Programming with Objects

A Comparative Presentation of Object-Oriented Programming with C++ and Java

Avinash C. Kak

Purdue University

A Wiley-Interscience Publication

JOHN WILEY & SONS, INC.
The errata and other information for this book is posted at

http://www.programming-with-objects.com
This book presents object-oriented programming with C++ and Java, which are today’s two dominant languages for such programming. The presentation format is mostly comparative, all the way from the basic language constructs to application-level issues dealing with graphics programming, network programming, and database programming. This book is intended for a reader who is well-conversant with the important features of C: pointers, strings, arrays, and structures.

The author strongly believes in the notion that, in addition to the syntax, it is essential to also show a programming language through its applications to fully establish its beauty and power. Teaching a programming language divorced from its applications – not uncommon in many educational programs – would be like teaching English through just its grammar.

This book grew out of an attempt to meet a specific academic need for a comprehensive educational program in object-oriented programming. We wanted a program that would not be too indoctrinating with regard to any one style (or any one language, since language often dictates style) of object-oriented programming. While programming skill could have been taught by focusing on a single language, education in its larger sense demanded that we provide a broader menu of styles and concepts. The result was what the reader sees in this book: An integrated presentation of C++ and Java. There is educational value in comparing and contrasting the two languages, from basic language constructs to how the languages are used in application-level programming. Such comparisons may even inspire an enterprising student to think of new and more powerful object-oriented languages of the future. To further enhance
the educational value of this comparative approach, this book also includes treatment of simulated object-orientation in plain C, with GNOME/GTK+ presented as a major example of this approach.

This book is based on the philosophy that learning by comparison is very efficient and can be a lot of fun. Sometimes we find it easier to remember and learn things if we can anchor our memory and comprehension in interesting differences and similarities between supposedly similar objects, structures, and situations. Learning C++ and Java together can exploit this aspect of human cognition. Students find it interesting to compare C++ and Java programming constructs for doing the same thing.

Teaching and learning C++ and Java together have some unique advantages. First, because both C++ and Java were born out of C, they have much in common at the level of basic language structures. Teaching these structures together saves time. For example, once the concept of a vector in C++ is made clear and some of the more useful functions associated with C++ vectors are elucidated, the discussion of the Java ArrayList takes hardly any time. The Java discussion consists mostly of pointing out the Java functions that do the same thing as the previously discussed C++ functions.

Then there is also the unique process of learning by coding up a program in C++ that does the same thing as a given program in Java, or vice versa. My experience is that this approach enables the students to tackle more difficult projects in both C++ and Java than would otherwise be the case under the time constraints of a course.

Learning two large languages together does have its down side. One can get confused as to what feature belongs to which language. Fortunately, this difficulty is minimized by the modern programming practice of keeping one eye on the on-line documentation in one terminal window while programming in another terminal window. Both Java and C++ have become so large that it would be impossible for anyone to commit to memory all of the classes and all of the functions and attributes defined for the classes. So even if one were not learning two languages simultaneously, one would still need to refer to documentation while writing programs.

The book contains more material than can be accommodated in a typical one-semester course. In my experience, the book works well for a sequence of two back-to-back courses, the first focusing on the basic language constructs as presented in the first fifteen chapters, and the second focusing on application- and design-level issues. For the second course, I complement the material in the last five chapters with a book on design patterns.

It would be naive of me to assume that a manuscript as large as this would be free of errors. I’d be much grateful to the readers who would bring the errors to my attention at kak@purdue.edu. All corrections will be made available online at www.programming-with-objects.com, and the authors of the corrections will be duly acknowledged. The same applies to any slip-ups on my part in giving proper attributions to authors. Where my example programs were inspired directly by what I saw in other sources, I have acknowledged their authors in the “Credits and Suggestions for Further Reading” section at the end of each chapter.
The author will be glad to make available to the prospective instructors the solutions to the homework problems.

Finally, the book should also be useful to those who are transitioning from C++ to Java, or vice versa.

Purdue University

Avinash Kak
Acknowledgments

Whatever merit this book has should go in large measure to the stalwarts of the object-oriented programming movement, to those who created C++ and Java, and to those who have been the chief expositors of these two languages over the last several years (see the references at the end of the book).

This book would not have been possible without the help of the following people:

- Guilherme DeSouza, a man with insights that are as deep as they are broad and for whom Linux is a religion to which he has converted many, this author included. Guilherme’s insights in multiprocessing and multithreading played an important role in the revamping of Chapter 18.

- Elvia Suryadi, who can spot from a mile the slightest flaw in a logical argument, the minutest weakness in an explanation. Her constant feedback helped with the cleanup of many sections of the book, too numerous to be listed here individually. The homework problems in Chapters 13 and 20 are by Elvia.

- Malcolm Slaney, with an uncanny eye for rigor and precision, for providing critical feedback on the first five chapters.

- Carl Crawford, never a man to mince words, who insisted that my earlier version of the front matter did not do justice to the rest of the book.

- Susan Gottschlich, with deep insights in the software development cycles of industry, for her careful reading and feedback of the first 11 chapters.
Robert Cromwell, who has always had his ear to the ground for the latest happenings in the world of software and hardware, for looking carefully at the C-related material in the book and suggesting improvements.

Sarah Sellke, with many years of object-oriented software design and development experience for telecom and other applications, for her feedback on Chapter 19.

Christina Pavlopoulou, as a source of great help in the early stages of my putting together this book.

Prathima Venkatesan, who proofread and caught many errors in the draft copies of Chapters 4, 5, 6, and 7.

Bob Mulvey, for helping me better understand the various shortcomings associated with the use of `setjmp–longjmp` macros for achieving multilevel return in C.

Brett Maden, for creating the final versions of the figures in Chapters 15 and 17. He also contributed to the homework section of Chapter 17; the Qt and GNOME/GTK+ problems were supplied by him.

Kheng Tan, for many of the final production figures for Chapters 14 and 16. Kheng also supplied the problems for the homework section of Chapter 14.

Thanks also go to the anonymous reviewers of the book; many of the comments I received through the review process helped in the revision of much material. Of the publisher reviews received nonanonymously, I wish to thank Simon Gray in particular for catching many errors in Chapter 2, 8, 10, 14, and 19 and suggesting improvements.

Many additional sources of help and information that proved important to the writing of this book are acknowledged near the end of each chapter in a section entitled “Credits and Suggestions for Further Reading,” or sometimes just “Suggestions for Further Reading.” Occasionally, I have also used a footnote for the same purpose.

I am also grateful to Subhash Kak, whose powers of exposition border on the lyrical, for his many wonderful suggestions for smoothing out the text at various places.

Finally, and most importantly, many thanks go to Carla for her loving friendship, support, and understanding, all things that give true meaning to life. Thanks also go to Carla for her direct contributions to the book by way of critical reading of its various sections.

A.C.K.
## Contents in Brief

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Why OO Programming – Some Parallels with Things at Large</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Baby Steps</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>The Notion of a Class and Some Other Key Ideas</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Strings</td>
<td>107</td>
</tr>
<tr>
<td>5</td>
<td>Using the Container Classes</td>
<td>147</td>
</tr>
<tr>
<td>6</td>
<td>The Primitive Types and Their Input/Output</td>
<td>211</td>
</tr>
<tr>
<td>7</td>
<td>Declarations, Definitions, and Initializations</td>
<td>287</td>
</tr>
<tr>
<td>8</td>
<td>Object Reference and Memory Allocation</td>
<td>339</td>
</tr>
<tr>
<td>9</td>
<td>Functions and Methods</td>
<td>353</td>
</tr>
</tbody>
</table>
10 Handling Exceptions 389
11 Classes, The Rest of the Story 423
12 Overloading Operators in C++ 513
13 Generics and Templates 569
14 Modeling Diagrams for OO Programs 607
15 Extending Classes 637
16 Multiple Inheritance in C++ 743
17 OO for Graphical User Interfaces, A Tour of Three Toolkits 811
18 Multithreaded Object-Oriented Programming 985
19 Network Programming 1049
20 Database Programming 1137
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td></td>
<td>xi</td>
</tr>
<tr>
<td>1</td>
<td>Why OO Programming – Some Parallels with Things at Large</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Baby Steps</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Simple Programs: Summing an Array of Integers</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Simple Programs: Terminal I/O</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Simple Programs: File I/O</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>Suggestions for Further Reading</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Homework</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>The Notion of a Class and Some Other Key Ideas</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Defining a Class in C++</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Defining a Class in Java</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>Constructing Objects: Differences and Similarities between C++ and Java</td>
<td>40</td>
</tr>
<tr>
<td>3.4</td>
<td>Defining a subclass in C++</td>
<td>42</td>
</tr>
</tbody>
</table>

xv
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.1</td>
<td>A Small Demonstration of Polymorphism in C++</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Defining a Subclass in Java</td>
<td>46</td>
</tr>
<tr>
<td>3.5.1</td>
<td>A Small Demonstration of Polymorphism in Java</td>
<td>48</td>
</tr>
<tr>
<td>3.6</td>
<td>Blocking Inheritance</td>
<td>49</td>
</tr>
<tr>
<td>3.7</td>
<td>Creating Print Representations for Objects</td>
<td>52</td>
</tr>
<tr>
<td>3.8</td>
<td>Object Destruction</td>
<td>54</td>
</tr>
<tr>
<td>3.9</td>
<td>Packages in Java</td>
<td>55</td>
</tr>
<tr>
<td>3.10</td>
<td>Namespaces in C++</td>
<td>61</td>
</tr>
<tr>
<td>3.10.1</td>
<td>Using Declaration Versus Using Directive</td>
<td>64</td>
</tr>
<tr>
<td>3.10.2</td>
<td>Which Namespace Owns Names Imported from Another Namespace?</td>
<td>65</td>
</tr>
<tr>
<td>3.10.3</td>
<td>Using Declarations and Directives Have Scope</td>
<td>66</td>
</tr>
<tr>
<td>3.10.4</td>
<td>Nesting Namespaces and Namespace Aliases</td>
<td>67</td>
</tr>
<tr>
<td>3.10.5</td>
<td>Unnamed Namespaces</td>
<td>69</td>
</tr>
<tr>
<td>3.10.6</td>
<td>Koenig Lookup for Unqualified Function Names</td>
<td>70</td>
</tr>
<tr>
<td>3.11</td>
<td>Access Control for Class Members</td>
<td>71</td>
</tr>
<tr>
<td>3.12</td>
<td>Abstract Classes and Interfaces</td>
<td>73</td>
</tr>
<tr>
<td>3.13</td>
<td>Comparing Objects</td>
<td>77</td>
</tr>
<tr>
<td>3.14</td>
<td>Static Members of a Class</td>
<td>81</td>
</tr>
<tr>
<td>3.15</td>
<td>Template Classes</td>
<td>83</td>
</tr>
<tr>
<td>3.16</td>
<td>Nested Types</td>
<td>84</td>
</tr>
<tr>
<td>3.16.1</td>
<td>Nested Classes in C++</td>
<td>84</td>
</tr>
<tr>
<td>3.16.2</td>
<td>Nested Classes in Java</td>
<td>89</td>
</tr>
<tr>
<td>3.17</td>
<td>Implementing OO Behavior in C Programs</td>
<td>94</td>
</tr>
<tr>
<td>3.18</td>
<td>Suggestions for Further Reading</td>
<td>102</td>
</tr>
<tr>
<td>3.19</td>
<td>Homework</td>
<td>102</td>
</tr>
</tbody>
</table>

