1. Record Nr. UNINA9910151657403321

Autore	Gaddis Tony
Titolo	Starting Out with C++
Pubbl/distr/stampa	, : Pearson Education UK, , 2013 ©2013
ISBN	0-273-79403-5
Edizione	[8th ed.]
Descrizione fisica	1 online resource (1251 pages)
Altri autori (Persone)	WaltersJudy MugandaGodfrey
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Cover -- Contents -- Preface -- CHAPTER 1 Introduction to Computers and Programming -- 1.1 Why Program? -- 1.2 Computer Systems: Hardware and Software -- 1.3 Programs and Programming Languages -- 1.4 What Is a Program Made of? -- 1.5 Input, Processing, and Output -- 1.6 The Programming Process -- 1.7 Tying It All Together: Hi! It's Me -- CHAPTER 2 Introduction to C++ -- 2.1 The Parts of a C++ Program -- 2.2 The cout Object -- 2.3 The #include Directive -- 2.4 Standard and Prestandard C++ -- 2.5 Variables, Literals, and the Assignment Statement -- 2.6 Identifiers -- 2.7 Integer Data Types -- 2.8 Floating-Point Data Types -- 2.9 The char Data Type -- 2.10 The C++ string Class -- 2.11 The bool Data Type -- 2.12 Determining the Size of a Data Type -- 2.13 More on Variable Assignments and Initialization -- 2.14 Scope -- 2.15 Arithmetic Operators -- 2.16 Comments -- 2.17 Programming Style -- 2.18 Tying It All Together: Smile! -- CHAPTER 3 Expressions and Interactivity -- 3.1 The cin Object -- 3.2 Mathematical Expressions -- 3.3 Data Type Conversion and Type Casting -- 3.4 Overflow and Underflow -- 3.5 Named Constants -- 3.6 Multiple and Combined Assignment -- 3.7 Formatting Output -- 3.8 Working with Characters and Strings -- 3.9 Using C-Strings -- 3.10 More Mathematical Library Functions -- 3.11 Focus on Debugging: Hand Tracing a Program -- 3.12 Green Fields Landscaping Case Study-Part 1 -- 3.13 Tying It All Together: Word Game --

