1.	Record Nr.	UNINA9910154775003321
	Autore	Liang Y. Daniel
	Titolo	Introduction to Java programming : brief version / / Y. Daniel Liang ; global edition contributions by Ming-Jyh Tsai
	Pubbl/distr/stampa	Boston : , : Pearson Education Limited, , [2015] ©2015
	ISBN	1-292-07857-X
	Edizione	[Tenth edition, Global edition.]
	Descrizione fisica	1 online resource (800 pages) : illustrations, tables
	Collana	Always Learning
	Disciplina	005.133
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Includes index.
	Nota di contenuto	Cover -- Title -- Copyright -- Chapter 1 Introduction to Computers, Programs,and Java -- 1.1 Introduction -- 1.2 What Is a Computer? -- 1.3 Programming Languages -- 1.4 Operating Systems -- 1.5 Java, the World Wide Web, and Beyond -- 1.6 The Java Language Specification, API, JDK, and IDE -- 1.7 A Simple Java Program -- 1.8 Creating, Compiling, and Executing a Java Program -- 1.9 Programming Style and Documentation -- 1.10 Programming Errors -- 1.11 Developing Java Programs Using NetBeans -- 1.12 Developing Java Programs Using Eclipse -- Chapter 2 Elementary Programming -- 2.1 Introduction -- 2.2 Writing a Simple Program -- 2.3 Reading Input from the Console -- 2.4 Identifiers -- 2.5 Variables -- 2.6 Assignment Statements and Assignment Expressions -- 2.7 Named Constants -- 2.8 Naming Conventions -- 2.9 Numeric Data Types and Operations -- 2.10 Numeric Literals -- 2.11 Evaluating Expressions and Operator Precedence -- 2.12 Case Study: Displaying the Current Time -- 2.13 Augmented Assignment Operators -- 2.14 Increment and Decrement Operators -- 2.15 Numeric Type Conversions -- 2.16 Software Development Process -- 2.17 Case Study: Counting Monetary Units -- 2.18 Common Errors and Pitfalls -- Chapter 3 Selections -- 3.1 Introduction -- 3.2 boolean Data Type -- 3.3 if Statements -- 3.4 Two-Way if-else Statements -- 3.5 Nested if and Multi-Way if-else Statements -- 3.6 Common Errors and Pitfalls -- 3.7 Generating Random Numbers -- 3.8 Case Study: Computing Body Mass Index --