4 Strings                                                                 107

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Strings in C, A Brief Review</td>
<td>108</td>
</tr>
<tr>
<td>4.2</td>
<td>Some Common Shortcomings of C-Style Strings</td>
<td>111</td>
</tr>
<tr>
<td>4.3</td>
<td>C++ Strings</td>
<td>113</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Constructing a C++ String Object</td>
<td>113</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Accessing Individual Characters</td>
<td>114</td>
</tr>
<tr>
<td>4.3.3</td>
<td>String Comparison</td>
<td>115</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Joining Strings Together</td>
<td>118</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Searching for Substrings and Characters</td>
<td>120</td>
</tr>
</tbody>
</table>
CONTENTS

6 The Primitive Types and Their Input/Output 211
  6.1 Tokens, Identifiers, and Variable Names 211
  6.2 Primitive Types in C++ and Java 213
  6.3 Boolean Type 213
  6.4 Character Types 214
  6.5 Integer Types 220
  6.6 Floating-point Types 222
  6.7 Type Conversion for the Primitive Types 223
    6.7.1 Implicit Type Conversions in C++ 224
    6.7.2 Implicit Type Conversions in Java 228
    6.7.3 Explicit Type Conversion in C++ 232
    6.7.4 Explicit Type Conversion in Java 234
  6.8 I/O Streams for C++ 236
    6.8.1 The C++ Stream Hierarchy 237
    6.8.2 Input–Output Operations for Character Streams 238
    6.8.3 Input–Output Operations for Byte Streams 246
    6.8.4 Controlling the Format 252
    6.8.5 String Streams 257
  6.9 I/O Streams for Java 258
    6.9.1 Writing Primitive Types 259
    6.9.2 Writing Strings 267
    6.9.3 Reading the Primitive Types 271
    6.9.4 Reading Strings 273
  6.10 Suggestions for Further Reading 274
  6.11 Homework 275

7 Declarations, Definitions, and Initializations 287
  7.1 When is a Declaration also a Definition? 287
  7.2 Are the Defined Variables in C++ Initialized by Default? 290
  7.3 Are the Defined Variables in Java Initialized by Default? 298
  7.4 Declaration of Pointer Types in C++ 302
  7.5 Arrays of Pointers in C++ 306
  7.6 Declaring Multiple Names 308
  7.7 Scope of an Identifier in C++ 309
  7.8 Scope of an Identifier in Java 311
  7.9 Arrays and Their Initialization in C++ 312
CONTENTS

7.10 Arrays and their Initialization in Java 318
  7.10.1 A Java Array is an Object 321
  7.10.2 java.lang.Arrays Class for Sorting, Searching, etc. 322
7.11 Symbolic Constants 326
7.12 Macros in C++ 328
7.13 The Enumeration Type in C++ 329
7.14 Suggestions for Further Reading 334
7.15 Homework 334

8 Object Reference and Memory Allocation 339
  8.1 Object Reference in C++ 339
  8.2 Object Reference in Java 342
  8.3 Memory Allocation in C++ 343
  8.4 Memory Allocation in Java 345
  8.5 Structures in C++ 346
  8.6 Homework 349

9 Functions and Methods 353
  9.1 Function Declarations 354
  9.2 Passing Arguments in C++ 354
    9.2.1 Passing a Primitive Type Argument by Value 355
    9.2.2 Passing a Primitive Type Argument by Pointer 355
    9.2.3 Passing a Primitive Type Argument by Reference 357
    9.2.4 Passing a Class Type Argument by Value 358
    9.2.5 Passing a Class Type Argument by Pointer 359
    9.2.6 Passing a Class Type Argument by Reference 361
  9.3 Passing Arguments in Java 362
    9.3.1 Passing a Primitive Type Argument 362
    9.3.2 Passing a Class Type Argument 362
  9.4 C++ Functions Returning Reference Types 367
  9.5 Function Inlining in C++ 368
  9.6 Static Variables in C++ 370
  9.7 const Parameter and Return Type for C++ Functions 371
  9.8 final Parameters for Java Methods 372
  9.9 Array Arguments 373
  9.10 Overloading of Function Names and Constructors in C++ 373
CONTENTS  

11.15 A C++ Study of Interleaved Classes of Moderate Complexity 483
11.16 A Java Study of Interleaved Classes of Moderate Complexity 496
11.17 Suggestions for Further Reading 504
11.18 Homework 504

12 Overloading Operators in C++ 513
12.1 Operator Tokens and Operator Functions 514
12.2 Global Overload Definitions for Operators 515
12.3 Member-Function Overload Definitions for Operators 517
12.4 Global Overload Definitions for Unary Operators 520
12.5 Member-Function Overload Definitions for Unary Operators 521
12.6 A Case Study in Operator Overloading 522
12.7 Smart Pointers: Overloading of Dereferencing Operators 536
12.8 Overloading Increment and Decrement Operators 546
12.9 User-Defined Conversions 551
12.10 Overloading of the ‘()’ Operator 555
12.11 Sorting Class Type Objects by Overloading the < Operator 558
12.12 Suggestions for Further Reading 561
12.13 Homework 561

13 Generics and Templates 569
13.1 Templatized Classes and Functions in C++ 572
13.1.1 A C++ Implementation of a Linked-List Program 572
13.1.2 A Parameterized Linked-List Program 576
13.1.3 Function Templates in C++ 577
13.1.4 Template Specialization 580
13.1.5 General Syntax of a Template Declaration 583
13.2 Iterators Revisited 585
13.2.1 Iterator Categories for Generic Algorithms 585
13.2.2 How to Declare an Iterator 587
13.3 Parameterized Classes in Java 588
13.3.1 Creating Your Own Parameterized Types in Java 591
13.3.2 Parameterization of Methods 596
13.3.3 Constraining the Parameters 599
13.4 Suggestions for Further Reading 602
13.5 Homework 603

14 Modeling Diagrams for OO Programs 607
14.1 Use Case Diagram 608
14.2 Class Diagram 610
14.2.1 Association as a Relationship Between Classes 612
14.2.2 Aggregation and Composition as Relationships Between Classes 613
14.2.3 Representing Attributes 615
14.2.4 Representing Operations 615
14.2.5 Stereotypes 616
14.3 Interaction Diagram 617
14.3.1 Sequence Diagram 617
14.3.2 Collaboration Diagram 623
14.4 Package Diagram 624
14.5 Statechart Diagram 626
14.6 Activity Diagram 631
14.7 Credits and Suggestions for Further Reading 633
14.8 Homework 635

15 Extending Classes 637
15.1 Public Derivation of a Subclass in C++ 637
15.2 Constructors for Derived Classes in C++ 641
15.3 Copy Constructors for Derived Classes in C++ 643
15.4 Assignment Operators for Derived Classes in C++ 646
15.5 Overloading Operators for Derived Classes in C++ 648
15.6 Destructors for Derived Classes in C++ 653
15.7 Virtual Member Functions in C++ 659
15.7.1 Restrictions on Virtual Function Declarations 664
15.7.2 Virtual Functions in Multilevel Hierarchies 664
15.7.3 Can Operators Be Made to Behave Polymorphically? 667
15.7.4 Polymorphic Types 667
15.8 Static versus Dynamic Binding for Functions in C++ 668
15.9 Restrictions on Overriding Functions in C++ 672
15.10 Virtual Destructors in C++ 676
15.11 Constructor Order Dependencies in C++ 678
15.12 Abstract Classes in C++ 681
15.13 Protected and Private Derived Classes in C++ 686
15.14 Extending Classes in Java 691
15.15 Restrictions on Overriding Methods in Java 695
15.16 Constructor Order Dependencies in Java 698
15.17 Abstract Classes in Java 699
15.18 Interfaces in Java 702
  15.18.1 Implementing Multiple Interfaces in Java 707
  15.18.2 Extending Interfaces in Java 708
  15.18.3 Constants in Interfaces 711
15.19 A C++ Study of a Small Class Hierarchy with Moderately Complex Behavior 712
15.20 A Java Study of a Small Class Hierarchy exhibiting Moderately Complex Behavior 727
15.21 Credits and Suggestions for Further Reading 736
15.22 Homework 737

16 Multiple Inheritance in C++ 743
  16.1 Some Examples for MI 744
  16.2 Issues that Arise with Repeated Inheritance 751
  16.3 Virtual Bases for Multiple Inheritance 753
  16.4 Virtual Bases and Copy Constructors 759
  16.5 Virtual Bases and Assignment Operators 762
  16.6 Avoiding Name Conflicts for Member Functions 769
  16.7 Dealing with Name Conflicts for Data Members 771
  16.8 Implementation of an Example in Repeated Inheritance 773
  16.9 Using Mixin Classes 782
  16.10 Using Role-Playing Classes 790
  16.11 Run-Time Type Identification in C++ 802
  16.12 Credits and Suggestions for Further Reading 804
  16.13 Homework 804

