfer. It also contains an extensive bibliography, so it is a good starting point if you want to study the research on transfer of learning.

Shekerjian, D. (1990). *Uncommon genius: How great ideas are born*.

NY: Penguin Books.

This book is a study of 40 people who have received the John D. and Catherine T. MacArthur Foundation fellowships. These are often called "genius" awards; the fellows are selected on the basis of their creative intelligence. Howard Gardner was a MacArthur Foundation fellow. The recipients are given five-year awards, with the yearly award being in the range of about \$50,000. They are free to pursue whatever interests them. This book explores commonalities among various MacArthur fellows. One commonality is persistence, with great success often following many years of lack of success.

Sternberg, R. (1988). *The triarchic mind: A new theory of human intelligence*. NY: Penguin Books.

This book provides an excellent overview of the history and work on attempting to define and test intelligence. Sternberg argues that previous theories are inadequate, and he presents a three-part definition of intelligence. Sternberg is a strong supporter of the idea that intelligence can be improved.

Toffler, A. (1990). *Powershift: Knowledge, wealth, and violence at the edge of the 21st century.* NY: Bantam Books. Alvin Toffler and his wife Heidi are futurists who have written three major books (published in 1970, 1980, and 1990 respectively) that represent our changing world. The key idea in this most recent book is that knowledge is power (knowledge is a resource) and that this form of power is rapidly changing the world. The book explores other forms of power (other resources), such as agricultural productivity as power, industrial manufacturing capacity as power, and violence (military might) as power. Various countries are analyzed on the basis of the balance that they have in these different areas of power.

Yoder, S., & Moursund, D. (1995). *Introduction to ClarisWorks 4.0: A tool for personal productivity*. Eugene, OR: International Society for Technology in Education.

ClarisWorks is an integrated package of software designed to run on both IBM-compatible and Macintosh platforms. It is representative of the modern integrated packages of software designed as general-purpose aids to problem solving across many different disciplines. This book contains a major focus on problem solving.

Index

 7 ± 249 Abacus 41 Accomplish a task 10 ACM (see Association for Computing Machinery) Advance organizer 4 Agent 94, 127 AI (see Artificial intelligence) Arabic numerals 15 Artifact cognitive 2 physical 2 Artificial intelligence 69, 89, 127 Assembler language 113 Association for Computing Machinery 78 Baker, Eva 130 BASIC 86, 114 **BBR 47** and transfer of learning 76 computer-based 49 Behavioral learning theory 80 Bell, Alexander Graham 26 Benet, Alfred 33 Bereiter, Carl 5, 127 Black box 50 Building block resource 47

Bullock, T. 129 CAD (see Computer-assisted design) CAD/CAM 107 CAI (see Computer-assisted instruction) Calculator 41 as black box 53 handheld 41, 96 Calculus 29 CAM (see Computer-assisted manufacturing) Cards punch 88 Chess experts 63 Chip 84 Chunking 49, 115 ClarisWorks 108, 133 Clearly defined problem 12 COBOL 86, 88, 114 Cognitive artifact 2, 25 mathematics as a 44 Cognitive learning theory 81 Cognitive research trust 128 Cognitive science 4 Command-level programming 112 Complex problem solving 9, 12, 128

Complex problems 12 Computer 3, 45 as black box 50 as productivity tool 78 as tool 9 engineering 78 first 84 history 84 IBM model 650 84 instructional uses 78 laptop 85 literacy 105 memory 84 networks 79 program 45, 111 science 78 software 111 speed 84 time-shared 88 Computer literacy 115 Computer model 16 database 24 spreadsheet 24 Computer programming 79 Computer-assisted design (CAD) 102, 107 Computer-assisted instruction (CAI) 78, 79, 130 drill and practice 79 simulation 80 tutorial 80 Computer-assisted manufacturing (CAM) 107 Concept 43, 66 and theory 66 learning underlying 44 **Connectivity 94** Contemporary standards 5, 16, 44 Control structure 114 **CoRT 128** Costa, Arthur L. 128

Cramming for a test 58 Creativity 128, 129 Data 26 processing 78 structure 114 Database 24, 102 de Bono, Edward 25, 128 Declarative knowledge 57, 61 episodic 58 semantic 58 Design 101, 103, 130 Desktop presentation 103 Desktop publication 100, 101, 103 Digital connectivity 94 Disk magnetic 85 Distributed intelligence 95, 125 Domain of a problem 59 specific knowledge 59 specificity 59 Don't reinvent the wheel 56, 60 **DOS 85** Draw a picture strategy 65 Drill and practice 79 Dull saw 106 Education and training 73 Electrocardiograph 92 Electronic digital computer 3 ENIAC 84 Episodic memory (see Declarative knowledge) **Errors** detecting and correcting 51 Essential declarative information 58 Experiential intelligence 38 Expert 5

Expert system 92 as BBR 94 Expertise 4, 5, 19, 127 contemporary standards 5 in computer programming 118 increasing 19 versus specialization 5 Far transfer 76 Fault tolerant 89 Feedback 20 Formal problem 12 FORTRAN 86, 88, 113 Frederiksen, N. 128 Frensch, P. 9, 128 Funke, J. 9, 128 Gall, J. 129 Gall, M. 129 Gardner, Howard 4, 33, 34, 129 General-purpose programming language 112 Generic tool 102 Geometry 106 Givens 12 Goal 12 Graphical user interface 88 Graphics 103 draw 103 paint 103 Graphing 103 Group-productivity tools 79 Groupware 56, 79, 103 Guess and check strategy 69 GUI (see Graphical user interface) Handheld calculator 41, 96 Hard work 7 Heuristic 64 High-road transfer 76, 120, 132 Higher level language 113 Higher-order skills 43, 58

