

1. Record Nr.	UNINA9910154950803321
Autore	Deitel Paul J.
Titolo	C : how to program : with an introduction to C++ / / Paul Deitel, Harvey Deitel
Pubbl/distr/stampa	Harlow, England : , : Pearson, , [2016] ©2016
ISBN	1-292-11098-8
Edizione	[Eight, Global edition.]
Descrizione fisica	1 online resource (1,006 pages) : illustrations
Collana	How to program series
Disciplina	005.133
Soggetti	C (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover -- Title Page -- Copyright Page -- Contents -- Preface -- 1 Introduction to Computers, the Internet and the Web 33 -- 1.1 Introduction 34 -- 1.2 Hardware and Software 35 -- 1.2.1 Moore's Law 35 -- 1.2.2 Computer Organization 36 -- 1.3 Data Hierarchy 37 -- 1.4 Machine Languages, Assembly Languages and High-Level Languages 40 -- 1.5 The C Programming Language 41 -- 1.6 C Standard Library 42 -- 1.7 C++ and Other C-Based Languages 43 -- 1.8 Object Technology 44 -- 1.8.1 The Automobile as an Object 45 -- 1.8.2 Methods and Classes 45 -- 1.8.3 Instantiation 45 -- 1.8.4 Reuse 45 -- 1.8.5 Messages and Method Calls 46 -- 1.8.6 Attributes and Instance Variables 46 -- 1.8.7 Encapsulation and Information Hiding 46 -- 1.8.8 Inheritance 46 -- 1.9 Typical C Program-Development Environment 47 -- 1.9.1 Phase 1: Creating a Program 48 -- 1.9.2 Phases 2 and 3: Preprocessing and Compiling a C Program 48 -- 1.9.3 Phase 4: Linking 48 -- 1.9.4 Phase 5: Loading 49 -- 1.9.5 Phase 6: Execution 49 -- 1.9.6 Problems That May Occur at Execution Time 49 -- 1.9.7 Standard Input, Standard Output and Standard Error Streams 49 -- 1.10 Test-Driving a C Application in Windows, Linux and Mac OS X 49 -- 1.10.1 Running a C Application from the Windows Command Prompt 50 -- 1.10.2 Running a C Application Using GNU C with Linux 53 -- 1.10.3 Running a C Application Using the Terminal on Mac OS X 56 -- 1.11 Operating Systems 59 -- 1.11.1 Windows-A Proprietary Operating System 59 -- 1.11.2 Linux-An Open-Source Operating System 59 --

1.11.3 Apple's Mac OS X -- Apple's iOS for iPhone®, iPad® and iPod Touch® Devices 60 -- 1.11.4 Google's Android 60 -- 1.12 The Internet and World Wide Web 61 -- 1.12.1 The Internet: A Network of Networks 61 -- 1.12.2 The World Wide Web: Making the Internet User-Friendly 61 -- 1.12.3 Web Services 62 -- 1.12.4 Ajax 64 -- 1.12.5 The Internet of Things 64.

1.13 Some Key Software Terminology 64 -- 1.14 Keeping Up-to-Date with Information Technologies 66 -- 2 Introduction to C Programming 71 -- 2.1 Introduction 72 -- 2.2 A Simple C Program: Printing a Line of Text 72 -- 2.3 Another Simple C Program: Adding Two Integers 76 -- 2.4 Memory Concepts 80 -- 2.5 Arithmetic in C 81 -- 2.6 Decision Making: Equality and Relational Operators 85 -- 2.7 Secure C Programming 89 -- 3 Structured Program Development in C 101 -- 3.1 Introduction 102 -- 3.2 Algorithms 102 -- 3.3 Pseudocode 102 -- 3.4 Control Structures 103 -- 3.5 The if Selection Statement 105 -- 3.6 The if...else Selection Statement 106 -- 3.7 The while Iteration Statement 110 -- 3.8 Formulating Algorithms Case Study 1: Counter-Controlled Iteration 111 -- 3.9 Formulating Algorithms with Top-Down, Stepwise Refinement Case Study 2: Sentinel-Controlled Iteration 114 -- 3.10 Formulating Algorithms with Top-Down, Stepwise Refinement Case Study 3: Nested Control Statements 120 -- 3.11 Assignment Operators 124 -- 3.12 Increment and Decrement Operators 125 -- 3.13 Secure C Programming 127 -- 4 C Program Control 145 -- 4.1 Introduction 146 -- 4.2 Iteration Essentials 146 -- 4.3 Counter-Controlled Iteration 147 -- 4.4 for Iteration Statement 148 -- 4.5 for Statement: Notes and Observations 151 -- 4.6 Examples Using the for Statement 152 -- 4.7 switch Multiple-Selection Statement 155 -- 4.8 do...while Iteration Statement 161 -- 4.9 break and continue Statements 162 -- 4.10 Logical Operators 164 -- 4.11 Confusing Equality (==) and Assignment (=) Operators 167 -- 4.12 Structured Programming Summary 169 -- 4.13 Secure C Programming 174 -- 5 C Functions 189 -- 5.1 Introduction 190 -- 5.2 Modularizing Programs in C 190 -- 5.3 Math Library Functions 191 -- 5.4 Functions 193 -- 5.5 Function Definitions 193 -- 5.5.1 square Function 194 -- 5.5.2 maximum Function 197.

