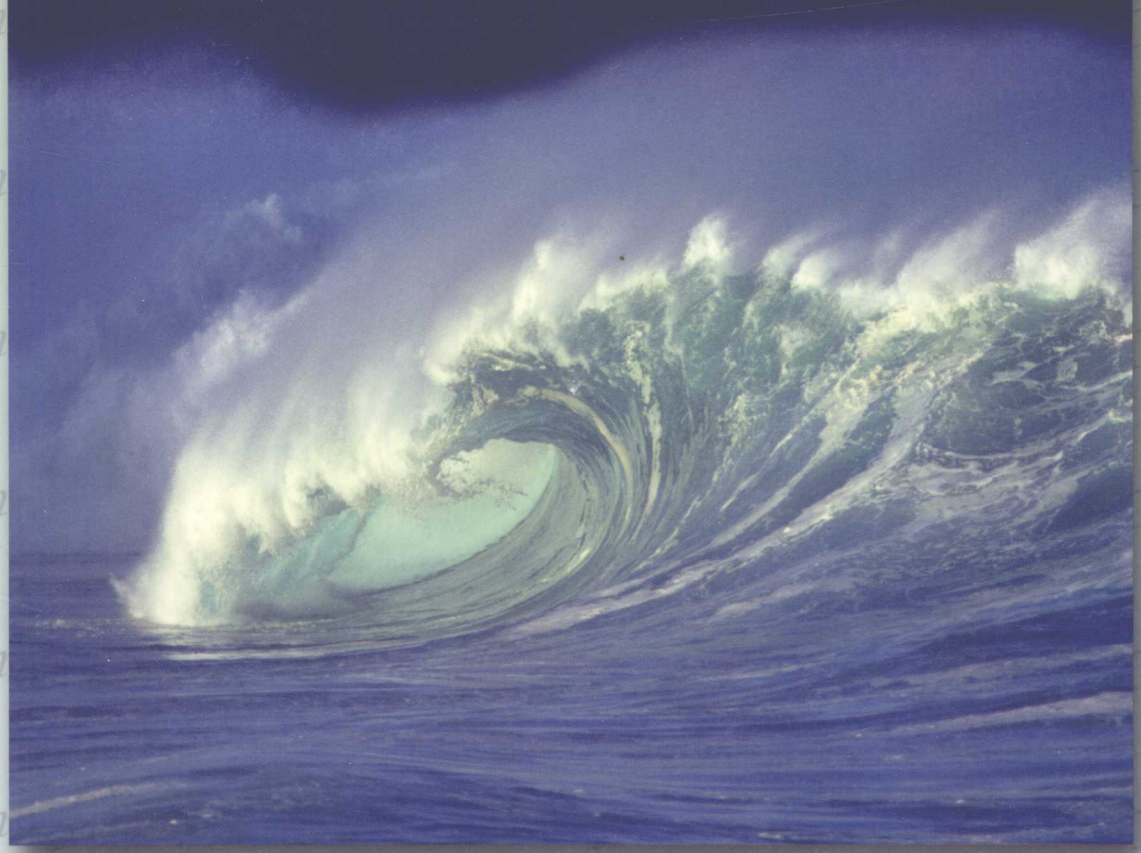
LAWRENCE SNYDER

fluency with Information Technology

Skills, Concepts, & Capabilities



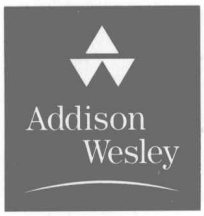
Fluency with Information Technology

SKILLS, CONCEPTS, AND CAPABILITIES

Preliminary Edition

LAWRENCE SNYDER
University of Washington, Seattle

"In which the secrets of computers and networks are revealed, in English, with no math, and starting at the very beginning."



Boston San Francisco New York
London Toronto Sydney Tokyo Singapore Madrid
Mexico City Munich Paris Cape Town Hong Kong Montreal

Executive Editor: Susan Hartman Sullivan
Associate Editor: Galia Shokry
Executive Marketing Manager: Michael Hirsch
Managing Editor: Pat Mahtani
Production Supervisor: Vicki Hochstedler
Cover Design: Regina Hagen Kolenda
Text Design: Vicki Hochstedler
Design Manager: Regina Hagen Kolenda
Composition: Vicki Hochstedler
Copyeditor: Margaret Hill
Proofreader: Helen Maggie Carr
Prepress and Manufacturing: Caroline Fell

Access the latest information about Addison-Wesley titles from our World Wide Web Site: http://www.aw.com/cs

Library of Congress Cataloging-in-Publication Data
Snyder, Lawrence.
Fluency with information technology : skills, concepts, and capabilities / Lawrence Snyder.— Preliminary ed.
p. cm.
ISBN 0-321-12201-1
1. Information technology. I. Title.