17 OO for Graphical User Interfaces, A Tour of Three Toolkits 811
  17.1 A Brief History of GUI Toolkits 812
  17.2 AWT/Swing Components 815
  17.3 Qt Widgets 817
  17.4 GNOME/GTK+ Widgets 817
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.5 Minimalist GUI Programs in AWT/Swing</td>
<td>818</td>
</tr>
<tr>
<td>17.6 Minimalist GUI Programs in Qt</td>
<td>823</td>
</tr>
<tr>
<td>17.7 Minimalist Programs in GNOME/GTK+</td>
<td>826</td>
</tr>
<tr>
<td>17.8 Layout Management in GUI Programs</td>
<td>830</td>
</tr>
<tr>
<td>17.9 Layout Management in AWT/Swing</td>
<td>831</td>
</tr>
<tr>
<td>17.9.1 Border Layout</td>
<td>832</td>
</tr>
<tr>
<td>17.9.2 Flow Layout</td>
<td>834</td>
</tr>
<tr>
<td>17.9.3 Box Layout</td>
<td>837</td>
</tr>
<tr>
<td>17.9.4 Grid Layout</td>
<td>841</td>
</tr>
<tr>
<td>17.9.5 Card Layout</td>
<td>843</td>
</tr>
<tr>
<td>17.9.6 Grid-Bag Layout</td>
<td>847</td>
</tr>
<tr>
<td>17.10 Layout Management in Qt</td>
<td>852</td>
</tr>
<tr>
<td>17.10.1 Box Layout</td>
<td>853</td>
</tr>
<tr>
<td>17.10.2 Grid Layout</td>
<td>856</td>
</tr>
<tr>
<td>17.11 Layout Management in GNOME/GTK+</td>
<td>859</td>
</tr>
<tr>
<td>17.11.1 Box Layout</td>
<td>859</td>
</tr>
<tr>
<td>17.11.2 Table Layout</td>
<td>861</td>
</tr>
<tr>
<td>17.12 Event Processing in GUI Programs</td>
<td>864</td>
</tr>
<tr>
<td>17.13 Event Processing in AWT/Swing</td>
<td>867</td>
</tr>
<tr>
<td>17.13.1 An Example in Inter-Component Communication in AWT/Swing</td>
<td>875</td>
</tr>
<tr>
<td>17.14 Event Processing in Qt</td>
<td>880</td>
</tr>
<tr>
<td>17.14.1 A Qt Example that requires Meta Object Compilation</td>
<td>883</td>
</tr>
<tr>
<td>17.14.2 Summary of Facts about Signals and Slots</td>
<td>892</td>
</tr>
<tr>
<td>17.15 Event Processing in GNOME/GTK+</td>
<td>893</td>
</tr>
<tr>
<td>17.15.1 Communicating Events to Other Widgets in GNOME/GTK+</td>
<td>894</td>
</tr>
<tr>
<td>17.15.2 Summary of Facts about Callbacks in GNOME/Gtk+</td>
<td>901</td>
</tr>
<tr>
<td>17.16 Windows with Menus in AWT/Swing</td>
<td>903</td>
</tr>
<tr>
<td>17.17 Windows with Menus in Qt</td>
<td>908</td>
</tr>
<tr>
<td>17.18 Windows with Menus in GNOME/GTK+</td>
<td>916</td>
</tr>
<tr>
<td>17.19 Drawing Shapes, Text, and Images in AWT/Swing</td>
<td>925</td>
</tr>
<tr>
<td>17.20 Drawing Shapes, Text, and Images in Qt</td>
<td>940</td>
</tr>
<tr>
<td>17.21 Drawing Shapes, Text, and Images in GNOME/GTK+</td>
<td>948</td>
</tr>
<tr>
<td>17.22 Java Applets</td>
<td>958</td>
</tr>
<tr>
<td>17.22.1 Life Cycle of an Applet</td>
<td>959</td>
</tr>
<tr>
<td>17.22.2 The Applet Tag</td>
<td>960</td>
</tr>
</tbody>
</table>
17.22.3 An Applet Example 962
17.22.4 Dual-Purpose Programming for Applets 970
17.22.5 The AppletContext Interface 973
17.22.6 Security Issues Related to Applets 978
17.23 Credits and Suggestions for Further Reading 980
17.24 Homework 980

18 Multithreaded Object-Oriented Programming 985
18.1 Creating and Executing Simple Threads in Java 986
18.2 The Runnable Interface in Java 990
18.3 Thread States 992
18.4 Thread Interference in Java 993
18.5 Thread Synchronization in Java 1001
18.6 Java’s `wait–notify` Mechanism for Dealing with Deadlock 1005
18.7 Data I/O Between Threads in Java 1010
18.8 Java Threads for Applets 1012
18.9 The Event Dispatch Thread in AWT/Swing 1015
18.10 Multithreaded Programming in C/C++ 1024
   18.10.1 Demonstrating Thread Interference with POSIX Threads 1030
   18.10.2 MUTEX for Dealing with POSIX Thread Interference 1032
   18.10.3 POSIX Threads: Condition Variables and the `wait–signal` Mechanism for Dealing with Deadlock 1035
18.11 Object-Oriented Multithreading in C++ 1040
18.12 Credits and Suggestions for Further Reading 1046
18.13 Homework 1047

19 Network Programming 1049
19.1 Establishing Socket Connections with Existing Servers in Java 1050
19.2 Server Sockets in Java 1053
19.3 Establishing Socket Connections with Existing Servers in C++ 1059
19.4 Server Sockets in C++ (Qt) 1066
19.5 Suggestions for Further Reading 1075
19.6 Homework 1075
# CONTENTS

## 20 Database Programming

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.1 Relational Databases</td>
<td>1138</td>
</tr>
<tr>
<td>20.2 The MySQL Database Manager</td>
<td>1140</td>
</tr>
<tr>
<td>20.3 SQL</td>
<td>1141</td>
</tr>
<tr>
<td>20.4 JDBC Programming: Invoking SQL through Java</td>
<td>1152</td>
</tr>
<tr>
<td>20.5 Mysql++ Programming: Invoking SQL through C++</td>
<td>1157</td>
</tr>
<tr>
<td>20.6 Suggestions for Further Reading</td>
<td>1163</td>
</tr>
<tr>
<td>20.7 Homework</td>
<td>1163</td>
</tr>
</tbody>
</table>

## Index

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1165</td>
</tr>
</tbody>
</table>
Index

* as iteration marker (UML), 599
  for iterator dereferencing (C++), 154
  for multiple triggers (UML), 613
  for pointer dereferencing, 519
  for representing multiplicity (UML), 593
to show all columns (SQL), 1062
&
  address operator (C++), 297
  for iterator initialization (C++), 157
— , member access operator (C++), 519
— >*, (C++), 463
//, 9, 14
:: (C++), 61, 619
[] (C++), 114, 150, 510
.* (C++), 463
#, for protected visibility (UML), 595
+, for private visibility (UML), 595
+, for public visibility (UML), 595
1 =, 152, 513
— =, 154
+++, 154
<< (C++), 9, 628
>>, (C++), 16
==
(C++), 78, 115, 152, 171, 513
Java), 79, 135
(C++), 79, 171, 176, 178, 182
<=(C++), 152
> (C++), 152, 513
>= (C++), 152
<<<> (UML), 596
A
abort() (C++), 23, 393
absolute positioning for layout, 810
abstract
  a keyword, 677
class, 73, 75, 617, 660, 677, 761
function, 662
method, 662
AbstractCollection (Java), 75–76
AbstractList (Java), 75–76
AbstractShape.cc, 661
AbstractShapeIncremental.cc, 662
AbstractShapeIncremental.java, 678
Abstract Window Toolkit (AWT/Swing), 793
accelerator, keyboard, 889, 895
accept() (Java), 1032, 1045
Access, 1059
access control, 3, 30, 71
  package, 71
private, 34, 38, 71
protected, 71
public, 34, 38, 71
acl.read property (Java), 957
acl.write property (Java), 957
acquiring lock
on a class, 979
on an object, 979
ActionEvent (AWT/Swing), 882
ActionListener (AWT/Swing), 845, 847, 850,
882–883, 941, 954
actionPerformed() (AWT/Swing), 850, 882, 884,
941
action (UML), 606
activate() (Qt), 833, 836
activation (UML), 602
icon, 598
activity
diagram (UML), 588, 611
state (UML), 606
actor (UML), 588
adapter
class (AWT/Swing), 680, 845
container (C++), 151, 173, 175
ADD (SQL), 1069
add() (Java), 182, 186, 190
addActionListener() (AWT/Swing), 799, 848, 850
addAll() (Java), 186
AddArray1.c, 6
AddArray2.c, 7
AddArray.cc, 7
AddArray.java, 10
addComponentListener() (AWT/Swing), 852
addElement() (Java), 194, 196
addFocusListener() (AWT/Swing), 852
addImage() (AWT/Swing), 915
addItemListener() (AWT/Swing), 852
ddKeyListener() (AWT/Swing), 852
addMouseListener() (AWT/Swing), 852
addMultiCellWidget() (Qt), 836
addWidget() (Qt), 833, 835–836
addWindowListener() (AWT/Swing), 799, 848,
850
adjacent find() (C++), 567
affine transformation, 919, 928
aggregation (UML), 591, 594
algorithm, a C++ header file, 8
ALIGN, 937–938
alignment (Qt), 833
ALT, 938
ALTER (SQL), 1067, 1069
ALTER TABLE (SQL), 1062
American National Standards Institute, 8
amortized constant-time complexity, 151
anchor
(C), 927
(Java), 827
animated applets, 964
animation, 990
Animator.java, 991
anonymous class (Java), 571, 850
ANSI, 8
API, Application Programming Interface, 791
app (C++), 238–239
append()
(C++), 119
(Java), 133, 138
append file mode
(C++), 239
(Java), 267
APPLET tag (HTML), 937–938, 947
Applet (Java), 936
applet (Java)
animated, 964
context, 951
security, 936
appletviewer tool (Java), 956, 937, 947
application, 13
programming interface, 791
approximate numerics (SQL), 1060
argc, 20, 804
argument-dependent name lookup (C++), 70
argument object, 503
argv, 20, 804
ArithmeticException (Java), 409
arithmetic types (C++), 223
arity of an operator, 498, 504
array
(C), 6
(Java), 14
declaration
(C++), 306
(Java), 311
initialization
(C++), 301, 308
(Java), 311
of pointers (C++), 300
parameter (C), 7
ArrayBasic.java, 313
array-like random access, 150
ArrayList (Java), 75, 185–186, 194, 569, 205,
1034, 569
Arrays (Java), 55, 315
ArraysFill.java, 316
ArraySizeOf.cc, 128
ArraysShuffle.java, 317
ASCII, 109, 212, 214–215, 218, 247
asList() (Java), 201, 315
assert() (C++), 121
assign() (C++), 129
assignment operator
(C++), 452
for derived class, 626
(Java), 455
AssignTest.java, 456
association (UML), 591, 769
bidirectional, 593
INDEX 1167