CHAPTER 4 Making Decisions -- 4.1 Relational Operators -- 4.2 The if Statement -- 4.3 The if/else Statement -- 4.4 The if/else if Statement -- 4.5 Menu-Driven Programs -- 4.6 Nested if Statements -- 4.7 Logical Operators -- 4.8 Validating User Input -- 4.9 More About Blocks and Scope -- 4.10 More About Characters and Strings -- 4.11 The Conditional Operator -- 4.12 The switch Statement -- 4.13 Enumerated Data Types.
4.14 Focus on Testing and Debugging: Validating Output Results -- 4.15 Green Fields Landscaping Case Study-Part 2 -- 4.16 Tying It All Together: Fortune Teller -- CHAPTER 5 Looping -- 5.1 Introduction to Loops: The while Loop -- 5.2 Using the while Loop for Input Validation -- 5.3 The Increment and Decrement Operators -- 5.4 Counters -- 5.5 The do-while Loop -- 5.6 The for Loop -- 5.7 Keeping a Running Total -- 5.8 Sentinels -- 5.9 Focus on Software Engineering: Deciding Which Loop to Use -- 5.10 Nested Loops -- 5.11 Breaking Out of a Loop -- 5.12 Using Files for Data Storage -- 5.13 Focus on Testing and Debugging: Creating Good Test Data -- 5.14 Central Mountain Credit Union Case Study -- 5.15 Tying It All Together: What a Colorful World -- CHAPTER 6 Functions -- 6.1 Modular Programming -- 6.2 Defining and Calling Functions -- 6.3 Function Prototypes -- 6.4 Sending Data into a Function -- 6.5 Passing Data by Value -- 6.6 The return Statement -- 6.7 Returning a Value from a Function -- 6.8 Returning a Boolean Value -- 6.9 Using Functions in a Menu-Driven Program -- 6.10 Local and Global Variables -- 6.11 Static Local Variables -- 6.12 Default Arguments -- 6.13 Using Reference Variables as Parameters -- 6.14 Overloading Functions -- 6.15 The exit() Function -- 6.16 Stubs and Drivers -- 6.17 Little Lotto Case Study -- 6.18 Tying It All Together: Glowing Jack-o-lantern -- CHAPTER 7 Introduction to Classes and Objects -- 7.1 Abstract Data Types -- 7.2 Object-Oriented Programming -- 7.3 Introduction to Classes -- 7.4 Creating and Using Objects -- 7.5 Defining Member Functions -- 7.6 Constructors -- 7.7 Destructors -- 7.8 Private Member Functions -- 7.9 Passing Objects to Functions -- 7.10 Object Composition -- 7.11 Focus on Software Engineering: Separating Class Specification, Implementation, and Client Code -- 7.12 Structures.
7.13 Home Software Company OOP Case Study -- 7.14 Introduction to Object-Oriented Analysis and Design -- 7.15 Screen Control -- 7.16 Tying It All Together: Yoyo Animation -- CHAPTER 8 Arrays -- 8.1 Arrays Hold Multiple Values -- 8.2 Accessing Array Elements -- 8.3 Inputting and Displaying Array Contents -- 8.4 Array Initialization -- 8.5 Processing Array Contents -- 8.6 Using Parallel Arrays -- 8.7 The typedef Statement -- 8.8 Arrays as Function Arguments -- 8.9 Two-Dimensional Arrays -- 8.10 Arrays with Three or More Dimensions -- 8.11 Vectors -- 8.12 Arrays of Objects -- 8.13 National Commerce Bank Case Study -- 8.14 Tying It All Together: Rock, Paper, Scissors -- CHAPTER 9 Searching, Sorting, and Algorithm Analysis -- 9.1 Introduction to Search Algorithms -- 9.2 Searching an Array of Objects -- 9.3 Introduction to Sorting Algorithms -- 9.4 Sorting an Array of Objects -- 9.5 Sorting and Searching Vectors -- 9.6 Introduction to Analysis of Algorithms -- 9.7 Case Studies -- 9.8 Tying It All Together: Secret Messages -- CHAPTER 10 Pointers -- 10.1 Pointers and the Address Operator -- 10.2 Pointer Variables -- 10.3 The Relationship Between Arrays and Pointers -- 10.4 Pointer Arithmetic -- 10.5 Initializing Pointers -- 10.6 Comparing Pointers -- 10.7 Pointers as Function Parameters -- 10.8 Pointers to Constants and Constant Pointers -- 10.9 Focus on Software Engineering: Dynamic Memory Allocation -- 10.10 Focus on Software Engineering: Returning Pointers from Functions -- 10.11 Pointers to Class Objects and Structures --

