

1. Record Nr.	UNINA9910154799003321
Autore	Deitel Paul J.
Titolo	Java : how to program : early objects / / Paul Deitel, Harvey Deitel ; global edition contributions by Sherif G. Aly, Saleh Al-Hazbi
Pubbl/distr/stampa	Boston : , : Deitel : , : Prentice Hall, , [2015] ©2015
ISBN	1-292-07095-1
Edizione	[Tenth, Global edition.]
Descrizione fisica	1 online resource (1,250 pages) : illustrations
Collana	How to program series
Disciplina	005.133
Soggetti	Java (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Cover -- Title -- Copyright -- Foreword -- Preface -- Before You Begin -- 1 Introduction to Computers, the Internet and Java -- 1.1 Introduction -- 1.2 Hardware and Software -- 1.2.1 Moore's Law -- 1.2.2 Computer Organization -- 1.3 Data Hierarchy -- 1.4 Machine Languages, Assembly Languages and High-Level Languages -- 1.5 Introduction to Object Technology -- 1.5.1 The Automobile as an Object -- 1.5.2 Methods and Classes -- 1.5.3 Instantiation -- 1.5.4 Reuse -- 1.5.5 Messages and Method Calls -- 1.5.6 Attributes and Instance Variables -- 1.5.7 Encapsulation and Information Hiding -- 1.5.8 Inheritance -- 1.5.9 Interfaces -- 1.5.10 Object-Oriented Analysis and Design (OOAD) -- 1.5.11 The UML (Unified Modeling Language) -- 1.6 Operating Systems -- 1.6.1 Windows-A Proprietary Operating System -- 1.6.2 Linux-An Open-Source Operating System -- 1.6.3 Android -- 1.7 Programming Languages -- 1.8 Java -- 1.9 A Typical Java Development Environment -- 1.10 Test-Driving a Java Application -- 1.11 Internet and World Wide Web -- 1.11.1 The Internet: A Network of Networks -- 1.11.2 The World Wide Web: Making the Internet User-Friendly -- 1.11.3 Web Services and Mashups -- 1.11.4 Ajax -- 1.11.5 The Internet of Things -- 1.12 Software Technologies -- 1.13 Keeping Up-to-Date with Information Technologies -- 2 Introduction to Java Applications -- Input/Output and Operators -- 2.1 Introduction -- 2.2 Your First Program in Java: Printing a Line of Text -- 2.3 Modifying Your First Java Program -- 2.4