T58.5 .S645 2003
004—dc21

2002026034

ISBN 0-321-12201-1

1 2 3 4 5 6 7 8 9 10—PA—040302

For Julie Ann, Dan, and Dave

Acknowledgments

Many people have contributed to this work. First to be thanked are my collaborators in the creation of the Fluency concept, the NRC Committee on Computer Literacy—Al Aho, Marcia Linn, Arnie Packer, Allen Tucker, Jeff Ullman, and Andy van Dam. Special thanks go to Herb Lin of the NRC staff who assisted throughout the Fluency effort, tirelessly and in his usual great good humor. Two enthusiastic supporters of Fluency— Bill Wulf of the National Academy of Engineering and John Cherniavski of the National Science Foundation—have continually supported this effort in more ways than I am aware. It has been a pleasure to know and work with this team.

As the material was developed for this book, many have contributed: Ken Yasuhara and Brian Bannon, teaching assistants on the first offering of CSE100, contributed in innumerable ways and injected a needed dose of practicality into my ideas. Grace Whiteaker, Alan Borning, and Batya Friedman have been generous with their ideas regarding Fluency. Martin Dickey of CSE and Mark Donovan of UWired have been constant sources of support. The original offerings of CSE100 benefited greatly from contributions by Nana Lowell, Anne Zald, Mike Eisenberg, and Fred Videon. Frank Tompa was extremely helpful with the *OED* material.

I am particularly grateful for the keen insights and valuable feedback from the reviewers of this project: Robert M. Aiken, Temple University; Anne Condon, University of British Columbia; Nicholas Cravotta, University of California, Berkeley; Gordon Davies, Open University; Peter J. Denning, George Mason University; John P. Dougherty, Haverford College; Philip East, University of Northern Iowa; Michael B. Eisenberg, University of Washington; Robert S. Fenchel, University of Wisconsin, Madison; Esther Grassian, UCLA College Library; Raymond Greenlaw, Armstrong Atlantic State University; A. J. Hurst, Monash University, Australia; Doris K. Lidtke, Towson University; Wen Liu, ITT Technical Institute; Jim McKeown, Dakota State University; David R. Musicant, Carleton College; Laurie J. Patterson, University of North Carolina, Wilmington; John Rosenberg, Monash University, Australia; Robert T. Ross, California Polytechnic State University; Robert J. Shive, Jr., Millsaps College; and Mark Urban-Lurain, Michigan State University.

The Fluency material has been a topic of discussion with many international colleagues. Discussions with Hans Hinterberger, John Rosenberg, and John Hirsch have been especially valuable in critiquing the material from an overseas perspective. Other helpful international commentary came from Anne Condon, Hannes Jonsson,

Jerg Nievergelt, Clark Thomberson, Barbara Thomberson, Ewan Tempero, and Kazuo Iwama.

Among the many thoughtful computer users who have either generously described their misunderstandings about IT or patiently listened to my explanations about IT, I wish to thank Esther Snyder, Helene Fowler, Judy Watson, Brendan Healey, Victory Grund, Shelley Burr, Ken Burr, and Noelle Lamb.

It is my great pleasure to thank my editors, Susan Hartman Sulllivan and Mary Clare McEwing. Susan's excellent judgment and insight, coupled with her enthusiasm for this book, have kept me focused amid a multitude of distractions. Mary Clare has been an excellent colleague. Her high standards for literate presentation have taken on the beast of technological jargon and style and been victorious. The day-to-day exchange of bytes and tastes with her has been a particular joy to me. Others of the Addison-Wesley team to whom I owe thanks are Michael Hirsch, Regina Kolenda, Kim Ellwood, Galia Shobry, and Lesly Hershman. Thanks especially to Vicki Hochstedler for her patience in working across a dozen time zones.

Finally, my wife Julie, and sons, Dan and Dave, have been patient, encouraging, and, most important, a continual source of good humor throughout this effort. It is with my deepest appreciation that I thank them for everything.

The names of the people used in the examples are 100% fictitious.

