1. Record Nr. UNINA9910132468303321

Autore	Abelson Harold
Titolo	Structure and interpretation of computer programs / / Harold Abelson and Gerald Jay Sussman, with Julie Sussman ; foreword by Alan J. Perlis
Pubbl/distr/stampa	London : , : MIT Press, , [1996] ©1996
Edizione	[Second edition.]
Descrizione fisica	1 online resource (xxiii, 657 pages) : illustrations
Disciplina	005.13/3
Soggetti	Computer programming LISP (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Foreword xi -- Preface to the Second Edition xv -- Preface to the First Edition xvii -- Acknowledgments xxi -- 1 Building Abstractions with Procedures 1 (78) -- 1.1 The Elements of Programming 4 (27) -- 1.1.1 Expressions 5 (2) -- 1.1.2 Naming and the Environment 7 (2) -- 1.1.3 Evaluating Combinations 9 (2) -- 1.1.4 Compound Procedures 11 (2) -- 1.1.5 The Substitution Model for Procedure Application 13 (4) -- 1.1.6 Conditional Expressions and Predicates 17 (4) -- 1.1.7 Example: Square Roots by Newton's Method 21 (5) -- 1.1.8 Procedures as Black-Box Abstractions 26 (5) -- 1.2 Procedures and the Processes They Generate 31 (25) -- 1.2.1 Linear Recursion and Iteration 32 (5) -- 1.2.2 Tree Recursion 37 (5) -- 1.2.3 Orders of Growth 42 (2) -- 1.2.4 Exponentiation 44 (4) -- 1.2.5 Greatest Common Divisors 48 (2) -- 1.2.6 Example: Testing for Primality 50 (6) -- 1.3 Formulating Abstractions with Higher-Order Procedures 56 (23) -- 1.3.1 Procedures as Arguments 57 (5) -- 1.3.2 Constructing Procedures Using Lambda 62 (4) -- 1.3.3 Procedures as General Methods 66 (6) -- 1.3.4 Procedures as Returned Values 72 (7) -- 2 Building Abstractions with Data 79 (138) -- 2.1 Introduction to Data Abstraction 83 (14) -- 2.1.1 Example: Arithmetic Operations for Rational Numbers 83 (4) -- 2.1.2 Abstraction Barriers 87 (3) -- 2.1.3 What Is Meant by Data? 90 (3) -- 2.1.4 Extended Exercise: Interval Arithmetic 93 (4) -- 2.2 Hierarchical Data and the Closure Property 97 (45) -- 2.2.1 Representing Sequences

(45) -- 5.5.1 Structure of the Compiler 569 (5) -- 5.5.2 Compiling Expressions 574 (7) -- 5.5.3 Compiling Combinations 581 (6) -- 5.5.4 Combining Instruction Sequences 587 (4) -- 5.5.5 An Example of Compiled Code 591 (9) -- 5.5.6 Lexical Addressing 600 (3) -- 5.5.7 Interfacing Compiled Code to the Evaluator -- References 611 (8) -- List of Exercises 619 (2) -- Index 621.

Sommario/riassunto

Structure and Interpretation of Computer Programs has had a dramatic impact on computer science curricula over the past decade. This long-awaited revision contains changes throughout the text. There are new implementations of most of the major programming systems in the book, including the interpreters and compilers, and the authors have incorporated many small changes that reflect their experience teaching the course at MIT since the first edition was published. A new theme has been introduced that emphasizes the central role played by different approaches to dealing with time in computational models: objects with state, concurrent programming, functional programming and lazy evaluation, and nondeterministic programming. There are new example sections on higher-order procedures in graphics and on applications of stream processing in numerical programming, and many new exercises. In addition, all the programs have been reworked to run in any Scheme implementation that adheres to the IEEE standard.