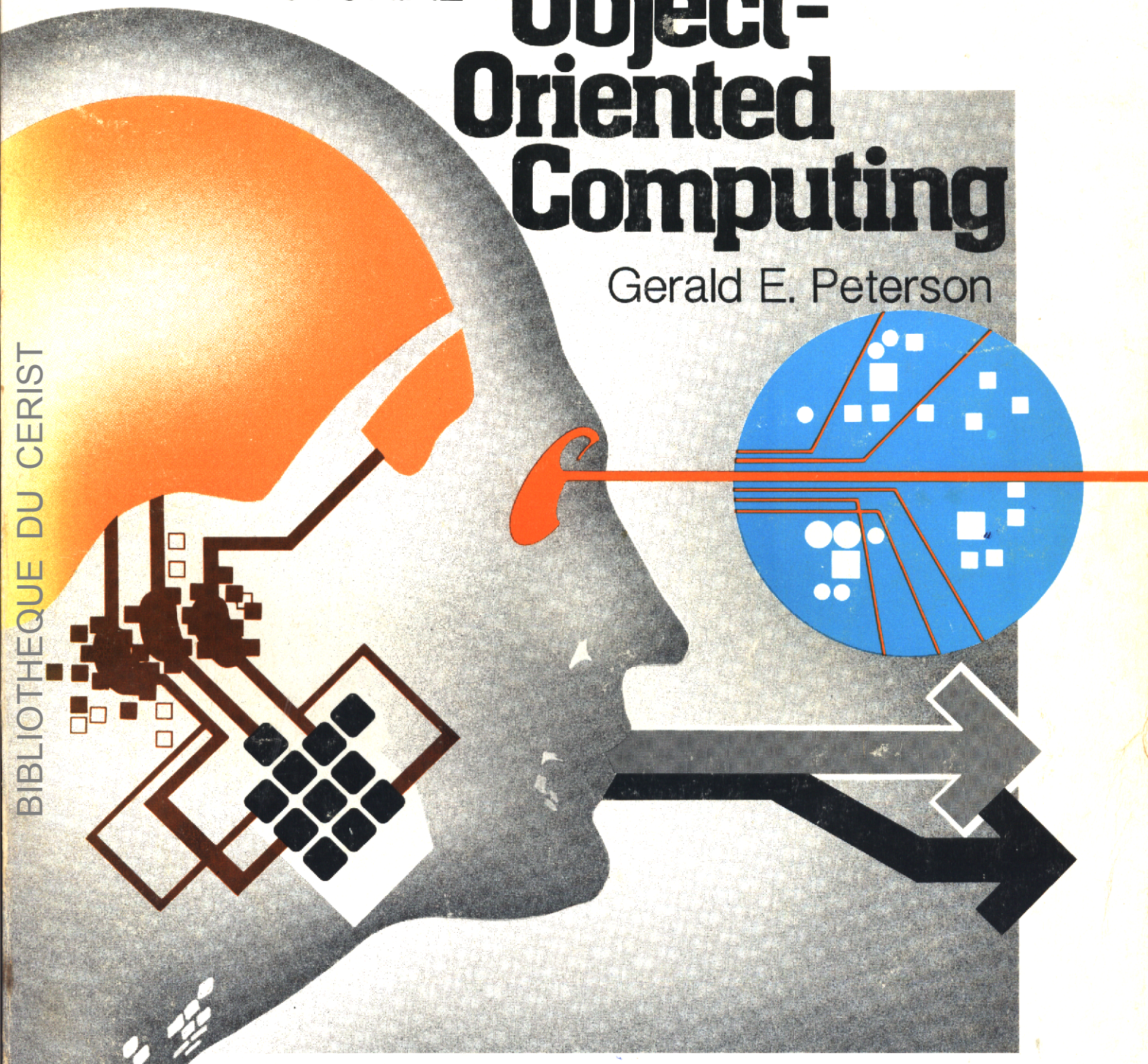
Volume 1: Concepts

TUTORIAL: Object-Oriented Computing

Gerald E. Peterson



COMPUTER SOCIETY ORDER NUMBER 821
LIBRARY OF CONGRESS NUMBER 87-80433
IEEE CATALOG NUMBER EH0257-6
ISBN 0-8186-0821-8
SAN 264-620X