Displaying Text with printf -- 2.5 Another Application: Adding Integers -- 2.5.1 import Declarations -- 2.5.2 Declaring Class Addition -- 2.5.3 Declaring and Creating a Scanner to Obtain User Input from the Keyboard -- 2.5.4 Declaring Variables to Store Integers -- 2.5.5 Prompting the User for Input -- 2.5.6 Obtaining an int as Input from the User. 2.5.7 Prompting for and Inputting a Second int -- 2.5.8 Using Variables in a Calculation -- 2.5.9 Displaying the Result of the Calculation -- 2.5.10 Java API Documentation -- 2.6 Memory Concepts -- 2.7 Arithmetic -- 2.8 Decision Making: Equality and Relational Operators -- 2.9 Wrap-Up -- 3 Introduction to Classes, Objects, Methods and Strings -- 3.1 Introduction -- 3.2 Instance Variables, set Methods and get Methods -- 3.2.1 Account Class with an Instance Variable, a set Method and a get Method -- 3.2.2 AccountTest Class That Creates and Uses an Object of ClassAccount -- 3.2.3 Compiling and Executing an App with Multiple Classes -- 3.2.4 Account UML Class Diagram with an Instance Variable and set and get Methods -- 3.2.5 Additional Notes on Class AccountTest -- 3.2.6 Software Engineering with private Instance Variables and public set and get Methods -- 3.3 Primitive Types vs. Reference Types -- 3.4 Account Class: Initializing Objects with Constructors -- 3.4.1 Declaring an Account Constructor for Custom Object Initialization -- 3.4.2 Class AccountTest: Initializing Account Objects When They're Created -- 3.5 Account Class with a Balance -- Floating-Point Numbers -- 3.5.1 Account Class with a balance Instance Variable of Type double -- 3.5.2 AccountTest Class to Use Class Account -- 3.6 (Optional) GUI and Graphics Case Study: Using Dialog Boxes -- 3.7 Wrap-Up -- 4 Control Statements: Part 1 -- Assignment, ++ and -- Operators -- 4.1 Introduction -- 4.2 Algorithms -- 4.3 Pseudocode -- 4.4 Control Structures -- 4.5 if Single-Selection Statement -- 4.6 if...else Double-Selection Statement -- 4.7 Student Class: Nested if...else Statements -- 4.8 while Repetition Statement -- 4.9 Formulating Algorithms: Counter-Controlled Repetition -- 4.10 Formulating Algorithms: Sentinel-Controlled Repetition -- 4.11 Formulating Algorithms: Nested Control Statements. 4.12 Compound Assignment Operators -- 4.13 Increment and Decrement Operators -- 4.14 Primitive Types -- 4.15 (Optional) GUI and Graphics Case Study: Creating Simple Drawings -- 4.16 Wrap-Up -- 5 Control Statements: Part 2 -- Logical Operators -- 5.1 Introduction -- 5.2 Essentials of Counter-Controlled Repetition -- 5.3 for Repetition Statement -- 5.4 Examples Using the for Statement -- 5.5 do...while Repetition Statement -- 5.6 switch Multiple-Selection Statement -- 5.7 Class AutoPolicy Case Study: Strings in switch Statements -- 5.8 break and continue Statements -- 5.9 Logical Operators -- 5.10 Structured Programming Summary -- 5.11 (Optional) GUI and Graphics Case Study: Drawing Rectangles and Ovals -- 5.12 Wrap-Up -- 6 Methods: A Deeper Look -- 6.1 Introduction -- 6.2 Program Modules in Java -- 6.3 static Methods, static Fields and Class Math -- 6.4 Declaring Methods with Multiple Parameters -- 6.5 Notes on Declaring and Using Methods -- 6.6 Method-Call Stack and Stack Frames -- 6.7 Argument Promotion and Casting -- 6.8 Java API Packages -- 6.9 Case Study: Secure Random-Number Generation -- 6.10 Case Study: A Game of Chance -- Introducing enum Types -- 6.11 Scope of Declarations -- 6.12 Method Overloading -- 6.13 (Optional) GUI and Graphics Case Study: Colors and Filled Shapes -- 6.14 Wrap-Up -- 7 Arrays and ArrayLists -- 7.1 Introduction -- 7.2 Arrays -- 7.3 Declaring and Creating Arrays -- 7.4 Examples Using Arrays -- 7.4.1 Creating and Initializing an Array -- 7.4.2 Using an Array Initializer -- 7.4.3 Calculating the Values to Store in an Array --

7.4.4 Summing the Elements of an Array -- 7.4.5 Using Bar Charts to Display Array Data Graphically -- 7.4.6 Using the Elements of an Array as Counters -- 7.4.7 Using Arrays to Analyze Survey Results -- 7.5 Exception Handling: Processing the Incorrect Response -- 7.5.1 The try Statement.

7.5.2 Executing the catch Block -- 7.5.3 toString Method of the Exception Parameter -- 7.6 Case Study: Card Shuffling and Dealing Simulation -- 7.7 Enhanced for Statement -- 7.8 Passing Arrays to Methods -- 7.9 Pass-By-Value vs. Pass-By-Reference -- 7.10 Case Study: Class GradeBook Using an Array to Store Grades -- 7.11 Multidimensional Arrays -- 7.12 Case Study: Class GradeBook Using a Two-Dimensional Array -- 7.13 Variable-Length Argument Lists -- 7.14 Using Command-Line Arguments -- 7.15 Class Arrays -- 7.16 Introduction to Collections and Class ArrayList -- 7.17 (Optional) GUI and Graphics Case Study: Drawing Arcs -- 7.18 Wrap-Up -- 8 Classes and Objects: A Deeper Look -- 8.1 Introduction -- 8.2 Time Class Case Study -- 8.3 Controlling Access to Members -- 8.4 Referring to the Current Object's Members with the this Reference -- 8.5 Time Class Case Study: Overloaded Constructors -- 8.6 Default and No-Argument Constructors -- 8.7 Notes on Set and Get Methods -- 8.8 Composition -- 8.9 enum Types -- 8.10 Garbage Collection -- 8.11 static Class Members -- 8.12 static Import -- 8.13 final Instance Variables -- 8.14 Package Access -- 8.15 Using BigDecimal for Precise Monetary Calculations -- 8.16 (Optional) GUI and Graphics Case Study: Using Objects with Graphics -- 8.17 Wrap-Up -- 9 Object-Oriented Programming: Inheritance -- 9.1 Introduction -- 9.2 Superclasses and Subclasses -- 9.3 protected Members -- 9.4 Relationship Between Superclasses and Subclasses -- 9.4.1 Creating and Using a CommissionEmployee Class -- 9.4.2 Creating and Using a BasePlusCommissionEmployee Class -- 9.4.3 Creating a CommissionEmployee-BasePlusCommissionEmployee Inheritance Hierarchy -- 9.4.4 CommissionEmployee-BasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables.