Preface

The overarching goal of *Fluency with Information Technology* is to give students the experience, knowledge, and capabilities needed to apply information technology effectively throughout their lives. Such an ambitious goal eclipses the more modest objectives of the computer literacy syllabus, which has traditionally focused only on imparting immediately useful skills. Fluency *contains* literacy, but adds problem solving, reasoning, complexity management, and other higher-level thinking processes, as well as a broader coverage of technological topics that prepare students to keep pace with the ever-advancing technology. The result is an intellectually rich curriculum constructed from equal parts of application proficiency, timeless knowledge, and thinking.

The inspiration for *Fluency with Information Technology* comes from a report by the National Research Council, *Being Fluent with Information Technology*.[1] In that study, commissioned by the National Science Foundation, the authors asserted that traditional computer literacy classes do not have the "staying power" necessary for students to keep pace with the rapid changes in information technology (IT). They concluded that the educational "bar needs to be raised" if students' knowledge is to evolve and adapt. The recommended alternative, dubbed *fluency with information technology,* or *FIT*, described a tripartite formulation of skills, concepts, and capabilities wrapped around a project-oriented delivery mechanism that assures the content is fully integrated. *Fluency with Information Technology* is designed to implement the Fluency proposal as outlined in the NRC study.

To present an overview of *Fluency*'s content and the rationale for its approach, consider the following four topics:

- Tripartite Content
- Projects as a Delivery Medium
- Debate about Programming
- Lifelong Learning

[1] *Being Fluent with Information Technology*, National Academy Press, 1999. [The author chaired the NRC's Committee on Computer Literacy that produced this report.]

Tripartite Content

The NRC's *Being Fluent* report asserted that to be successful, IT education must prepare students to adapt to an ever-changing IT landscape. It envisioned them engaged in a process of lifelong learning, adapting to and exploiting the new opportunities presented by the perfectly malleable media of information, computers, and networks. In order to launch the lifelong learning process students should be taught three different types of content—Skills, Concepts, and Capabilities.

Skills refers to proficiency with contemporary computer applications such as email, word processing, Web searching, and so on. Skills make the technology immediately useful to students and ground their learning of the other content in practical experience. The Skills component approximates traditional computer literacy topics, and is time varying. As students arrive at college knowing say, browsing, it is dropped from the curriculum and a new application can be added instead.

Concepts refers to the fundamental knowledge underpinning IT, such as how a computer works, digital representation of information, assessing information authenticity, and the like. Concepts provide the principles on which students will build new understanding as IT evolves. The Concepts component has been selected carefully from the curricula of the underlying IT disciplines, and is essentially timeless.

Capabilities refers to higher-level thinking processes such as problem solving, reasoning, complexity management, troubleshooting, and so on. Capabilities embody modes of thinking that are essential to exploiting IT, but they apply in many other situations as well. The Capabilities component is a standard element of all education, of course, but it is essential to the effective use of IT, making it an explicit focus of *Fluency with Information Technology.*

For each component the NRC report listed ten recommended items. These are shown in the following table.

The NRC's list of top 10 Skills, Concepts and Capabilities.

Fluency with Information Technology		
Skills	**Concepts**	**Capabilities**
Set up a personal computer	Fundamentals of computers	Engage in sustained reasoning
Use basic operating system facilities	Organization of information systems	Manage complexity
Use a word processor to create a document	Fundamentals of networks	Test a solution
Use a graphics or artwork package to manipulate an image	Digital representation of information	Find problems in a faulty use of IT
Connect a computer to the Internet	Structuring information	Navigate a collection and assess the quality of the information
Use the Internet to locate information	Modeling and abstraction	Collaborate using IT
Use a computer to communicate with others	Algorithmic thinking and programming	Communicate using IT about IT
Use a spreadsheet to model a simple process	Universality	Expect the unexpected
Use a database to access information	Limitations of Information Technology	Anticipate technological change
Use online help and instructional materials	Social impact of computers and technology	Think abstractly about Information Technology

Projects as a Delivery Medium

The Skills, Concepts, and Capabilities represent different kinds of knowledge that are co-equal in their contribution to IT Fluency. They span separate dimensions of

understanding. Rather than attempting to teach them separately, however, the NRC report recommends the material be integrated using a project-centric approach. *Fluency with Information Technology* adopts this technique.