5.6 Function Prototypes: A Deeper Look 198 -- 5.7 Function Call Stack and Stack Frames 200 -- 5.8 Headers 204 -- 5.9 Passing Arguments By Value and By Reference 205 -- 5.10 Random Number Generation 206 -- 5.11 Example: A Game of Chance -- Introducing enum 210 -- 5.12 Storage Classes 214 -- 5.13 Scope Rules 216 -- 5.14 Recursion 219 -- 5.15 Example Using Recursion: Fibonacci Series 222 -- 5.16 Recursion vs. Iteration 226 -- 5.17 Secure C Programming 227 -- 6 C Arrays 246 -- 6.1 Introduction 247 -- 6.2 Arrays 247 -- 6.3 Defining Arrays 249 -- 6.4 Array Examples 249 -- 6.4.1 Defining an Array and Using a Loop to Set the Array's Element Values 249 -- 6.4.2 Initializing an Array in a Definition with an Initializer List 250 -- 6.4.3 Specifying an Array's Size with a Symbolic Constant and Initializing Array Elements with Calculations 252 -- 6.4.4 Summing the Elements of an Array 253 -- 6.4.5 Using Arrays to Summarize Survey Results 254 -- 6.4.6 Graphing Array Element Values with Histograms 256 -- 6.4.7 Rolling a Die 60,000,000 Times and Summarizing the Results in an Array 257 -- 6.5 Using Character Arrays to Store and Manipulate Strings 257 -- 6.5.1 Initializing a Character Array with a String 258 -- 6.5.2 Initializing a Character Array with an Initializer List of Characters 258 -- 6.5.3 Accessing the Characters in a String 258 -- 6.5.4 Inputting into a Character Array 258 -- 6.5.5 Outputting a Character Array That Represents a String 259 -- 6.5.6 Demonstrating Character Arrays 259 -- 6.6 Static Local Arrays and Automatic Local Arrays 260 -- 6.7

Passing Arrays to Functions 262 -- 6.8 Sorting Arrays 266 -- 6.9 Case Study: Computing Mean, Median and Mode Using Arrays 268 -- 6.10 Searching Arrays 273 -- 6.10.1 Searching an Array with Linear Search 273 -- 6.10.2 Searching an Array with Binary Search 274 -- 6.11 Multidimensional Arrays 278.

6.11.1 Illustrating a Double-Subscripted Array 278 -- 6.11.2 Initializing a Double-Subscripted Array 279 -- 6.11.3 Setting the Elements in One Row 281 -- 6.11.4 Totaling the Elements in a Two-Dimensional Array 281 -- 6.11.5 Two-Dimensional Array Manipulations 282 -- 6.12 Variable-Length Arrays² 285 -- 6.13 Secure C Programming 288 -- 7 C Pointers 306 -- 7.1 Introduction 307 -- 7.2 Pointer Variable Definitions and Initialization 308 -- 7.3 Pointer Operators 309 -- 7.4 Passing Arguments to Functions by Reference 311 -- 7.5 Using the const Qualifier with Pointers 315 -- 7.5.1 Converting a String to Uppercase Using a Non-Constant Pointer to Non-Constant Data 316 -- 7.5.2 Printing a String One Character at a Time Using a Non-Constant Pointer to Constant Data 317 -- 7.5.3 Attempting to Modify a Constant Pointer to Non-Constant Data 319 -- 7.5.4 Attempting to Modify a Constant Pointer to Constant Data 320 -- 7.6 Bubble Sort³ Using Pass-by-Reference 321 -- 7.7 sizeof Operator 324 -- 7.8 Pointer Expressions and Pointer Arithmetic 327 -- 7.8.1 Allowed Operators for Pointer Arithmetic 327 -- 7.8.2 Aiming a Pointer at an Array 327 -- 7.8.3 Adding an Integer to a Pointer 328 -- 7.8.4 Subtracting an Integer from a Pointer 328 -- 7.8.5 Incrementing and Decrementing a Pointer 328 -- 7.8.6 Subtracting One Pointer from Another 329 -- 7.8.7 Assigning Pointers to One Another 329 -- 7.8.8 Pointer to void 329 -- 7.8.9 Comparing Pointers 329 -- 7.9 Relationship between Pointers and Arrays 330 -- 7.9.1 Pointer/Offset Notation 330 -- 7.9.2 Pointer/Index Notation 331 -- 7.9.3 Cannot Modify an Array Name with Pointer Arithmetic 331 -- 7.9.4 Demonstrating Pointer Indexing and Offsets 331 -- 7.9.5 String Copying with Arrays and Pointers 332 -- 7.10 Arrays of Pointers 334 -- 7.11 Case Study: Card Shuffling and Dealing Simulation 335 -- 7.12 Pointers to Functions 340.