9.4.5 CommissionEmployee-BasePlusCommissionEmployeeInheritance Hierarchy Using private Instance Variables -- 9.5 Constructors in Subclasses -- 9.6 Class Object -- 9.7 (Optional) GUI and Graphics Case Study: Displaying Text and Images Using Labels -- 9.8 Wrap-Up -- 10 Object-Oriented Programming: Polymorphism and Interfaces -- 10.1 Introduction -- 10.2 Polymorphism Examples -- 10.3 Demonstrating Polymorphic Behavior -- 10.4 Abstract Classes and Methods -- 10.5 Case Study: Payroll System Using Polymorphism -- 10.5.1 Abstract Superclass Employee -- 10.5.2 Concrete Subclass SalariedEmployee -- 10.5.3 Concrete Subclass HourlyEmployee -- 10.5.4 Concrete Subclass CommissionEmployee -- 10.5.5 Indirect Concrete Subclass BasePlusCommissionEmployee -- 10.5.6 Polymorphic Processing, Operator instanceof and Downcasting -- 10.6 Allowed Assignments Between Superclass and Subclass Variables -- 10.7 final Methods and Classes -- 10.8 A Deeper Explanation of Issues with Calling Methods from Constructors -- 10.9 Creating and Using Interfaces -- 10.9.1 Developing a Payable Hierarchy -- 10.9.2 Interface Payable -- 10.9.3 Class Invoice -- 10.9.4 Modifying Class Employee to Implement Interface Payable -- 10.9.5 Modifying Class SalariedEmployee for Use in the Payable Hierarchy -- 10.9.6 Using Interface Payable to Process Invoices and Employees Polymorphically -- 10.9.7 Some Common Interfaces of the Java API -- 10.10 Java SE 8 Interface Enhancements -- 10.10.1 default Interface Methods -- 10.10.2 static Interface Methods -- 10.10.3 Functional Interfaces -- 10.11 (Optional) GUI and Graphics

Case Study: Drawing with Polymorphism -- 10.12 Wrap-Up -- 11
Exception Handling: A Deeper Look -- 11.1 Introduction -- 11.2
Example: Divide by Zero without Exception Handling -- 11.3 Example:
Handling ArithmeticExceptions and InputMismatchExceptions -- 11.4
When to Use Exception Handling.
11.5 Java Exception Hierarchy.

Sommario/riassunto

Intended for use in the Java programming course The Deitels' groundbreaking How to Program series offers unparalleled breadth and depth of object-oriented programming concepts and intermediate-level topics for further study. Java How to Program (Early Objects), Tenth Edition, teaches programming by presenting the concepts in the context of full working programs and takes an early-objects approach Teaching and Learning Experience This program presents a better teaching and learning experience-for you and your students. Teach Programming with the Deitels' Signature Live Code Approach: Java language features are introduced with thousands of lines of code in hundreds of complete working programs. Introduce Objects Early: Presenting objects and classes early gets students "thinking about objects" immediately and mastering these concepts more thoroughly. Keep Your Course Current: This edition can be used with Java SE 7 or Java SE 8, and is up-to-date with the latest technologies and advancements. Facilitate Learning with Outstanding Applied Pedagogy: Making a Difference exercise sets, projects, and hundreds of valuable programming tips help students apply concepts. Support Instructors and Students: Student and instructor resources are available to expand on the topics presented in the text.