The overall strategy is to focus the Skills instruction in the lab, the Concepts instruction in standard lecture material, and the Capabilities instruction in lecture/lab demonstrations. The projects are the opportunity to use the three kinds of knowledge for a specific purpose. They illustrate IT as it is often applied in practice—to solve information-processing tasks of a substantial nature involving non-obvious complexities.

A project is a multiweek assignment to achieve a specific IT goal. For example, create a database to track patient testing for a walk-in Drug and HIV Testing Clinic, and give a presentation to convince an audience that patient privacy has been preserved. (The NRC report lists a series of projects.[2]) Students, who can work in groups as appropriate, will employ a variety of Skills such as using a database, Web searching and presentation facilities. They will rely on their understanding of Concepts such as database keys, table structure, and the Join query operator. They will use Capabilities such as reasoning, debugging, complexity management, testing, and others. The Fluency components will be applied together to produce the final result, leading to an integrated understanding of IT and preparing for significant "real-life" applications of IT.

Debate about Programming

Since the advent of computer literacy syllabi nearly three decades ago, there has been debate as to whether non-specialists should be taught programming. Rational arguments have been offered on both sides, and thoughtful, well-intentioned adherents espouse each point of view. The NRC's study received strong input on both sides of the issue, but in the final report the definition of Fluency specifies "algorithmic thinking and programming" as one of the ten essential Concepts. The topic is sufficiently controversial that the original rationale and the strategy used in *Fluency with Information Technology* should be reviewed.

The NRC committee's report acknowledges the dilemma about programming in the opening to a long discussion[3] as to why programming was, in fact, finally included:

> Algorithmic thinking is a valuable ability for many educated people, yet programming—the act of expressing an algorithm in a specific form to solve a problem—is widely seen as the purview of the specialist. Algorithmic thinking and programming were listed together as a concept because they are very closely related forms of the same phenomenon.

The discussion makes clear that the committee is *not* claiming one must be a professional programmer to be Fluent. Its intentions are much more limited. The committee defines a modest set of basic programming ideas—variable, conditional, iteration, and so on—that are sufficient for programming's role in the Fluency context. Programming's role, as the report makes clear, is to support algorithmic thinking, reasoning, debugging, and other components of Fluency. The recommended programming concepts are essential to this use, but are insufficient to produce a practicing programmer. Thus, the committee might have more precisely specified, "algorithmic thinking and related programming concepts essential to algorithm specification."

---

[2] *Being Fluent*, Appendix A.

[3] Ibid., pp. 41–48.

Since the report recommends only a handful of basic programming concepts, *Fluency with Information Technology* treats only that set. Nevertheless, the perception that programming is a difficult and intellectually challenging topic suitable only for "techies" raises the question of whether even this small set of concepts can be taught to a general student population. Moreover, since programming is usually taught to students with a strong mathematics background, one wonders if it isn't hopeless to expect "non-techie" students to learn it. The answer, based on the experience of teaching from early drafts of *Fluency with Information Technology,* is that it *can* be done and the students find it rewarding.

Specifically, the considerations that lead to success are three in number. First, programming must be taught in a patient, accessible way. The general student population does not find it intuitive, and so needs more complete explanations than would be given to a "techie" audience. Second, the programs must be non-intimidating and be applied to problems of interest to the class. Students expect to find programming terrible, and are amazed when they have early success on an interesting problem. Third, the "mathematical reasoning" expectations assumed in standard programming classes must be replaced with an expectation of "reasoning" ability only. This is sufficient. Programming is purely a reasoning activity and is within the ability level of college students. It relies only incidentally on mathematics. Similar success has been observed in related situations[4].

With a clear understanding of the report's intent regarding programming, and evidence that *Fluency's* gentler treatment makes programming accessible to the general student population, the final concern is whether including programming in the Fluency curriculum is worthwhile. This is an obvious concern because the amount of programming is too limited to produce programmers, and the time might be better spent on some other content. Without long-term studies it is impossible to say definitively, but there is some evidence that the answer will be "yes." First, the report's claim that programming knowledge facilitates other aspects of Fluency seems correct. From comprehending the Fetch/Execute Cycle to debugging, programming knowledge can be observed to assist students in understanding and to deepen their level of comprehension. Second, the inclusion of programming assures that the course is a substantive, ideas-based course. It is an intellectually challenging class equivalent to other "science classes," but without a heavy reliance on mathematics. Students report the class "is intellectually challenging" and that it "expanded my thinking." Third, succeeding with programming is an intellectual achievement that aids a student's confidence as a computer user. Confident users are the goal of Fluency, and in the limited cases studied so far, the goal has been reached.