THE COMPUTER SOCIETY OF THE IEEE

IEEE THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

IEEE COMPUTER SOCIETY PRESS

Volume 1: Concepts

TUTORIAL: Object-Oriented Computing

Gerald E. Peterson

COMPUTER SOCIETY ORDER NUMBER 821
LIBRARY OF CONGRESS NUMBER 87-80433
IEEE CATALOG NUMBER EH0257-6
ISBN 0-8186-0821-8
SAN 264-620X

THE COMPUTER SOCIETY OF THE IEEE

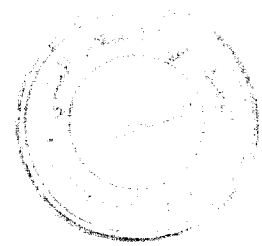
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

COMPUTER SOCIETY PRESS

Published by Computer Society Press of the IEEE
1730 Massachusetts Avenue, N.W.
Washington, D.C. 20036-1903

Cover designed by Jack I. Ballestero

Computer Society of the IEEE Order Number 821
Library of Congress Number 87-80433
IEEE Catalog Number EH0257-6
ISBN 0-8186-0821-8 (Paper)
ISBN 0-8186-4821-X (Microfiche)
SAN 264-620X



Order from: Computer Society of the IEEE
Terminal Annex
Post Office Box 4699
Los Angeles, CA 90080

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331

Computer Society of the IEEE
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC

Preface

Object-oriented computing is a style of computing in which data and associated procedures are encapsulated to form an *object*. An object is a useful computing entity existing at a higher level than procedures or data structures.

Object-oriented computing has gained considerable interest in the last few years. This interest is being fanned by the U. S. Department of Defense's push of Ada, a common (partially) object-oriented language.

This tutorial should be of interest to that group of computer professionals who have heard the term "object-oriented," know it is important, and want to obtain a more substantial understanding of the concept. This group includes:

- software engineers,
- AI professionals,
- professors and students of programming language and computer architecture courses,
- those who are building or proposing new computer languages or architectures.

Most of the papers are at the level of the practicing computer engineer. Some papers, especially those in Chapters I and II of Volume 1, are accessible to the computer-knowledgeable layman.

All aspects of object-oriented computing are considered, including object-oriented languages, object-oriented design and development, examples of object-oriented programming, object-oriented databases, and object-oriented computer architectures.

Volume 1 presents the basic concepts of object-oriented programming and describes several object-oriented languages.

An overview of the fundamental ideas of object-oriented programming is given in Chapter 1.

Smalltalk is a language based entirely on the use of objects. Everything in the language, including integers, is an object. A great deal of knowledge about object-oriented computing can be obtained by studying Smalltalk. Chapter 2 is devoted exclusively to this language.

Other languages are supportive of the object philosophy. These languages include Ada, in which objects can be created as packages; Modula-2, in which objects are created as modules; and some dialects of Lisp. The manner in which the object-oriented philosophy can be incorporated in these languages is considered in Chapters 3 and 4.

Volume 2 is devoted to the manner in which programs are implemented using object-oriented methods, and the manner in which object-oriented languages are themselves implemented.

It has been found that many problems decompose naturally into objects and messages that pass between them. Object-oriented development is the process of decomposing a problem into objects and messages and maintaining these structures in the implementation. Object-oriented development techniques and examples of their use are presented in Chapter 1.

Much insight into the nature of object-oriented programming can be obtained by studying examples where it was successfully put to use. Several examples of object-oriented programming are considered in Chapter 2.

The biggest problem with languages that fully support the object-oriented philosophy is efficient implementation. On a dedicated processor costing in the neighborhood of $100K, Smalltalk runs satisfactorily. On less expensive machines, however, Smalltalk implementations have been too slow to be practical for large programs. Chapter 3 contains many ideas about how to overcome this problem.