10.12 Focus on Software Engineering: Selecting Members of Objects -- 10.13 United Cause Relief Agency Case Study -- 10.14 Tying It All Together: Pardon Me, Do You Have the Time? -- CHAPTER 11 More About Classes and Object-Oriented Programming -- 11.1 The this Pointer and Constant Member Functions -- 11.2 Static Members. 11.3 Friends of Classes -- 11.4 Memberwise Assignment -- 11.5 Copy Constructors -- 11.6 Operator Overloading -- 11.7 Type Conversion Operators -- 11.8 Convert Constructors -- 11.9 Aggregation and Composition -- 11.10 Inheritance -- 11.11 Protected Members and Class Access -- 11.12 Constructors, Destructors, and Inheritance -- 11.13 Overriding Base Class Functions -- 11.14 Tying It All Together: Putting Data on the World Wide Web -- CHAPTER 12 More on C-Strings and the string Class -- 12.1 C-Strings -- 12.2 Library Functions for Working with C-Strings -- 12.3 Conversions Between Numbers and Strings -- 12.4 Writing Your Own C-String Handling Functions -- 12.5 More About the C++ string Class -- 12.6 Creating Your Own String Class -- 12.7 Advanced Software Enterprises Case Study -- 12.8 Tying It All Together: Program Execution Environments -- CHAPTER 13 Advanced File and I/O Operations -- 13.1 Input and Output Streams -- 13.2 More Detailed Error Testing -- 13.3 Member Functions for Reading and Writing Files -- 13.4 Binary Files -- 13.5 Creating Records with Structures -- 13.6 Random-Access Files -- 13.7 Opening a File for Both Input and Output -- 13.8 Online Friendship Connections Case Study: Object Serialization -- 13.9 Tying It All Together: File Merging and Color-Coded HTML -- CHAPTER 14 Recursion -- 14.1 Introduction to Recursion -- 14.2 The Recursive Factorial Function -- 14.3 The Recursive gcd Function -- 14.4 Solving Recursively Defined Problems -- 14.5 A Recursive Binary Search Function -- 14.6 Focus on Problem Solving and Program Design: The QuickSort Algorithm -- 14.7 The Towers of Hanoi -- 14.8 Focus on Problem Solving: Exhaustive and Enumeration Algorithms -- 14.9 Focus on Software Engineering: Recursion versus Iteration -- 14.10 Tying It All Together: Infix and Prefix Expressions -- CHAPTER 15 Polymorphism and Virtual Functions. 15.1 Type Compatibility in Inheritance Hierarchies -- 15.2 Polymorphism and Virtual Member Functions -- 15.3 Abstract Base Classes and Pure Virtual Functions -- 15.4 Focus on Object-Oriented Programming: Composition versus Inheritance -- 15.5 Secure Encryption Systems, Inc., Case Study -- 15.6 Tying It All Together: Let's Move It -- CHAPTER 16 Exceptions, Templates, and the Standard Template Library (STL) -- 16.1 Exceptions -- 16.2 Function Templates -- 16.3 Class Templates -- 16.4 Class Templates and Inheritance -- 16.5 Introduction to the Standard Template Library -- 16.6 Tying It All Together: Word Transformers Game -- CHAPTER 17 Linked Lists -- 17.1 Introduction to the Linked List ADT -- 17.2 Linked List Operations -- 17.3 A Linked List Template -- 17.4 Recursive Linked List Operations -- 17.5 Variations of the Linked List -- 17.6 The STL list Container -- 17.7 Reliable Software Systems, Inc., Case Study -- 17.8 Tying It All Together: More on Graphics and Animation -- CHAPTER 18 Stacks and Queues -- 18.1 Introduction to the Stack ADT -- 18.2 Dynamic Stacks -- 18.3 The STL stack Container -- 18.4 Introduction to the Queue ADT -- 18.5 Dynamic Queues -- 18.6 The STL deque and queue Containers -- 18.7 Focus on Problem Solving and Program Design: Eliminating Recursion -- 18.8 Tying It All Together: Converting Postfix Expressions to Infix -- CHAPTER 19 Binary Trees -- 19.1 Definition and Applications of Binary Trees -- 19.2 Binary Search Tree Operations -- 19.3 Template Considerations for Binary Search Trees -- 19.4 Tying It All Together: Genealogy Trees -- Appendix A: The ASCII Character Set -- Appendix B: Operator Precedence and Associativity --

Appendix C: Answers to Checkpoints -- Appendix D: Answers to Odd-Numbered Review Questions -- Index -- A -- B -- C -- D -- E -- F -- G -- H -- I -- J -- K -- L -- M -- N -- O -- P -- Q -- R -- S -- T -- U. V.

Sommario/riassunto

Tony Gaddis's accessible, step-by-step presentation helps beginning students understand the important details necessary to become skilled programmers at an introductory level. Gaddis motivates the study of both programming skills and the C++ programming language by presenting all the details needed to understand the "how" and the "why" -but never losing sight of the fact that most beginners struggle with this material. His approach is both gradual and highly accessible, ensuring that students understand the logic behind developing high-quality programs. In Starting Out with C++: Early Objects, Gaddis covers objects and classes early after functions and before arrays and pointers. As with all Gaddis texts, clear and easy-to-read code listings, concise and practical real-world examples, and an abundance of exercises appear in every chapter. This text is intended for either a one-semester accelerated introductory course or a traditional two-semester sequence covering C++ programming.In Starting Out with C++: Early Objects, Gaddis covers objects and classes early after functions and before arrays and pointers. As with all Gaddis texts, clear and easy-to-read code listings, concise and practical real-world examples, and an abundance of exercises appear in every chapter. This text is intended for either a one-semester accelerated introductory course or a traditional two-semester sequence covering C++ programming. NEW! This edition is available with MyProgrammingLab, an innovative online homework and assessment tool. Through the power of practice and immediate personalized feedback, MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Note: If you are purchasing the standalone text or electronic version, MyProgrammingLab does not come automatically packaged with the text. To purchase MyProgrammingLab,
please visit: myprogramminglab.com or you can purchase a package of the physical text + MyProgrammingLab by searching the Pearson Higher Education web site. MyProgrammingLab is not a self-paced technology and should only be purchased when required by an instructor.