1.	Record Nr.	UNINA9910154948103321
	Autore	Gaddis Tony
	Titolo	Starting out with c++ : from control structures through objects / / Tony Gaddis
	Pubbl/distr/stampa	Harlow, England : , : Pearson Education, Limited, , [2016] Â©2016
	ISBN	1-292-11943-8
	Edizione	[Eighth edition.]
	Descrizione fisica	1 online resource (1,050 pages) : illustrations, charts, photographs
	Disciplina	005.133
	Soggetti	C++ (Computer program language)
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Nota di bibliografia	Includes bibliographical references and index.
	Nota di contenuto	Cover -- Title Page -- Copyright Page -- Supplements -- Acknowledgments -- About the Author -- 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 Procedural and Object-Oriented Programming -- 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 Variables and Literals -- 2.5 Identifiers -- 2.6 Integer Data Types -- 2.7 The char Data Type -- 2.8 The C++ string Class -- 2.9 Floating-Point Data Types -- 2.10 The bool Data Type -- 2.11 Determining the Size of a Data Type -- 2.12 Variable Assignments and Initialization -- 2.13 Scope -- 2.14 Arithmetic Operators -- 2.15 Comments -- 2.16 Named Constants -- 2.17 Programming Style -- CHAPTER 3 Expressions and Interactivity -- 3.1 The cin Object -- 3.2 Mathematical Expressions -- 3.3 When You Mix Apples and Oranges: Type Conversion -- 3.4 Overflow and Underflow -- 3.5 Type Casting -- 3.6 Multiple Assignment and Combined Assignment -- 3.7 Formatting Output -- 3.8 Working with Characters and string Objects -- 3.9 More Mathematical Library Functions -- 3.10 Focus on Debugging: Hand Tracing a Program -- 3.11 Focus on Problem Solving: A Case Study -- CHAPTER 4 Making Decisions -- 4.1 Relational Operators -- 4.2 The if

Statement -- 4.3 Expanding the if Statement -- 4.4 The if/else Statement -- 4.5 Nested if Statements -- 4.6 The if/else if Statement -- 4.7 Flags -- 4.8 Logical Operators -- 4.9 Checking Numeric Ranges with Logical Operators -- 4.10 Menus -- 4.11 Focus on Software Engineering: Validating User Input -- 4.12 Comparing Characters and Strings -- 4.13 The Conditional Operator.

4.14 The switch Statement -- 4.15 More About Blocks and Variable Scope -- CHAPTER 5 Loops and Files -- 5.1 The Increment and Decrement Operators -- 5.2 Introduction to Loops: The while Loop -- 5.3 Using the while Loop for Input Validation -- 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 Using Files for Data Storage -- 5.12 Optional Topics: Breaking and Continuing a Loop -- CHAPTER 6 Functions -- 6.1 Focus on Software Engineering: 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 Focus on Software Engineering: Using Functions in a Menu-Driven Program -- 6.7 The return Statement -- 6.8 Returning a Value from a Function -- 6.9 Returning a Boolean Value -- 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 -- CHAPTER 7 Arrays -- 7.1 Arrays Hold Multiple Values -- 7.2 Accessing Array Elements -- 7.3 No Bounds Checking in C++ -- 7.4 Array Initialization -- 7.5 The Range-Based for Loop -- 7.6 Processing Array Contents -- 7.7 Focus on Software Engineering: Using Parallel Arrays -- 7.8 Arrays as Function Arguments -- 7.9 Two-Dimensional Arrays -- 7.10 Arrays with Three or More Dimensions -- 7.11 Focus on Problem Solving and Program Design: A Case Study -- 7.12 If You Plan to Continue in Computer Science: Introduction to the STL vector -- CHAPTER 8 Searching and Sorting Arrays -- 8.1 Focus on Software Engineering: Introduction to Search Algorithms -- 8.2 Focus on Problem Solving and Program Design: A Case Study.

8.3 Focus on Software Engineering: Introduction to Sorting Algorithms -- 8.4 Focus on Problem Solving and Program Design: A Case Study -- 8.5 If You Plan to Continue in Computer Science: Sorting and Searching vectors -- CHAPTER 9 Pointers -- 9.1 Getting the Address of a Variable -- 9.2 Pointer Variables -- 9.3 The Relationship Between Arrays and Pointers -- 9.4 Pointer Arithmetic -- 9.5 Initializing Pointers -- 9.6 Comparing Pointers -- 9.7 Pointers as Function Parameters -- 9.8 Focus on Software Engineering: Dynamic Memory Allocation -- 9.9 Focus on Software Engineering: Returning Pointers from Functions -- 9.10 Using Smart Pointers to Avoid Memory Leaks -- 9.11 Focus on Problem Solving and Program Design: A Case Study -- CHAPTER 10 Characters, C-Strings, and More About the string Class -- 10.1 Character Testing -- 10.2 Character Case Conversion -- 10.3 C-Strings -- 10.4 Library Functions for Working with C-Strings -- 10.5 C-String/Numeric Conversion Functions -- 10.6 Focus on Software Engineering: Writing Your Own C-String-Handling Functions -- 10.7 More About the C++ string Class -- 10.8 Focus on Problem Solving and Program Design: A Case Study -- CHAPTER 11 Structured Data -- 11.1 Abstract Data Types -- 11.2 Focus on Software Engineering: Combining Data into Structures -- 11.3 Accessing Structure Members -- 11.4 Initializing a Structure -- 11.5 Arrays of Structures -- 11.6 Focus on Software Engineering: Nested Structures -- 11.7 Structures as Function Arguments -- 11.8 Returning a Structure from a Function -- 11.9 Pointers to Structures -- 11.10 Focus on Software Engineering:

Sommario/riassunto

For introductory courses in computer programming A Problem-Solving Approach to Programming In Starting Out with C++: From Control Structures through Objects, Brief Edition, Gaddis takes a problem-solving approach, inspiring students to understand the logic behind developing quality programs while introducing the C++ programming language. This style of teaching builds programming confidence and enhances each student's development of programming skills. This edition in the Starting Out With Series covers the core programming concepts that are introduced in the first semester introductory programming course. 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. The Eighth Edition is updated and revised to reflect changes to the C++ programming language. Also available with MyProgrammingLab™ This title is also available with MyProgrammingLab to help students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students who often struggle with the basic concepts and paradigms of popular high-level programming languages. MyProgrammingLab consists of hundreds of practice exercises

organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong-and why. For instructors, a comprehensive gradebook tracks students' submissions and provides educators a dynamic tool for monitoring individual and class performance. 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 product designed to personalize learning and improve results. With a wide range of interactive, engaging, and assignable activities, students are encouraged to actively learn and retain tough course concepts.