Lifelong Learning

The students registering for a Fluency with Information Technology course can have a variety of expectations, which rarely include the course's actual objectives. Some are expecting to learn the popular computer applications and Web skills. Others, who already know these skills, are expecting an easy class and a good grade. For both of these groups a classic literacy course would fulfill their needs with less effort. For students with broader, less specific goals, Fluency is likely a better choice because it prepares students to use computers today and to be effective IT users forever through lifelong learning.

To appreciate the magnitude of the task of teaching the essential IT knowledge for life, consider that as a "product" a college education has an expected useful lifetime

[4] Ibid., Appendix B.

of 55 years for men, based on a life expectancy of 77 years, and slightly longer for women. From that perspective what should the Class of '44 (the students graduating 55 years prior to the publication of the *Being Fluent* report) have been taught, given the following facts:

- The first electronic computer would not be completed for two more years.
- The first computer network would be created for their 25th reunion.
- The term *personal computer* would take 35 years to be invented.
- The World Wide Web would be effectively a half-century away.

Clearly, more wonders from IT can be expected, and predicting what they will be is hopeless. Focusing on fundamental principles, foundational ideas, and higher-level thinking skills is the key to preparing graduates for the task of keeping pace with future developments. Thus, the committee has emphasized a fundamentals-rich approach.

Will the *Being Fluent* report's recommendations achieve the desired goal of launching lifelong learning? The question cannot be answered definitively before long-term studies are complete, so anecdotes must suffice. There are those (non-technical) people who are proficient computer users and make effective use of the technology based entirely on their own understanding and the new knowledge they are able to teach themselves. The common thread among the handful of people of my acquaintance who meet this definition of Fluent Citizen is that they learned the fundamentals of Information Technology, especially programming. Then, once they began using personal computers to solve some personally relevant problem, they continued steadily and incrementally to expand their knowledge. Their success became an incentive to apply IT more broadly and provided the confidence to believe that accomplishing the goal was within their grasp. This is Fluency, and it is the goal of *Fluency with Information Technology*.

A Word to Students about *Fluency with Information Technology*

Fluency with information technology is a somewhat unusual subject, making this a somewhat unusual textbook. There are three things I think you should know about using *Fluency With Information Technology:*

- Part of the Fluency content comes from labs, and is not described in this book.
- Though you must spend significant time in the lab, the "book learning" part is the most important.
- Read this book with a computer at hand.

Let me amplify on this advice.

Of the three kinds of knowledge defining Fluency—Skills, Concepts and Capabilities—the skills are the most likely to change. Writing about a software system in a textbook is useless to students if the software changes before the book is printed. Keeping such information current is nearly impossible. But it's also true that learning skills is best done on a computer in a lab using contemporary applications. You can use the latest and greatest software installed, and there is usually someone nearby to whom you can ask questions when you get stuck. Consequently, almost all of the skills in the Fluency curriculum will be learned in the lab, not with this book. What *can* be taught in a book is *how to learn* a new application, since there are guidelines about learning new software. These principles are the subject of Chapter 3. So, *Fluency with Information Technology* does treat skills, but it does so at a more

general level than for any particular computer software system. I do expect that you will be learning skills in the lab, so the chapters of the book assume more and more familiarity with computer applications as they progress.

Though the skills are taught in the lab, the concepts and capabilities are the subject of this book. They are the more important material, since they are the foundations on which you will learn. It's easy to be distracted by the "hands on" part of Fluency, because it can be fun, active, and practically useful in your other classes. But don't forget to read the book. The book explains the complex parts of Fluency in a patient, understandable way, enabling you to apply them in the lab. It is always easiest and fastest to read about the concepts first before attempting a project or assignment. I know it's obvious advice, but so many ignore it for too long.

In writing this book on my laptop I obviously had a computer always available to explore the Web or test out an example. To get the most out of the text you too will need a computer for about half of the content. As you read you should "follow along online," often reproducing the running example so that you can watch active aspects of it that cannot be printed in a book. Every section in which I intend for you to have a computer available while reading will be clearly marked by a computer icon,  .

Finally, good luck! Writing this book has truly been a pleasure. I hope reading it is equally enjoyable.

Fluency with Information Technology

The Master said: "To learn something and then put it into practice at the right time. Is this not a joy?"

—The Analects of Confucius

Contents