<table>
<thead>
<tr>
<th>Binary, 593</th>
<th>I/O (C++), 247</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicity, 592</td>
<td>(Java), 259</td>
</tr>
<tr>
<td>Reflexive, 593</td>
<td>Mode, 246</td>
</tr>
<tr>
<td>Rolename, 592</td>
<td>Numeric promotion, 215, 227, 232</td>
</tr>
<tr>
<td>Associative container (C++), 151, 565</td>
<td>Operator, 499</td>
</tr>
<tr>
<td>Asynchronous</td>
<td>Stream (C++), 246</td>
</tr>
<tr>
<td>Image loading, 915</td>
<td>Association (UML), 593</td>
</tr>
<tr>
<td>Method invocation, 601</td>
<td>BinaryFileIO.cc, 247</td>
</tr>
<tr>
<td><code>at()</code> (C++), 114</td>
<td>BinaryFileIO2.cc, 248</td>
</tr>
<tr>
<td><code>ate</code> (C++), 239, 248</td>
<td>BinaryFileIO3.cc, 251</td>
</tr>
<tr>
<td>Atomic components</td>
<td>BinarySearch() (Java), 316</td>
</tr>
<tr>
<td>(AWT/Swing), 794</td>
<td>BIT (SQL), 1061</td>
</tr>
<tr>
<td>(GNOME/GTK+), 796</td>
<td>BIT (n) (SQL), 1061</td>
</tr>
<tr>
<td>(Qt), 795</td>
<td>Bitset</td>
</tr>
<tr>
<td>Attribute(s)</td>
<td>A C++ container class, 150, 152</td>
</tr>
<tr>
<td>(UML), 590</td>
<td>A C++ header file, 8</td>
</tr>
<tr>
<td>Of POSIX threads, 1003</td>
<td>Bit strings (SQL), 1060</td>
</tr>
<tr>
<td>representation (UML), 595</td>
<td>BIT VARYING (n) (SQL), 1061</td>
</tr>
<tr>
<td>Visibility (UML), 595</td>
<td>Block, 304</td>
</tr>
<tr>
<td>Automatic type conversion</td>
<td>Nested (C++), 304</td>
</tr>
<tr>
<td>(C++), 224</td>
<td>Scope (C++), 66, 303–304</td>
</tr>
<tr>
<td>(Java), 228</td>
<td>(Java), 305</td>
</tr>
<tr>
<td>For assignment, 223</td>
<td>Blocked state, 970</td>
</tr>
<tr>
<td>For conversion of operands to common type, 223</td>
<td>Blocking</td>
</tr>
<tr>
<td>For conversion of returned value, 223</td>
<td>Inheritance (Java), 49, 680</td>
</tr>
<tr>
<td>For initialization, 223</td>
<td>I/O, 16</td>
</tr>
<tr>
<td>For matching argument with parameter, 223</td>
<td>BlockInheritance.cc, 52</td>
</tr>
<tr>
<td>For primitive types, 223</td>
<td>BlockInheritance.java, 49</td>
</tr>
<tr>
<td>Auto_ptr (C++), 529</td>
<td>BlockInheritance2.java, 50</td>
</tr>
<tr>
<td>AWT (Java), 794, 790–792</td>
<td>Bool (C++), 213</td>
</tr>
<tr>
<td>Thread, 993</td>
<td>Boolean (Java), 19, 211, 213</td>
</tr>
<tr>
<td>B</td>
<td>Bootclasspath (Java), 571</td>
</tr>
<tr>
<td>Back() (C++), 157, 174–175</td>
<td>Border (AWT/Swing), 810, 813</td>
</tr>
<tr>
<td>Bad() (C++), 243</td>
<td>BorderFactory (AWT/Swing), 810, 813, 817, 822</td>
</tr>
<tr>
<td>Base</td>
<td>BorderLayout (AWT/Swing), 799–800, 809–810, 817</td>
</tr>
<tr>
<td>Class, 31</td>
<td>BorderLayoutTest.java, 810</td>
</tr>
<tr>
<td>(C++), 42</td>
<td>Born state, 967, 970</td>
</tr>
<tr>
<td>(Java), 47, 669</td>
<td>BOTH (AWT/Swing), 826</td>
</tr>
<tr>
<td>Private (C++), 618</td>
<td>Bottom (AWT/Swing), 813, 822</td>
</tr>
<tr>
<td>Protected (C++), 618</td>
<td>BottomAttach (GNOME/GTK+), 840</td>
</tr>
<tr>
<td>Public (C++), 618</td>
<td>BoxLayout (AWT/Swing), 809, 815</td>
</tr>
<tr>
<td>Slice, 622</td>
<td>BoxLayoutTest.java, 817</td>
</tr>
<tr>
<td>Indirect, 732</td>
<td>Bridge driver, 1055</td>
</tr>
<tr>
<td>Virtual (C++), 731</td>
<td>Brighten() (AWT/Swing), 905</td>
</tr>
<tr>
<td>Baseline for drawing a string, 908</td>
<td>Browser, 936–938</td>
</tr>
<tr>
<td>Basicifstream&lt;char&gt; (C++), 237</td>
<td>Java enabled, 939</td>
</tr>
<tr>
<td>Basicifstream&lt;wchar_t&gt; (C++), 237</td>
<td>JVM, 937, 940, 942–943</td>
</tr>
<tr>
<td>Basicistream&lt;char&gt; (C++), 237</td>
<td>BufferedOutputStream (Java), 264</td>
</tr>
<tr>
<td>Basicistream&lt;wchar_t&gt; (C++), 237</td>
<td>BufferedReader (Java), 1029, 205</td>
</tr>
<tr>
<td>Basicostream (C++), 8</td>
<td>Buffered TCP socket, 1038</td>
</tr>
</tbody>
</table>
buffer flushing, 258
buffering of a stream, 258
byte (Java), 221
bytecode (Java), 11
byteWritten() (Qt), 1038

C
callback, 807, 844, 871, 873, 879, 932
capacity, 197
(C++), 124
(Java), 132
capacity() (Java), 132
CardLayout (AWT/Swing), 809, 821
CardLayoutTest.java, 822
CASE_INSENSITIVE_ORDER, 137
case label, 305
cast, 117, 223, 232, 234
casting away const (C++), 781
catch clause
(C++), 389
(Java), 400
C++ compilation, 5
with CC, 10
with g++, 9
CDE, 791
CENTER (AWT/Swing), 810, 813, 819, 822, 827
cerr (C++), 21, 23, 239
class, 2, 29
abstract, 617
anonymous (Java), 850
base, 31
(C++, 42
(Java), 47
behavior of, 29
concrete, 76
definition
(C++), 32
(Java), 37
derivation
private (C++), 664
protected (C++), 664
public (C++), 664
(Java), 669
derived, 31
(C++), 42, 617
(Java), 46, 669
diagram (UML), 588, 590
conceptual, 592
implementation, 592
specifier, 592

bitset, 8
collection, 8
deque, 8
dependence, 8
exception, 8
iostream, 8
functional, 8
iomanip, 8
ios, 8
iostream, 8
istream, 8
iterator, 8
limits, 8
locale, 8
map, 8
memory, 8
new, 8
numeral, 8
ostream, 8
queue, 8
set, 8
stringstream, 8
stack, 8
std except, 8
streambuf, 8
string, 8
typeinfo, 8
utility, 8
valarray, 8
vector, 8
checked exception (Java), 25, 399, 408
cin (C++), 16–17, 239
class, 2, 29
abstract, 617
anonymous (Java), 850
base, 31
(C++, 42
(Java), 47
behavior of, 29
concrete, 76
definition
(C++), 32
(Java), 37
derivation
private (C++), 664
protected (C++), 664
public (C++), 664
(Java), 669
derived, 31
(C++), 42, 617
(Java), 46, 669
diagram (UML), 588, 590
conceptual, 592
implementation, 592
specifier, 592

char* (C), 108
CHAR (n) (SQL), 1060–1061
CHAR (SQL), 1061
CHARACTER
(SQL), 1061
VARYING(n) (SQL), 1061
CHARACTER(n) (SQL), 1061
Character (Java), 194
cache, 211
cost, 215
cast, 215
class, 215
constant, 215
elementary, 215
strings (SQL), 1060
charAt() (Java), 134
CharEscapes.cc, 216
CharEscapes.java, 219
ChatServer
(C++), 1045
(Java), 1031
ChatServer.cc, 1048
ChatServer.h (Qt), 1047
ChatServer.java, 1035
C header files used in C++, 386
C++ header files used in C++, 386
algorithm, 8

bitset, 8
collection, 8
deque, 8
dependence, 8
exception, 8
iostream, 8
functional, 8
iomanip, 8
ios, 8
iostream, 8
istream, 8
iterator, 8
limits, 8
locale, 8
map, 8
memory, 8
new, 8
numeral, 8
ostream, 8
queue, 8
set, 8
stringstream, 8
stack, 8
std except, 8
streambuf, 8
string, 8
typeinfo, 8
utility, 8
valarray, 8
vector, 8
checked exception (Java), 25, 399, 408
cin (C++), 16–17, 239
class, 2, 29
abstract, 617
anonymous (Java), 850
base, 31
(C++, 42
(Java), 47
behavior of, 29
concrete, 76
definition
(C++), 32
(Java), 37
derivation
private (C++), 664
protected (C++), 664
public (C++), 664
(Java), 669
derived, 31
(C++), 42, 617
(Java), 46, 669
diagram (UML), 588, 590
conceptual, 592
implementation, 592
specifier, 592