Several important concepts and issues that did not fit in other chapters are considered in Chapter 4.

Object-oriented computer architectures have also been studied and implemented in recent years. Chapter 5 describes principles, such as capability-based addressing, which have guided these implementations. Several examples of object-based architectures are described in Chapter 6. These include the Intel iAPX 432 and the IBM System/38.

Gerald E. Peterson

Acknowledgments

The author wishes to thank the McDonnell Douglas Astronautics and Aerospace Information Services Companies for providing a pleasant climate in which this work could be undertaken. This tutorial resulted from studies which were undertaken in order to improve the manner in which software is developed at McDonnell Douglas Corporation.

Table of Contents

Preface iii

Acknowledgments v

Chapter 1: Introduction to Object-Oriented Computing 1

Object-Oriented Software Systems 5
D. Robson (Byte, August 1981, pages 74–86)

Values and Objects in Programming Languages 9
B.J. MacLennan (SIGPLAN Notices, December 1982, pages 70–79)

Elements of Object-Oriented Programming 15
G.A. Pascoe (Byte, August 1986, pages 307–316)

Object-Oriented Programming 21
T. Rentsch (SIGPLAN Notices, September 1982, pages 51–57)

Software-IC's 28
L. Ledbetter and B. Cox (Byte, June 1985, pages 307–315)

Chapter 2: Smalltalk: An Object-Based Language 37

Microelectronics and the Personal Computer 41
A.C. Kay (Scientific American, September 1977, pages 231–244)

Experiences with Smalltalk-80™ for Application Development 52
T. Love (Proceedings, Softfair: A Conference on Software Development Tools, Techniques, and Alternatives, 1983, pages 61–65)

The Smalltalk-80 System 57
The Xerox Learning Research Group (Byte, August 1981, pages 36–48)

Design Principles Behind Smalltalk 70
D.H.H. Ingalls (Byte, August 1981, pages 286–298)

User-Oriented Descriptions of Smalltalk Systems 75
T.M.H. Reenskaug (Byte, August 1981, pages 148–166)

Experimental Prototyping in Smalltalk 82
J. Diederich and J. Milton (IEEE Software, May 1987, pages 50–64)

Chapter 3: Object-Oriented Programming in Ada 97

Using Ada as a Programming Language for Robot-Based Manufacturing Cells 99
R.A. Volz, T.N. Mudge, and D.A. Gal (IEEE Transactions on Systems, Man, and Cybernetics, November/December 1984, pages 863–878)

Object-Based Computing and the Ada Programming Language 115
G.D. Buzzard and T.N. Mudge (Computer, March 1985, pages 11–19)

On the Unification of Data and Program Abstraction in Ada 124
P. Wegner (Proceedings of the 10th Annual ACM Symposium on Principles of Programming Languages, 1983, pages 256–264)

Chapter 4: Object-Oriented Programming in Other Languages 133

History and Goals of Modula-2 137
N. Wirth (Byte, August 1984, pages 145–152)

Modular Programming in C: An Approach and an Example 142
K. Dutta (SIGPLAN Notices, March 1985, pages 9–15)

Message/Object Programming: An Evolutionary
Change in Programming Technology 150
B.J. Cox (IEEE Software, January 1984, pages 50–61)
Object-Oriented Languages for the Macintosh 162
K.J. Schmucker (Byte, August 1986, pages 177–185)
CommonLoops: Merging Lisp and Object-Oriented Programming 169
D.G. Bobrow, K. Kahn, G. Kiczales, L. Masinter, M. Stefik, and F. Zdybel (Proceedings of the Conference on Object-Oriented Programming Systems, Languages and Applications, 1986, pages 17–29)
Object-Oriented Programming: Themes and Variations 182
M. Stefik and D.G. Bobrow (The AI Magazine, Winter 1986, pages 40–62)

Author Biography 205