SCHAUM'S ouTlines

PRINCIPLES OF COMPUTER SCIENCE

CARL REYNOLDS, Ph.D. PAUL TYMANN, M.S.

- Introductions to the theoretical foundations of computer science
- 250 fully solved problems
- Hundreds more practice problems with answers

TRUSTED BY MORE THAN 40 MILLION STUDENTS

Use with these courses: ✓ Introduction to Computer Science ✓ Theory of Computation ✓ Computer Ethics

SCHAUM'S OUTLINE OF

Principles of COMPUTER SCIENCE

CARL REYNOLDS
Department of Computer Science
Rochester Institute of Technology

PAUL TYMANN
Department of Computer Science
Rochester Institute of Technology

Schaum's Outline Series

McGRAW-HILL
New York | Chicago | San Francisco | Lisbon | London | Madrid
Mexico City | Milan | New Delhi | San Juan
Seoul | Singapore | Sydney | Toronto

CARL REYNOLDS teaches courses in database, operating systems, programming, and programming language theory in the Computer Science Department at the Rochester Institute of Technology. He has taught at the college level for 10 years, and in the computer industry for 4 years. Before coming to RIT, Dr. Reynolds spent 19 years in the computer industry working in technical and training capacities for both hardware and software suppliers, and 6 years with a Dow Jones Industrial manufacturer creating expert systems for machine control. His interests include genetic algorithms, expert systems, and image processing.

PAUL TYMANN is Professor and Chair of the Computer Science Department at the Rochester Institute of Technology. He has taught both basic and advanced programming techniques for over 15 years. More recently he has been involved with development of a new bioinformatics program at RIT. Prior to entering academia, Professor Tymann worked in industry developing control software for point-of-sale terminals. For the past 5 years he has worked in the area of bioinformatics and has completed joint software development projects at the University of Rochester and Rutgers University.

The **McGraw·Hill** *Companies*

Schaum's Outline of Principles of
COMPUTER SCIENCE

1 2 3 4 5 6 7 8 9 0 CUS/CUS 0 1 4 3 2 1 0 9 8

ISBN 978-0-07-146051-4
MHID 0-07-146051-9

Library of Congress Cataloging-in-Publication Data

Reynolds, Carl, date.
Schaum's outline of principles of computer science / Carl Reynolds, Paul Tymann.
p. cm.—(Schaum's outline series)
Includes index.
ISBN 978-0-07-146051-4
1. Computer science—Problems, exercises, etc. I. Tymann, Paul T. II. Title.
QA76.28.R49 2008
004.076—dc22 2007041732

CONTENTS

CHAPTER 1

Introduction to Computer Science

WHAT IS COMPUTER SCIENCE?

Computer Science is defined in different ways by different authors. Wikipedia (http://en.wikipedia.org/wiki/Computer_science) defines computer science as the collection of a variety of disciplines related to computing, both theoretical and practical: theoretical foundations of information and computation, language theory, algorithm analysis and development, implementation of computing systems, computer graphics, databases, data communications, etc.

The US National Coordination Office for *Networking and Information Technology Research and Development* (NITRD) defines computer science in a similarly broad way:

> the systematic study of computing systems and computation. The body of knowledge resulting from this discipline contains theories for understanding computing systems and methods; design methodology, algorithms, and tools; methods for the testing of concepts; methods of analysis and verification; and knowledge representation and implementation. (http://www.nitrd.gov/pubs/bluebooks/1995/section.5.html)

Another broad definition comes from the *Association for Computing Machinery* (ACM) Model Curriculum. It says that computer science is the "study of computers and algorithmic processes, including their principles, their hardware and software design, their applications, and their impact on society."

A famous definition of computer science by Gibbs and Tucker (Gibbs and Tucker, "A Model Curriculum for a Liberal Arts Degree in Computer Science," *Comm. of the ACM*, vol. 29, no. 3, March 1986) emphasizes algorithm development and analysis as the central focus of computer science.

It's also a fair question to ask, "How is computer science a science?" In contrast to physics, biology, and chemistry, computer science is not based on the study of the natural world. In that sense, computer science is more like mathematics than science. Some argue that computer science is really computer art (where "art" means practice). On the other hand, computer scientists do use the scientific method to propose and test hypotheses, and some very nonobvious discoveries in computer science have important real-world implications. An example, which we will discuss later, is the discovery that some important problems simply cannot be solved by computation.

Despite many variations, essentially all definitions of computer science emphasize the study of algorithms. Algorithms, in one form or another, are central to computer science. Computer science combines the theoretical concepts of algorithm design and analysis with the practical considerations of how to implement algorithms on a computer and solve practical problems.