char* (C), 108
CHAR (n) (SQL), 1060–1061
CHAR (SQL), 1061
CHARACTER
(SQL), 1061
VARYING(n) (SQL), 1061
CHARACTER(n) (SQL), 1061
Character (Java), 194
cache, 211
cost, 215
cast, 215
class, 215
constant, 215
elementary, 215
strings (SQL), 1060
charAt() (Java), 134
CharEscapes.cc, 216
CharEscapes.java, 219
ChatServer
(C++), 1045
(Java), 1031
ChatServer.cc, 1048
ChatServer.h (Qt), 1047
ChatServer.java, 1035
C header files used in C++, 386
C++ header files used in C++, 386
algorithm, 8
extended
  (C++), 42, 617
  (Java), 46, 669
field of, 30
generic, 547
hierarchy, 30
IsA, 31
role-based, 768
inner (Java), 91
instance of, 29
mixin, 721, 760–761
nested, 89
  static (Java), 89
parameterized
  (C++), 555
  (Java), 568
relationship
  HasA, 591
  IsA, 591
responsibility (UML), 592
role playing, 721, 768
scope operator (C++), 36
sub-, 31
super-, 31
template, 547
  (C++), 83, 555
variable, 422
ClassCastException (Java), 409, 568–569, 706
CLASSPATH environment variable, 12
classpath (Java), 11, 57, 570
  option, 5
clear() (Java), 182
clicked, a signal, 807
clicked(
  (GNOME/GTK+), 871
  (Qt), 859
client, 1027–1028
ClientHandler
  (C++), 1046
  (Java), 1031–1032
client–server networking model, 1027
ClientSocket.cc, 1041
ClientSocket.h (Qt), 1040
clone() (Java), 456–457
CloneArray1.java, 460
Cloneable (Java), 75–76, 457
ClonableX.java, 458
CloneBasic.java, 458
CloneClassTypeArr.java, 462
CloneNotSupportedException (Java), 457
cloning (Java), 457
close() (C++), 23
Closing (Qt), 1039
closing
  a stream
  (C++), 23, 247, 251
  (Java), 24, 261, 263, 266
CODE, 938
CODEBASE, 938
collaboration diagram (UML), 597, 603
Collator (Java), 145
Collection (Java), 75, 182, 184
GJ, 571
view, 193
CollectionMaxGeneric.java, 577
Collections (Java), 197
  Framework, 80
Color (AWT/Swing), 810, 813, 905
command-line
  arguments
    (C++), 19, 801
    (Java), 24
  compilation, 5
    (C++), 9
    (Java), 11
    (Qt), 802
comma separated values, 1073
comment delimiters, 9
commenting code, 9, 14
Common Desktop Environment, 791
communication (UML), 590
communication association (UML), 590
Comparable (Java), 81, 136, 579
Comparator (Java), 137, 145, 183, 186, 198,
  576–577, 579
compare()
  (C++), 115
  (Java), 137
compareTo() (Java), 81, 135–136, 183, 186
  comparing objects, 77
compiling
  (C++), 5, 9
  (GNOME/GTK+), 805
  (Java), 11
  (Qt), 802, 862
complex, a C++ header file, 8–9
Component (AWT/Swing), 797, 852, 903–904,
  916
component
  heavyweight (AWT/Swing), 793
  lightweight (AWT/Swing), 793
  peered (AWT/Swing), 793
  peerless (AWT/Swing), 793
ComponentEvent (AWT/Swing), 852
composite key (SQL), 1062
composition (UML), 591, 594
compound stream (Java), 264
concat() (Java), 129
declaration, 76
conceptual perspective (UML), 592
concurrent
computing, 963
processing, 602
statechart diagram (UML), 608
ConcurrentModificationException, 189, 205
condition
function (C++), 243
synchronization bar (UML), 611
variable (POSIX), 1010, 1013
connect
declaration (Qt), 844
signal with slot (Qt), 859
connected() (Qt), 1038
Connecting (Qt), 1039
connectionClosed() (Qt), 1038
Connection (Java), 1059, 1070
(Mysql++), 1075
connectToHost() (Qt), 1038
const (C++), 51, 76, 292, 317, 595
char*, 108, 116
function parameter, 345, 363
member function, 435
return type from a function, 364
constant-time complexity, 150
const_cast (C++), 234, 781
const_iterator (C++), 169, 568
ConstRefReturn.cc, 359
const_reverse_iterator (C++), 568
constructor, 33, 38
derived class
(C++), 621
(Java), 669
no-arg
(C++), 43, 286, 288–289, 307, 621, 623
(Java), 47, 290–297, 312–313, 669
one-arg, for implicit type conversion (C++), 535
order dependencies
(C++), 657
(Java), 676
overloading
(C++), 366
(Java), 371
ConstructorLeak.cc, 521
ConstructorLeakPlugged.cc, 528
ConstructorOrder.cc, 657
ConstructorOrderFoo.cc, 659
ConstructorOrderFoo.java, 676
Container (AWT/Swing), 797, 904
container
classes
(C++), 148
(Java), 182
GUI
intermediate (AWT/Swing), 794
intermediate (GNOME/GTK+), 796
intermediate (Qt), 795
top-level (AWT/Swing), 793
top-level (GNOME/GTK+), 796
top-level (Qt), 795
containment hierarchy, 793, 801–802
content pane (AWT/Swing), 797–798
conversion specifier (C), 15
count
assignment operator (C++), 166, 450, 452, 508
prototype, 452
constructor (C++), 113, 450, 452, 507
example definition, 452
for derived class, 624
for multiple bases, 625
prototype, 452
on return from function (C++), 359
copy
() (C++), 206, 566
CopyAssignX.cc, 454
copyInto() (Java), 195
CopyOnReturn.cc, 358
CountDestructorInvoc1.cc, 377
CountDestructorInvoc2.cc, 378
CountDestructorInvoc3.cc, 378
CountDestructorInvoc4.cc, 379
CountDestructorInvoc5.cc, 380
CountDestructorInvoc6.cc, 381
cout (C++), 8, 23, 239
cp, for classpath (Java), 570
CrazyWindow.c, 874
CrazyWindow.cc, 864
CrazyWindow.h (Qt), 864
CrazyWindow.java, 855
createEmptyBorder() (AWT/Swing), 817
createHorizontalGlue() (AWT/Swing), 816–817
createHorizontalTextPosition() (AWT/Swing), 822
createLineBorder() (AWT/Swing), 810, 813, 819, 822
createRigidArea() (AWT/Swing), 815, 817
createStatement() (Java), 1070
CREATE TABLE (SQL), 1060, 1063, 1073
currentTimeMillis() (Java), 55–56
D
daemon thread, 1003, 1025
darker() (AWT/Swing), 905
dashDotLine (Qt), 921
dashLine (Qt), 921
data
abstractions, uncoupling of, 726
encapsulation, 29
hiding, 30
member, 29, 32, 37
database, 1055
desktop-based, 1057
driver, 1055, 1058
management system, 1058
programming, 2
relational, 1056–1057
server-based, 1057
URL, 1059
DataInputStream (Java), 271, 273, 990
DataOutputStream (Java), 264, 269–270, 990
Date.cc, 425
Date.java, 432
DATE (SQL), 1061
datetimes (SQL), 1060
DB2, 1059
dBase, 1059
DBFriends1.java, 1071
DBFriends2.java, 1074
deadlock, 1013, 983
irresolvable, 987
dead state, 970
dec, a stream manipulator (C++), 253
DECIMAL (SQL), 1061
decimal integer literal, 221
decision activity (UML), 612
declaring
an identifier, 283
multiple names, 302
pointer types (C++), 297
decompling (Java), 975
decrement operator, overloading of (C++), 529
default arguments for C++ functions, 373,
initialization
array, 307,
(C++), 286
(Java), 293
template parameter (C++), 564
layout manager, 809
DefaultInit.cc, 287
DefaultInit.java, 294
DefaultInit2.cc, 288
DefaultInit2.java, 295
DefaultInit3.cc, 289
DefaultInit3.java, 296
DefaultInit4.cc, 290
DefaultInitClassArray.cc, 309
DefaultInitClassArray2.cc, 310
DefaultInitPrimArray.cc, 308
#define (C++), 319
defining an identifier, 283
delayedClosedFinished() (Qt), 1038
DELETE (SQL), 1068
delete (C++), 34, 46, 299, 334, 443, 520, 633
delete() (Java), 141
delete[] (C++), 46, 334, 444, 520
deleteCharAt() (Java), 135
delimiter characters, 192, 211
depth index, 798
define
a C++ container class, 149–150, 168, 174, 567
a C++ header file, 8
DequeFront.cc, 168
dereferring
an iterator (C++), 154
a pointer (C++), 297
operator (C++), 519
derivation
class (C++)
private, 664
protected, 664, 728
public, 617
class (Java), 669
DerivedAssignOp.cc, 626
derived class, 31
(C++), 42
private, 664
protected, 664
public, 621
constructor, 621
copy constructor, 624
destructor, 632
constructor, 669
DerivedConstructor.cc, 621
DerivedConstWithBaseNoArg.cc, 622
DerivedCopyConstruct.cc, 624
DerivedDestructorCase1.cc, 633
DerivedDestructorCase2.cc, 634
DerivedDestructorCase3.cc, 636
DerivedNameConflict.cc, 620
DerivedOverloadOp.cc, 629
DESCRIBE (SQL), 1064, 1069
deselecting a layout manager, 830
design patterns, 3, 789
desktop-based database, 1057
Dest.cc, 54
destroy() (AWT/Swing), 937
Destruct.cc, 441
Destruct2.cc, 442
destructor (C++), 34, 54, 166, 507, 617
default definition, 633
explicit invocation, 446
for derived class, 632
template parameter, 633
virtual, 727, 781
DestructNecessary.cc, 444
DestructWhenNot.cc, 445
detachstate (POSIX), 1003
diamond class hierarchy (C++), 785
DiffScope.cc, 370
Dimension (AWT/Swing), 813, 815–817
Dining Philosophers Problem, 1025
directive, using (C++), 9, 64
disjunctive activity (UML), 612
dispatcher thread (Java), 965
DNS lookup, 1038
Document (AWT/Swing), 854
DocumentEvent (AWT/Swing), 847, 854
DocumentListener (AWT/Swing), 847, 854
DotLine (Qt), 921
double, 213, 222

greatest positive value, 222
precision, 222
smallest positive value, 222
DOUBLE PRECISION (SQL), 1061
downcasting (C++), 780
dramatic result set (Mysql++), 1077
drawImage() (AWT/Swing), 916
drawRect() (AWT/Swing), 857
(Java), 905
(Qt), 920
drawString() (AWT/Swing), 908
driver manager, 1059
DriverManager (Java), 1059, 1070
DROP TABLE (SQL), 1062, 1067
DuplicateBase.cc, 743
dynamically expandable array, 75
dynamic binding (C++), 647
dynamic_cast (C++), 234, 700, 780
with object reference, 781

east (AWT/Swing), 810, 827
ECHO network service, 1028
elementAt() (Java), 195
EMBED tag, 947
emission
of event, 844
of signal, 844
Employee.h, 694
empty
space (AWT/Swing)
glue, 815
invisible component, 815
rigid area, 815
string
(C++), 113
(Java), 132
empty() (C++), 174–175
encapsulation, 3, 29–30
in simulated OO in C, 95
EnclosedClassAccess.cc, 87
EnclosingClassAccess.java, 90
end() (C++), 154, 565
endl, a stream manipulator (C++), 255
end-of-file, 16
ends, a stream manipulator (C++), 255
endsWith() (Java), 141
error() (Java), 408
error indicators for I/O stream (C), 21
ErrConnectionRefused (Qt), 1040
ErrHostNotFound (Qt), 1040
ErrSocketRead (Qt), 1040
escape
hexadecimal, 215
octal, 215
sequence, 215
event, 932
definition, 842
emission, 844
high-level (AWT/Swing), 845
high-level (Qt), 858
in UML, 606
low-level (AWT/Swing), 845
low-level (Qt), 858
processing, 842
by macros and messages, 844
by virtual function (Qt), 844
loop, 844
loop (AWT/Swing), 845, 857, 993
loop (GNOME/GTK+), 871
loop (Qt), 802, 858
queue, 842
source, 845
eventDestroy() (GNOME/GTK+), 807, 873
Event Dispatch Thread (AWT/Swing), 857, 942,
964–965, 993, 995, 999
Event[Guard]/Action (UML), 606
EventThreadDemo.java, 993
EventThreadDemo2.java, 995
"exactly one superclass" rule (Java), 680
exact numerics (SQL), 1060
Excel, 1059
Exception (Java), 399
exception
a C++ header file, 8–9
checked (Java), 399, 408
handler, 389
handling, differences between C++ and Java, 399
specification
(C++), 393
(Java), 403, 674
unchecked (Java), 399, 408
ExceptionUsage1.cc, 392
ExceptionUsage1.java, 402
ExceptionUsage2.cc, 392
ExceptionUsage2.java, 403
ExceptionUsage3.cc, 393
ExceptionUsage3.java, 403
ExceptionUsage4.cc, 394
ExceptionUsage4.java, 404
ExceptionUsage5.cc, 395
ExceptionUsage5.java, 405
ExceptionUsage6.cc, 396
ExceptionUsage6.java, 406
ExceptionUsage7.cc, 397
ExceptionUsage7.java, 407
ExceptionUsage8.cc, 397
ExceptionUsage8.java, 407
exec() (Qt), 802, 831–832, 834, 836
executable class, 13
execute() (Mysql++), 1076
executeQuery() (Java), 1070
exit()
(C), 19–20
(C++), 23
EXIT_FAILURE (C), 19–20
expand (GNOME/GTK+), 838
expandable empty space (AWT/Swing), 816
ExplicitCast1.cc, 232
ExplicitCast1.java, 234
ExplicitCast2.cc, 233
ExplicitCast2.java, 235
explicit
constructor invocation (Java), 422
keyword for suppressing implicit type conversion (C++), 527, 533–536
type conversion, 223
(C++), 232, 234, 618
(Java), 234
extend (UML), 590
extended
class (C++), 42
class (Java), 47,
extending
a class
C++, 664,
Java, 669
an interface (Java), 686
extends (Java), 47, 669
extension point (UML), 590
extern, 69
‘C’ directive (C++), 386, 488
extracting substring
(C++), 122
(Java), 141
extraction operator (C++), 16
F
fail() (C++), 243
fail-fast (Java), 189, 204
fast mutex (POSIX), 1011
fclose() (C), 19
ferror() (C), 19
field, 30, 32, 37
FILE* (C), 20
FileCopy.c, 19
FileCopy.cc, 21
FileCopy.java, 23
FileDialog (AWT/Swing), 883
FileInputStream (Java), 24
FileOutputStream (Java), 263, 266
file pointer
(C), 20
(C++), 239
(Java), 266
scope (C++), 303–304
FileReader (Java), 27, 205
FileReader (Java), 263, 266, 269
fill
(AWT/Swing), 826
(GNOME/GTK+), 838
(Qt), 920
methods for Java arrays, 315
methods for C++ containers, 182
fill
(C++), 253
(Java), 316
fill
(C++), 253
(Java), 316
fill
(AWT/Swing), 816
fill (Qt) (AWT/Swing), 908
finalization (Java), 55, 446
finalizer (Java), 55, 446–447
finalizer thread (Java), 965
final (Java), 317, 365, 595, 690
for blocking inheritance, 49
for read-only variable, 51, 76
function parameter, 345
finally (Java), 401, 405
fi (C++)
(C++), 120, 159, 182, 565–566
finally (C++), 120
fi (C++)
(C++), 120
fi (C++)
(C++), 120
fi (C++)
(C++), 120
fi (C++)
(C++), 120
a C++ header file, 8, 22, 238
a read and write stream (C++), 239
FTP network service, 1028
function