3.9 Case Study: Computing Taxes -- 3.10 Logical Operators -- 3.11 Case Study: Determining Leap Year -- 3.12 Case Study: Lottery -- 3.13 switch Statements -- 3.14 Conditional Expressions -- 3.15 Operator Precedence and Associativity -- 3.16 Debugging -- Chapter 4 Mathematical Functions, Characters,and Strings -- 4.1 Introduction -- 4.2 Common Mathematical Functions.
4.3 Character Data Type and Operations -- 4.4 The String Type -- 4.5 Case Studies -- 4.6 Formatting Console Output -- Chapter 5 Loops -- 5.1 Introduction -- 5.2 The while Loop -- 5.3 The do-while Loop -- 5.4 The for Loop -- 5.5 Which Loop to Use? -- 5.6 Nested Loops -- 5.7 Minimizing Numeric Errors -- 5.8 Case Studies -- 5.9 Keywords break and continue -- 5.10 Case Study: Checking Palindromes -- 5.11 Case Study: Displaying Prime Numbers -- Chapter 6 Methods -- 6.1 Introduction -- 6.2 Defining a Method -- 6.3 Calling a Method -- 6.4 void Method Example -- 6.5 Passing Arguments by Values -- 6.6 Modularizing Code -- 6.7 Case Study: Converting Hexadecimals to Decimals -- 6.8 Overloading Methods -- 6.9 The Scope of Variables -- 6.10 Case Study: Generating Random Characters -- 6.11 Method Abstraction and Stepwise Refinement -- Chapter 7 Single-Dimensional Arrays -- 7.1 Introduction -- 7.2 Array Basics -- 7.3 Case Study: Analyzing Numbers -- 7.4 Case Study: Deck of Cards -- 7.5 Copying Arrays -- 7.6 Passing Arrays to Methods -- 7.7 Returning an Array from a Method -- 7.8 Case Study: Counting the Occurrences of Each Letter -- 7.9 Variable-Length Argument Lists -- 7.10 Searching Arrays -- 7.11 Sorting Arrays -- 7.12 The Arrays Class -- 7.13 Command-Line Arguments -- Chapter 8 Multidimensional Arrays -- 8.1 Introduction -- 8.2 Two-Dimensional Array Basics -- 8.3 Processing Two-Dimensional Arrays -- 8.4 Passing Two-Dimensional Arrays to Methods -- 8.5 Case Study: Grading a Multiple-Choice Test -- 8.6 Case Study: Finding the Closest Pair -- 8.7 Case Study: Sudoku -- 8.8 Multidimensional Arrays -- Chapter 9 Objects and Classes -- 9.1 Introduction -- 9.2 Defining Classes for Objects -- 9.3 Example: Defining Classes and Creating Objects -- 9.4 Constructing Objects Using Constructors -- 9.5 Accessing Objects via Reference Variables.
9.6 Using Classes from the Java Library -- 9.7 Static Variables, Constants, and Methods -- 9.8 Visibility Modifiers -- 9.9 Data Field Encapsulation -- 9.10 Passing Objects to Methods -- 9.11 Array of Objects -- 9.12 Immutable Objects and Classes -- 9.13 The Scope of Variables -- 9.14 The this Reference -- Chapter 10 Object-Oriented Thinking -- 10.1 Introduction -- 10.2 Class Abstraction and Encapsulation -- 10.3 Thinking in Objects -- 10.4 Class Relationships -- 10.5 Case Study: Designing the Course Class -- 10.6 Case Study: Designing a Class for Stacks -- 10.7 Processing Primitive Data Type Values as Objects -- 10.8 Automatic Conversion between Primitive Types and Wrapper Class Types -- 10.9 The BigInteger and BigDecimal Classes -- 10.10 The String Class -- 10.11 The StringBuilder and StringBuffer Classes -- Chapter 11 Inheritance and Polymorphism -- 11.1 Introduction -- 11.2 Superclasses and Subclasses -- 11.3 Using the super Keyword -- 11.4 Overriding Methods -- 11.5 Overriding vs. Overloading -- 11.6 The Object Class and Its toString() Method -- 11.7 Polymorphism -- 11.8 Dynamic Binding -- 11.9 Casting Objects and the instanceof Operator -- 11.10 The Object's equals Method -- 11.11 The ArrayList Class -- 11.12 Useful Methods for Lists -- 11.13 Case Study: A Custom Stack Class -- 11.14 The protected Data and Methods -- 11.15 Preventing Extending and Overriding -- Chapter 12 Exception Handling and Text I/O -- 12.1 Introduction -- 12.2 Exception-Handling Overview -- 12.3 Exception Types -- 12.4 More on Exception Handling -- 12.5 The finally Clause -- 12.6 When to Use Exceptions --