Hirsch, E. D. 58, 129 History of programming languages 113 Human mind 1 aids to 2 Human-machine interface 30, 34, 88 Hunt and peck 100 HyperCard 86, 130 Hypermedia 104, 109 HyperStudio 86 IBM model 650 84 Icon 88 If-then rules 93 Information 26 processing 45 retrieval 31, 59 superhighway 56, 94 Information Age 2, 124 Information explosion 26 Integrated package 108 **Intelligence 33** definition of 34 distributed 95 nature 40 nurture 40 Intelligence quotient 4, 25, 33, 34 Intelligent agent 94 Intelligent digital connectivity 94, 125 Interface 30 Internet 70 IQ (see Intelligence quotient) Jacobsen, D. 129 Jordan, Michael 82 Journaling 20 Just in time education 31, 73 Just in time learning 123 Keyboarding 100

Know how 57 Know what 57 Knowledge 26 accumulated 56 declarative 57, 61 procedural 57, 60 Knowledge is power 2, 132 Knowledge workers 124 Kulik, James 81, 130 Language translation 92 Languages programming 86 Laptop computer 85 Learning styles 21 Learning theory behavioral 79, 80 cognitive 80, 81 Leibniz, Gottfried 29 Library 31 research 70 Lifelong learner 27 Logo 130 Low-road transfer 76, 120, 132 Machine intelligence 89 language 113 Macintosh 89 Magnetic disk 85 Math systems 104 Mathematical notation 14 problem solving 131 Mathematics as a language 14 formula 14 Memorization 61 Memory mapping 16 Mental model 13, 21 Metacognition 6, 20

Microcomputer installed base 85 Microsoft Works 108 Mind 1 limitations 1 Mindfulness 77, 119, 124 Motivation extrinsic 19 intrinsic 19 Moursund, David 130, 133 Multiple intelligences 4, 34 National Council of Teachers of Mathematics 41 Natural languages 92 Natural talent 7 Near transfer 76 Networks 45, 79, 88, 94 Neural intelligence 37, 49, 131 Newton, Isaac 29 Norman, Donald 34, 130 O'Neill, Harold 130 **Object-oriented programming** 116 One trial learning 58 Outliner 100 Ownership 12, 17 Paper tape 88 Pattern matching 63 Perkins, David 4, 37, 73, 120, 131, 132 Persistence 19 Personal computer 88 Personal-productivity tools 79 Peters, Tom 131 Physical artifact 2, 25 Pie chart 49 Polya, George 131 Positive attitude 21 Problem

clarification 17 clearly defined 12 difficulty levels 18 formal 12, 17 interdisciplinary 124 poorly defined 17 posing 17 representation 13, 105 situation 17 unsolvable 18 Problem solving concepts in 43, 50 persistence in 19 processes in 43, 50 role of memorization 61, 64 six-step strategy 65 Procedural knowledge 57, 60 Procedural memory 101 Procedural thinking 115 Procedure 115 Process 43 Process writing 99 Productivity tools 79, 99 generic 99 Productivity-tool programming 112 Programming 111 command level 112 general purpose 112 productivity tool 112

Programming languages 86, 111 assembler 111 higher level 111 history of 113 machine 111 Punch cards 88 Punch paper tape 88 Reflective intelligence 66 **Reflectiveness** 77 Reflexive intelligence 38 Representing a problem 13 computer model 16 mathematical model 14 mental model 13 oral model 13 written model 13 **Resources** 12 accumulated knowledge 24, 26 computer 30 creative intelligence 24, 25 education and training 24, 27 time 24, 28 tools 24, 25 Robot 2 Roman numerals 15 Rote memorization 61 in chess 64 Rote memory 63 Rule of thumb 64 Salomon, Gavriel 120, 132 Scardamalia, Marlene 5, 127 Semantic memory 58 Shekerjian, Dennis 39, 132 Short-term memory 49 Simon, Theodore 33 Simulation 80 Six-step strategy 65 Software 85, 111 applications 86 engineering 116, 119

programming languages 86 systems 85 Solve a problem 10 **Specialization 5** versus expertise 5 Speed of computer 84 Spreadsheet 24, 104 Square root 84 Sternberg, Robert 37, 132 Stored program 113 Strategy 59, 64 break into pieces 59 draw a picture 65 for learning 75 for memorization 65 for problem solving 65 guess and check 69 ineffective 65 six-step 65 top down 65 trial and error 69 Study skills 74, 129 Suite 108 Symbol manipulation 45 System 7 85 Telecommunications 24, 104 Theory and concepts 66 Thinking 128 Time-shared computing 88 Toffler, Alvin 132 Toffler, Heidi 132 Tools generic 99, 102 make us smart 34 Top-down strategy 65 Training and education 73 Transfer 132 Transfer of learning 61, 75, 89 far transfer 76 high road 76

in computer programming 118 in programming 120 low road 76 near transfer 76 Transistor 84 Trial and error strategy 69 **Tutorial 80** Unsolvable problem 18 Use it or lose it 38, 58 Vacuum tube 84 Virtual reality 67, 80 Voice input 92 What if? questions 69 Word processor 99, 104 World-class experts 61 Yoder, Sharon 130, 133
Error:

Declarative knowledge can be divided into two categories. Episodic declarative knowledge is concrete. The knowledge is established when an episode or unusual event occurs. (Think of this as one trial learning.) [Episodic] Declarative knowledge is very personal, intimately tied to specific episodes or settings such as: one's first kiss on a date; one's wedding; or a particularly nice vacation event.