calling modes
  differences between C++ and Java, 357
call operator (C++), 538
inlining (C++), 345, 360
object (C++), 79, 176, 207, 538
overloading, 345
  (C++), 366, 368
overriding, 44
  (C++), 46, 49
  (Java), 47
parameterized
  (C++), 557
  (Java), 577
prototype (C++), 22, 346
prototype scope (C++), 303, 305
scope (C++), 303, 305
signature, 369, 651
stack, 388
template
  (C++), 556–557
  (Java), 577
virtual (C++), 44
vs. method, 345
functional, a C++ header file, 8, 176
functor (C++), 538, 693
fundamental types, 213

G

g++, 9
garbage collection, 964
  C++, 334
  Java, 55, 335, 446
gboolean (GNOME/GTK+), 880
  gc() (Java), 55
gchar (GNOME/GTK+), 796
GC.java, 447
gcount() (C++), 245, 247
GC_Resurrection.java, 449
GDK_2BUTTON_PRESS, 932
GDK_3BUTTON_PRESS, 932
GDK_BUTTON_PRESS, 932
GDK_BUTTON_RELEASE, 932
GDK, 795
GDK_DRAG_ENTER, 932
GDK_DRAG_LEAVE, 932
GDK_DRAG_MOTION, 932
GdkEvent*, 932
GdkEvent4Button*, 932
gdk_imlib_create_image from xpm_data(), 927
gdk_imlib_image, 927
GDK_MOTION_NOTIFY, 932
generalization, 591, 769
INDEX 1175

UML, 590
generic algorithms
(C++), 181
(Java), 197
class, 547, 568
program, 548
Generic Java, 569
getAllFonts() (AWT/Swing), 908
getAppletContext() (Java), 956
getAppletInfo() (AWT/Swing), 946
getAvailableFontFamilyNames() (AWT/Swing), 908
g() (C++), 15, 17, 21–22, 243, 247
getC() (C), 19, 21
getc() (C), 14–15
getColumnCount() (Java), 1071
getContentPane() (AWT/Swing), 798
getConnnection() (Java), 1059
getContentPane() (AWT/Swing), 915
getFont() (AWT/Swing), 883
getFont() (Java), 266
getID() (AWT/Swing), 852, 914
getImage()
(AWT/Swing), 915
(Java), 991
gGetInputSteam() (Java), 1029
getInsets() (AWT/Swing), 904
getLine() (C++), 245
getLocalGraphicsEnvironment() (AWT/Swing), 908
getMetaData() (Java), 1071
getOutputStream() (Java), 1029
getInputStream() (Java), 1029
getPoint() (AWT/Swing), 915
getProperty() (Java), 957
GET request, 1028–1029
GnomeCanvas, 926, 932
GnomeCanvasGroup, 874, 926
GnomeCanvasItem, 874
GnomeCanvasPoints, 928
GnomeCanvasPoints*, 933
GnomeCanvasRoot(), 874
GnomeCanvasItem*(), 927
Gnome-config, 805
GnomeCanvas, desktop environment, 790, 793, 795
Gnome/GTK+, 2, 795
GnomeInit(), 804
GnomeUIInfo, 894, 897
GnomeUIINFO-END, 897
GnomeUIINFO-Help, 897
GnomeUIINFO-_SEPARATOR, 897
GnomeUIINFO-Subtree, 897
GNU
C compiler, g++, 9
General Public License, 1058
Image Manipulation Program (C), 795
Object Modeling Environment (C), 795
golden proportion, 789
good() (C++), 243
GPL, 1058
gprint() (Gnome/GTK+), 796
gprint() (GNOME/GTK+), 796
GIMP (GNOME/GTK+), 795
GIMP Drawing Kit (GNOME/GTK+), 795
GIMP Took Kit (GNOME/GTK+), 795
gint16 (GNOME/GTK+), 796
gint8 (GNOME/GTK+), 796
gint (GNOME/GTK+), 796, 880
gjc (Java), 569
gjc.Main (Java), 570
gjcr (Java), 569
GJ (Java), 569
glass pane (AWT/Swing), 797–799
glib library (GNOME/GTK+), 795
global namespace (C++), 62, 386
overload definition (C++), 498
scope (C++), 62
glue
(AWT/Swing), 816
code (Qt), 862
g_malloc() (GNOME/GTK+), 796
Gnome_appCreateMenus(), 896
Gnome_appNew(), 804
Gnome_app PIXMAP_FILENAME, 896
Gnome_app PIXMAP_NONE, 896
Gnome_app_UL_HELP, 895, 897
Gnome_app_UL_ITEM, 895
Gnome_app_UL_SUBTREE, 895, 897
Gnome, 871
GnomeCanvas, 926, 932
GnomeCanvasGroup, 874, 926
GnomeCanvasItem, 874
GnomeCanvasSet(), 874
GnomeCanvasItem new(), 874, 926
GnomeCanvasLine getType(), 933
GnomeCanvasPoints, 928
GnomeCanvasPoints*, 933
GnomeCanvasRoot(), 874
GnomeCanvasSetScrollRegion(), 927
Gnome-config, 805
Gnome, desktop environment, 790, 793, 795
Gnome/GTK+, 2, 795
GnomeInit(), 804
GnomeUIInfo, 894, 897
GnomeUIINFO-END, 897
GnomeUIINFO-Help, 897
GnomeUIINFO-_SEPARATOR, 897
GnomeUIINFO-Subtree, 897
GNU
C++ compiler, g++, 9
General Public License, 1058
Image Manipulation Program (C), 795
Object Modeling Environment (C), 795
golden proportion, 789
good() (C++), 243
GPL, 1058
gprint() (Gnome/GTK+), 796
grant table, 1058
graphical user interface, 2
Graphics2D (AWT/Swing), 903, 910
Graphics (AWT/Swing), 794, 854, 903–905, 907
graphics context, 903
GraphicsDevice (AWT/Swing), 909
GraphicsEnvironment (AWT/Swing), 908
green threads, 964
GridBagConstraints (AWT/Swing), 825, 827
GridBagLayout (AWT/Swing), 825, 827
GridBagLayout (AWT/Swing), 809, 825
GridBagLayoutTest.java, 828
gridheight (AWT/Swing), 826
GridLayout (AWT/Swing), 809, 819, 849
GridLayoutTest.java, 819
gridwidth (AWT/Swing), 826
gridx (AWT/Swing), 826
gridy (AWT/Swing), 826
GTK+, 790, 793
GtkAttachOptions, 840
GtkBin, 880
gtk_box_pack_end(), 839
gtk_box_pack_start(), 838–839
GtkButton, 879
gtk_button_new_with_label(), 807, 840
GTK, 795
GTK+, 791, 795, 871
GtkContainer, 880
gtk_container_set_border_width(), 805
gtk_editable_get_chars(), 874
gtk_editable_get_position(), 874
GtkHBox, 837
gtk_hbox_new(), 837
gtk_idle_add(), 881
gtk_main(), 805, 871, 881
gtk_main() (GNOME/GTK+), 808
gtk_main_quit(), 880
gtk_main_quit() (GNOME/GTK+), 805, 808
GtkModifierType, 896
GtkObject, 796, 880
GTK_POLICY_NEVER, 874
gtk scrolled_window_add_with_viewport(), 873
gtk scrolled_window_new(), 873
gtk scrolled_window_set_policy(), 873–874
gtk_signal_connect(), 871, 879, 932
gtk_signal_emit(), 871
gtk_signal_emit_by_name(), 871
gtk_signal_query(), 880
GtkSignalQuery, 880
GtkTable, 837, 839
gtk_table_attach_defaults(), 840
gtk_text_new(), 883
gtk_text_set_line_wrap(), 873
gtk_type_name(), 880
GtkVBox, 837
gtk_vbox_new(), 839
GtkWidget, 796, 880
gtk_window_new(), 804, 873
GTK_WINDOW_TOPLEVEL, 804
guard (UML), 606
guchar (GNOME/GTK+), 796
GUI, 2
history, 790

handler, exception, 389
handshaking for reliable transmission, 1028

HasA relationship, 591
hash
map
(C++, 180
(Java), 184, 191
table
(C++, 151
(Java), 184
hash_map, a C++ container class, 180
HashMap, a Java container class, 184, 191
HashSet, a Java container class, 189
HashTable, a Java container class, 184
hasMoreTokens(), 205
hasMoreTokens() (Java), 27
hasNext() (Java), 186, 190
hasPrevious() (Java), 186
heap, 299, 632
heavyweight component (AWT/Swing), 793
HEIGHT, 938
height-balanced binary tree, 183, 191
HelloThreadWithJoin.cc, 1018
hex, 216
a stream manipulator (C++), 253
dump, 246
hexadecimal
digit, 216
ecape, 215
integer
literal, 221
notation, 221
HideScope.cc, 304
hierarchical structures, 1
hierarchy, 30
role-based, 768
high-level event, 845, 858
hints to a layout manager, 815
homogeneous (GNOME/GTK+), 837, 840
HORIZONTAL (AWT/Swing), 826
hostFound() (Qt), 1038, 1041
HostLookup (Qt), 1039
hotjava, 957
HSIZE, 938
HTML, 937, 939, 947, 990
HTMLConverter tool, 947
HTTPD server, 1028–1029, 1040