12.7 Rethrowing Exceptions -- 12.8 Chained Exceptions -- 12.9 Defining Custom Exception Classes -- 12.10 The File Class -- 12.11 File Input and Output -- 12.12 Reading Data from the Web -- 12.13 Case Study: Web Crawler -- Chapter 13 Abstract Classes and Interfaces. 13.1 Introduction -- 13.2 Abstract Classes -- 13.3 Case Study: the Abstract Number Class -- 13.4 Case Study: Calendar and GregorianCalendar -- 13.5 Interfaces -- 13.6 The Comparable Interface -- 13.7 The Cloneable Interface -- 13.8 Interfaces vs. Abstract Classes -- 13.9 Case Study: The Rational Class -- 13.10 Class Design Guidelines -- Chapter 14 JavaFX Basics -- 14.1 Introduction -- 14.2 JavaFX vs Swing and AWT -- 14.3 The Basic Structure of a JavaFX Program -- 14.4 Panes, UI Controls, and Shapes -- 14.5 Property Binding -- 14.6 Common Properties and Methods for Nodes -- 14.7 The Color Class -- 14.8 The Font Class -- 14.9 The Image and ImageView Classes -- 14.10 Layout Panes -- 14.11 Shapes -- 14.12 Case Study: The ClockPane Class -- Chapter 15 Event-Driven Programming and Animations -- 15.1 Introduction -- 15.2 Events and Event Sources -- 15.3 Registering Handlers and Handling Events -- 15.4 Inner Classes -- 15.5 Anonymous Inner Class Handlers -- 15.6 Simplifying Event Handling Using Lambda Expressions -- 15.7 Case Study: Loan Calculator -- 15.8 Mouse Events -- 15.9 Key Events -- 15.10 Listeners for Observable Objects -- 15.11 Animation -- 15.12 Case Study: Bouncing Ball -- Chapter 16 JavaFX UI Controls and Multimedia -- 16.1 Introduction -- 16.2 Labeled and Label -- 16.3 Button -- 16.4 CheckBox -- 16.5 RadioButton -- 16.6 TextField -- 16.7 TextArea -- 16.8 ComboBox -- 16.9 ListView -- 16.10 ScrollBar -- 16.11 Slider -- 16.12 Case Study: Developing a Tic-Tac-Toe Game -- 16.13 Video and Audio -- 16.14 Case Study: National Flags and Anthems -- Chapter 17 Binary I/O -- 17.1 Introduction -- 17.2 How Is Text I/O Handled in Java? -- 17.3 Text I/O vs. Binary I/O -- 17.4 Binary I/O Classes -- 17.5 Case Study: Copying Files -- 17.6 Object I/O -- 17.7 Random-Access Files -- Chapter 18 Recursion -- 18.1 Introduction -- 18.2 Case Study: Computing Factorials.
18.3 Case Study: Computing Fibonacci Numbers -- 18.4 Problem Solving Using Recursion -- 18.5 Recursive Helper Methods -- 18.6 Case Study: Finding the Directory Size -- 18.7 Case Study: Tower of Hanoi -- 18.8 Case Study: Fractals -- 18.9 Recursion vs. Iteration -- 18.10 Tail Recursion -- Appendix A Java Keywords -- Appendix B The ASCII Character Set -- Appendix C Operator Precedence Chart -- Appendix D Java Modifiers -- Appendix E Special Floating-Point Values -- Appendix F Number Systems -- Appendix G Bitwise Operations -- Appendix H Regular Expressions -- Appendix I Enumerated Types -- Index -- Symbols.

Sommario/riassunto

This text is intended for a 1-, 2-, or 3-semester CS1 course sequence. Daniel Liang teaches concepts of problem-solving and object-oriented programming using a fundamentals-first approach. Beginning programmers learn critical problem-solving techniques then move on to grasp the key concepts of object-oriented, GUI programming, advanced GUI and Web programming using Java. Liang approaches Java GUI programming using JavaFX, not only because JavaFX is much simpler for new Java programmers to learn and use but because it has replaced Swing as the new GUI tool for developing cross-platform-rich Internet applications on desktop computers, on hand-held devices, and on the Web. Additionally, for instructors, JavaFXprovides a better teaching tool for demonstrating object-oriented programming. Teaching and Learning Experience To provide a better teaching and learning experience, for both instructors and students, this program offers: Fundamentals-First Approach: Basic programming concepts

are introduced on control statements, loops, functions, and arrays before object-oriented programming is discussed.   Problem-Driven Motivation: The examples and exercises throughout the book emphasize problem solving and foster the concept of developing reusable components and using them to create practical projects.   A Superior Pedagogical Design that Fosters Student Interest: Key concepts are reinforced with objectives lists, introduction and chapter overviews, easy-to-follow examples, chapter summaries, review questions, programming exercises, and interactive self-tests.   &nbsp.