7.12.1 Sorting in Ascending or Descending Order 340 -- 7.12.2 Using Function Pointers to Create a Menu-Driven System 343 -- 7.13 Secure C Programming 345 -- 8 C Characters and Strings 365 -- 8.1 Introduction 366 -- 8.2 Fundamentals of Strings and Characters 366 -- 8.3 Character-Handling Library 368 -- 8.3.1 Functions isdigit, isalpha, isalnum and isxdigit 368 -- 8.3.2 Functions islower, isupper, tolower and toupper 371 -- 8.3.3 Functions isspace, iscntrl, ispunct, isprint and isgraph 372 -- 8.4 String-Conversion Functions 374 -- 8.4.1 Function strtod 374 -- 8.4.2 Function strtol 375 -- 8.4.3 Function strtoul 376 -- 8.5 Standard Input/Output Library Functions 376 -- 8.5.1 Functions fgets and putchar 377 -- 8.5.2 Function getchar 378 -- 8.5.3 Function sprintf 379 -- 8.5.4 Function sscanf 380 -- 8.6 String-Manipulation Functions of the String-Handling Library 381 -- 8.6.1 Functions strcpy and strncpy 382 -- 8.6.2 Functions strcat and strncat 382 -- 8.7 Comparison Functions of the String-Handling Library 383 -- 8.8 Search Functions of the String-Handling Library 385 -- 8.8.1 Function strchr 386 -- 8.8.2 Function strcspn 387 -- 8.8.3 Function strpbrk 387 -- 8.8.4 Function strrchr 388 -- 8.8.5 Function strspn 389 -- 8.8.6 Function strstr 389 -- 8.8.7 Function strtok 390 -- 8.9 Memory Functions of the String-Handling Library 391 -- 8.9.1 Function memcpy 392 -- 8.9.2 Function memmove 393 -- 8.9.3 Function memcmp 394 -- 8.9.4 Function memchr 394 -- 8.9.5 Function memset 395 -- 8.10 Other Functions of the String-Handling Library 395 -- 8.10.1 Function strerror 396 -- 8.10.2 Function strlen 396 -- 8.11 Secure C Programming 397 -- 9 C Formatted

Input/Output 409 -- 9.1 Introduction 410 -- 9.2 Streams 410 -- 9.3
Formatting Output with printf 410 -- 9.4 Printing Integers 411 -- 9.5
Printing Floating-Point Numbers 412 -- 9.5.1 Conversion Specifiers e,
E and f 413.
9.5.2 Conversion Specifiers g and G 413.

Sommario/riassunto

For courses in computer programming C How to Program is a comprehensive introduction to programming in C. Like other texts of the Deitels' How to Program series, the book serves as a detailed beginner source of information for college students looking to embark on a career in coding, or instructors and software-development professionals seeking to learn how to program with C. The Eighth Edition continues the tradition of the signature Deitel "Live Code" approach--presenting concepts in the context of full-working programs rather than incomplete snips of code. This gives students a chance to run each program as they study it and see how their learning applies to real world programming scenarios. MyProgrammingLab® not included. Students, if MyProgrammingLab is a recommended/mandatory component of the course, please ask your instructor for the correct ISBN and course ID. MyProgrammingLab should only be purchased when required by an instructor. Instructors, contact your Pearson representative for more information. MyProgrammingLab is an online homework, tutorial, and assessment program designed to work with this text to engage students and improve results. Within its structured environment, students practice what they learn, test their understanding, and pursue a personalized study plan that helps them better absorb course material and understand difficult concepts.