I
IDE, 845
identifier, 211–212
declaration, 283
default initialization, 283
definition, 283
scope, 283
Idle (Qt), 1039
idle function, 881
ID number, 880
IEEE 754 Standard, 222, 227
ifstream (C++), 21–22, 237–238
ignore() (C++), 245
IllegalArgumentException (Java), 409
Image (AWT/Swing), 915, 940, 942
ImageIcon (AWT/Swing), 810, 813, 819, 822, 940, 942–943
ImageLoadAndDisplay.java, 917
image loading
asynchronous, 915
monitoring of, 915
ImageObserver (AWT/Swing), 916
immutable, 129, 200
implementation inheritance (C++), 665
ImplementationInheritance.cc, 667
implementation perspective (UML), 592
implementing an interface (Java), 77
implicit type conversion, 223
(C++), 224, 232, 533
for class-types (C++), 534
(Java), 228
import (Java), 24, 56–57, 605
in (C++), 239
include (UML), 590
including C header files in C++, 386
increment operator overloading (C++), 529
indefinite number (UML), 593
indexOf() (Java), 140
IndexOutOfBoundsException (Java), 315
redirect, 732
indirection (C++), 297
infinite, 222
inheritance, 3, 29, 31, 43
in simulated OO in C, 95
loop (C++), 760
multiple (C++), 25
inherited (POSIX), 1004
init) (AWT/Swing), 903, 936–937, 941, 991
initial capacity, 186, 190
initialCapacity, 197
initialization of
arrays
(C++), 301, 307–308
(Java), 311
base-class subobject
(C++), 623
(Java), 669
class type
default (C++), 287
default (Java), 296
const class member (C++), 292
reference type (C++), 292, 330
object reference
(Java), 333
static class members (C++), 423
static class members (Java), 433
static const class members (C++), 422
static final (Java), 430
variables
default (C++), 286
default (Java), 293
initialization static (C++), 422
inlining (C++), 36, 345, 360
inner class (Java), 91
InnerClass.java, 92
InnerClassThisPrefix.java, 92
input
operator (C++), 16
stream manipulators (C++), 255
stream object state (C++), 23
InputEvent (AWT/Swing), 845
InputIterator (C++), 566
InputStream (Java), 18, 1029, 259, 990
INSERT (SQL), 1060, 1062
insert()
(C++), 123, 153, 159
(Java), 141
insert substring
(C++), 123
(Java), 141
insertElement() (Java), 196
inserter iterator (C++), 567
insertion
formatted (C++), 9
operator (C++), 9
insertUpdate() (AWT/Swing), 854, 999
Insets (AWT/Swing), 904
insets (AWT/Swing), 827
instance of a class, 29
instantiating a class, 2, 33
INT (SQL), 1060
int, 213, 220
InCompararor (Java), 577
INTEGER (SQL), 1061
integer, 211
Integer (Java), 18, 939
Integer.java, 579
integer
literal, 221
decimal, 221
hexadecimal, 221
octal, 221
types, 220
IntegerGeneric.java, 581
Integer.parseInt() (Java), 17
integral types
(C++), 223
(Java), 223
interaction diagram (UML), 588, 597
Interactive Design Environment, 845
interface (Java), 3, 74–75, 680
for packaging constants, 689
nested, 89  
parameterized, 573  
interference, thread, 971  
Interleaved.cc, 470  
interleaved classes, 298, 466  
Interleaved.java, 481  
intermediate GUI containers, 794–796  
internal() (Java), 130  
InternalFrame (AWT/Swing), 794  
internal linkage (C++), 69, 304  
International Standards Organization, 8  
internet auction example, 588  
InterruptedException (Java), 984  
INTERVAL DAY (SQL), 1061  
intervals (SQL), 1060  
intValue() (Java), 191  
invariances, 789  
invisible component (AWT/Swing), 816  
ineventDispatchThread() (AWT/Swing), 1000  
ineventLater() (AWT/Swing), 1000  
invoking  
a function on an object, 16  
object, 503  
I/O  
binary  
(C), 20  
(C++), 246  
Java, 261  
character based  
(C++), 238  
Java, 261  
modes, 16  
asynchronous, 16  
blocking, 16  
nonblocking, 16  
stream hierarchy  
(C++), 237  
Java, 258  
IOException (Java), 25  
iosanip, a C++ header file, 8, 256  
ios, a C++ header file, 8  
ios::app (C++), 238–239  
ios::ate (C++), 239, 248  
ios::binary (C++), 239  
ios::cur (C++), 241  
ios::file (C++), 255  
ios::in (C++), 239, 241  
ios::left (C++), 253  
ios::noshownum (C++), 254  
ios::noskipws (C++), 255  
ios::out (C++), 239, 241  
ios::scientific (C++), 254  
ios::showbase (C++), 254  
ios::showpoint (C++), 254  
ios::skipws (C++), 255  
ios::trunc (C++), 239  
iosfwd, a C++ header file, 8  
iostream  
a C++ header file, 8  
a C++ I/O stream class, 237  
IP address, 1027–1028  
ipadx, padding variable (AWT/Swing), 827  
ipady, padding variable (AWT/Swing), 827  
isResolvable deadlock, 987  
isA relationship, 31, 591, 682  
isEmpty() (Java), 183  
isEventDispatchThread() (AWT/Swing), 993  
ISO, 8, 421  
ISO-Latin-1, 133  
istream  
a C++ header file, 8  
a C++ input stream class, 237  
istringstream (C++), 257  
istrstream (C++), 257  
itemStateChanged() (AWT/Swing), 822  
Iterator (Java), 186, 190  
GI, 571  
iterator  
a C++ header file, 8  
adapter (C++), 567  
(C++), 153, 171, 181, 565  
BidirectionalIterator, 567  
class, 565  
forward incrementing, 566  
initialization, 565  
RandomAccessIterator, 567  
Java, 190  
dereferencing (C++), 154  
iterator() (Java), 183  
J  
JApplet (AWT/Swing), 793–794, 797, 936, 1000  
JAR archive (Java), 12  
jar, the Java archive tool, 12  
Java  
bytecode, 11  
Collections Framework, 205  
compilation, 5, 11  
Database Connectivity, 1055  
enabled browser, 939  
Foundation Classes, 794  
Plug-in for Swing applets, 947  
Runtime Environment, 947  
Virtual Machine, 11, 936, 946–945  
java, the Java application launcher, 11  
java.awt.event package, 793  
java.awt.image package, 916  
java.awt package, 793  
javac, the compiler for Java, 11  
javadoc, Java documentation tool, 14  
java.io package, 24, 27, 267  
java.lang package, 56
java.net package, 1028, 1032
javap, the Java class file disassembler, 975
java.sql package, 1059, 1070
java.util package, 185–186, 190, 193
  GJ version, 569
javax.swing.event package, 793
javax.swing package, 793, 797
JButton (AWT/Swing), 794, 827, 850
JComboBox (AWT/Swing), 821
  editable, 821
  uneditable, 821
JComponent (AWT/Swing), 798–799, 850, 903
JDBC (Java), 1055, 1057, 1070
  driver, 1055
JDBC-ODBC bridge driver, 1059
JDialog (AWT/Swing), 793–794, 797, 1000
JFC (AWT/Swing), 794
JFrame (AWT/Swing), 793–794, 797, 948, 1000
JIT, just-in-time Java compilation, 11, 965
JLabel (AWT/Swing), 810, 813, 819, 821–822, 940, 942
JLayeredPane (AWT/Swing), 797
JList (AWT/Swing), 816, 822
JTextField (AWT/Swing), 794
JVM, 11, 936, 942

K

KEY (SQL), 1061
  keyboard
  accelerator, 889, 895
  buffer, 16
  KeyEvent (AWT/Swing), 845, 852
  keySet() (Java), 194
  key-sorted order, 184
  <key, value> pair, 149, 151, 178
  keyword, 211
  kind (UML), 596
    in, 596
    inout, 596
out, 596
Koenig.cc, 70
Koenig lookup (C++), 70, 304

L

label, 305
lastIndexOf() (Java), 140
Latin-1, 213, 218
layered pane (AWT/Swing), 797–798
layout management, 808–809
  (AWT/Swing), 809
  (GNOME/GTK+), 837
  (Qt), 830
LayoutManager (AWT/Swing), 830
lazy instantiation, 416
ldconfig, 802
ld.so.conf, 802
least-common-denominator approach, 794
left, a stream manipulator (C++), 253
left_attach (GNOME/GTK+), 840
Left.java, 954
length (Java), 314
length( ),
  (C++), 124
  (Java), 132
less, a function object (C++), 176
lexicographic ordering, 109, 136
libqt-mt (Qt), 1019
lightweight (AWT/Swing)
  component, 793–794
  container, 797
 limits , a C++ header file, 8
linear time complexity, 150
linkage, internal (C++), 69
LinkedList
  (C++), 548
    templatized, 555
    (Java)
      a container class, 186
      GJ, 571
LinkedList.cc, 551
LinkedList<double> (C++), 548
LinkedListGeneric.cc, 557
LinkedListGeneric.java, 574
LinkedList<int> (C++), 548
LinkedList<Integer> (Java), 549
LinkedList.java, 571
LinkedListSpecialized.cc, 561
LinkedList<string> (C++), 548
LinkedList<String> (Java), 549
Linux, 1058, 802, 964–965
LinuxThreads, 1013
List
  a Java container class, 75–76, 182, 186, 194, 569
  (AWT/Swing), 954
list
(C++)
a container class, 149–150, 170, 567
a header file, 8
(Java), 182, 185, 194
ListGeneric.java, 569
ListIterator (Java), 186
ListMixedType.java, 569
ListOps.cc, 171
ListOps.java, 186
List<String> (Java), 569
literal
character, 215
floating-point, 223
integer, 221
string, 108, 211, 216, 219
little-endian, 134, 250
LOAD DATA INFILE (SQL), 1073
load-factor, 190
local
identifier, 304
loopback address, 1035
Locale (Java), 145
locale, a C++ header file, 8
lock, 979, 984
long, 213, 220
double (C++), 213, 222
longjmp (C), 386
look-and-feel, 791–794, 797
loopback address, 1035, 1053
low-level event, 845, 858
M
macro (C++), 319
macros and messages, 844
main window (Qt), 801
makefile, 805–806, 808, 868, 878, 902, 1043, 1052
Makefile ChatServer, 1052
Makefile ClientSocket, 1043
Makefile GTK CrazyWindow, 878
Makefile GTK FirstWindow, 806
Makefile GTK RenderGraphics, 931
Makefile GTK Sketch, 935
Makefile QL WindowWithMenu, 902
Makefile QL CrazyWindow, 868
Makefile QL WindowWithMenu, 889
malloc(), 98, 110, 796, 1013
Manager.cc, 701
Manager.java, 707
Mandrake, 802
mangling, name (C++), 69, 386
Map, a Java container class, 191
map (C++)
a container class, 149, 151, 178, 208, 567
a header file, 8
(Java), 184, 191
Map.Entry (Java), 194
MapHist.cc, 179
MapHist.java, 193
mapping threads to native threads, 964
mastering OO paradigm, 2
Math (Java), 56
max() (Java), 577
maximum size recommendation, 815
MediaTracker (AWT/Swing), 915, 942
member
access operator (C++), 35, 39, 520
function, 29
overload definition (C++), 501
initialization syntax (C++), 36, 293, 658
of a class, 32, 37
memory
a C++ header file, 8–9, 529
allocation
heap (C++), 299, 333
heap (Java), 335
stack (C++), 299
deallocation (C++), 334
leak (C++), 46, 636
segmentation fault, 299
Menu (AWT/Swing), 882
menu, 881
MenuBar (AWT/Swing), 882
MenuItem (AWT/Swing), 882
merge-sort, 137, 197, 315
merging lists (C++), 171
message (UML)
in a sequence diagram, 597
sequence number, 603
status, 598
to self, 598
meta (Qt)
object code, 862
object compilation, 859, 862, 1043
method, 30
invocation
asynchronous, 601
synchronous, 601
MFC, 791, 844
MI (C++), 721
Microsoft Foundation Classes, 791, 844
min_element (C++), 182
minimum
capacity increment, 197
size recommendation, 815
minimumSizehint() (Qt), 837
MI_Utilities.h (C++), 752
Miscellaneous, 762
mixin class, 721, 760–761
<table>
<thead>
<tr>
<th>Index</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mixing C and C++, 488</td>
<td></td>
</tr>
<tr>
<td>mm.mysql, 1059</td>
<td></td>
</tr>
<tr>
<td>mm.mysql.Driver, 1070</td>
<td></td>
</tr>
<tr>
<td>moc (Qt), 1043 compiler, 859, 862</td>
<td></td>
</tr>
<tr>
<td>monitor, 979, 984</td>
<td></td>
</tr>
<tr>
<td>monitoring image loading, 915 of a port by a server, 1032</td>
<td></td>
</tr>
<tr>
<td>most-derived object (C++), 736</td>
<td></td>
</tr>
<tr>
<td>Motif, 791–792, 794</td>
<td></td>
</tr>
<tr>
<td>MOUSE_CLICKED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseClicked() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseDoubleClickEvent() (Qt), 923</td>
<td></td>
</tr>
<tr>
<td>MOUSE_DRAGGED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseDragged() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>MOUSE_ENTERED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseEntered() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>MouseEvent (AWT/Swing), 845, 852–853, 914–915 (Qt), 923</td>
<td></td>
</tr>
<tr>
<td>mouse event, low-level, 923</td>
<td></td>
</tr>
<tr>
<td>MOUSE_EXITED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseExited() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>MouseListener (AWT/Swing), 853, 913</td>
<td></td>
</tr>
<tr>
<td>MouseMotionListener (AWT/Swing), 853, 914</td>
<td></td>
</tr>
<tr>
<td>mouseMoved() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseMovedEvent() (Qt), 919, 923</td>
<td></td>
</tr>
<tr>
<td>MOUSE_PRESSED (AWT/Swing), 913</td>
<td></td>
</tr>
<tr>
<td>mousePressed() (AWT/Swing), 913</td>
<td></td>
</tr>
<tr>
<td>MOUSE_PRESSED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mousePressed() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mousePressEvent() (Qt), 919, 923</td>
<td></td>
</tr>
<tr>
<td>MOUSE_RELEASED (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseReleased() (AWT/Swing), 914</td>
<td></td>
</tr>
<tr>
<td>mouseReleaseEvent() (Qt), 923</td>
<td></td>
</tr>
<tr>
<td>MultiJmp.cc, 387</td>
<td></td>
</tr>
<tr>
<td>MultiClassifiers.java, 421</td>
<td></td>
</tr>
<tr>
<td>MultiCustomerAccount.c, 1015</td>
<td></td>
</tr>
<tr>
<td>MultiCustomerAccount.cc, 1022</td>
<td></td>
</tr>
<tr>
<td>MultiCustomerAccount.java, 985</td>
<td></td>
</tr>
<tr>
<td>multilevel return in C, 386</td>
<td></td>
</tr>
<tr>
<td>multimap, a C++ container class, 149, 151 multiple inheritance (C++), 3, 75, 721 triggers (UML), 613 multiplicity, 592 for attributes (UML), 595</td>
<td></td>
</tr>
<tr>
<td>MultiPolymorphism.java, 683</td>
<td></td>
</tr>
<tr>
<td>multiprocessing, 611, 963</td>
<td></td>
</tr>
<tr>
<td>multiset, a C++ container class, 150 multithreading, 611 for animation, 990</td>
<td></td>
</tr>
<tr>
<td>mutable, 134</td>
<td></td>
</tr>
<tr>
<td>mutating algorithm (C++), 182</td>
<td></td>
</tr>
<tr>
<td>mutex (POSIX) fast, 1011 kind, 1010 lock, 1010 recursive, 1011</td>
<td></td>
</tr>
<tr>
<td>mutual exclusion for thread synchronization, 1010</td>
<td></td>
</tr>
<tr>
<td>MyDrawPanel.cc, 867</td>
<td></td>
</tr>
<tr>
<td>MyDrawPanel.h (Qt), 867</td>
<td></td>
</tr>
<tr>
<td>MySQL, 1058</td>
<td></td>
</tr>
<tr>
<td>terminal monitor, 1063</td>
<td></td>
</tr>
<tr>
<td>Mysql++, 1075, 1057</td>
<td></td>
</tr>
<tr>
<td>mysqladmin, 1058</td>
<td></td>
</tr>
<tr>
<td>mysqld, 1058</td>
<td></td>
</tr>
<tr>
<td>MyString, 506</td>
<td></td>
</tr>
<tr>
<td>MyString.cc, 514</td>
<td></td>
</tr>
<tr>
<td>MyTextPanel.cc, 865</td>
<td></td>
</tr>
<tr>
<td>MyTextPanel.h (Qt), 865</td>
<td></td>
</tr>
</tbody>
</table>

**N**
- NAME, 938
- name conflict, 619, 730, 747, 749 hiding (C++), 619, 671 lookup (C++), 619, 671 argument-dependent, 70 lookup (Java), 671 mangling (C++), 69, 386, 488 NameConflictDataMem.cc, 750 NameConflictMemFunc.cc, 747 nameless namespace, 361 NameLookup.cc, 671 NameLookup.java, 671 namespace (C++), 9, 61 alias, 304 global, 386 nesting, 67 qualify ed name, 303 scope, 303 std, 22, 386 unnamed, 69 Namespace.cc, 62 Namespace2.cc, 65 Namespace3.cc, 65 Namespace4.cc, 66 namespace.h (Qt), 921 NamespaceNested.cc, 67 NaN, Not a Number, 222 narrowing conversion (Java), 228 primitive conversion (Java), 228 Narrowing.java, 231 NATIONAL CHARACTER(n) (SQL), 1061 native GUI toolkit, 792 threads, 964
natural (Java)

comparison method, 81
ordering, 80, 137, 183, 579

navigability (UML), 592

nested

block, 304
class

(C++), 84 (Java), 84, 89, 93, 571
enumeration (C++), 89
interface (Java), 84, 89, 93
namespaces (C++), 67, 304
typedef (C++), 84

NestedInterface.java, 93
NestedClassAsType.java, 90
NestedClass.cc, 84
NestedClassDefsNotInline.cc, 85
NestedClass.java, 89
NestedTypes.cc, 88

network

programming, 2
-transparent computing, 790

new

a C++ header file, 8–9
(C++), 34, 113, 299, 333, 443
(Java), 38, 131, 334
(UML), 599

new[ ] (C++), 333, 520

newConnection() (Qt), 1045
newline character, 215

next-element operation, 150

next() (Java), 1071, 186, 190

nextToken() (Java), 205, 27

no-arg constructor

(C++), 43, 288, 307, 419, 507, 623
(Java), 47, 312–313, 669

NoArgMissing.cc, 419

Nocase (C++), 208

NoCopyOnReturn.cc, 359

NONE (AWT/Swing), 826

nonmutating algorithm (C++), 182

nonrealtime scheduling, 1004

nontype parameter
template declaration (C++), 563

NoPen (Qt), 921

NORTH (AWT/Swing), 810, 827
NORTHEAST (AWT/Swing), 827
NORTHWEST (AWT/Swing), 827

noshowpoint, a stream manipulator (C++), 254

noskipws, a stream manipulator (C++), 255

NoSuchMethodError (Java), 408

notify() (Java), 984

notifyAll() (Java), 983–984

NOT NULL (SQL), 1062

nparams (GNOME/GTK+), 880

npos (C++), 120

null, 293

pointer (C++), 298

NullPointerException (Java), 409

NumberFormatException (Java), 18

NUMERIC (SQL), 1061

numeric

a C++ header file, 8
escape
(C++), 215
(Java), 218
types (Java), 223

O

OBJECT tag (HTML), 947

Object

(Java), 75, 79, 446, 547, 984
 Management Group, 588

object

cloning (Java), 457
deserialization (Java), 279
destruction
(C++), 54, 441
(Java), 446
finalization (Java), 55, 446–447
reference, 329
(C++), 329
(Java), 332
serialization (Java), 279

ObjectIO.java, 280

ObjectInputStream (Java), 280

ObjectOutputStream (Java), 280

oct, a stream manipulator (C++), 253

octal

escape, 215
integer

literal, 222
notation, 221

number, 216

ODBC, 1059

ofstream (C++), 21–22

OMG, 588

one-argument constructor (C++), 535

on-line Java documentation, 3

OO, 1, 29
design, 3

Open DataBase Connectivity, 1059

openmode (C++), 239

open source, 1058, 795

operation (UML), 590, 595

modifier, 596
query, 596

operator, 211

‘()’ (C++), 176
arity, 498, 504

binary, 499
equality (C++), 513
INDEX 1183

for dereferencing (C++), 519
for member access (C++), 520
function (C++), 498
insertion (C++), 9
left bitwise shift (C), 9
output (C++), 9
overload definition
  global (C++), 498
  member function (C++), 501
overloading (C++), 53, 497
decrement operator, 529
for derived class, 628
increment operator, 529
key points, 497
< for sorting, 540
precedence, 497
relational (C++), 513
scope (C++), 89
token, 498
unary, 504
operator<<() (C++), 628
ORDER BY (SQL), 1067, 1069
ostream
  a C++ file, 8
  a C++ output stream class, 237
ostream_iterator (C++), 206
ostreamstream (C++), 257
out (C++), 239
OutOfMemoryError (Java), 408
OutputIterator (C++), 566
output
  operator (C++), 9
  stream
    classes (C++), 237
    classes (Java), 264
    manipulators (C++), 252
    object (C++), 8
    object (Java), 13
    standard (C++), 239
    standard (Java), 265
OutputStream (Java), 259, 989, 1029
overload
  resolution, 345
  (C++), 369
  (Java), 371
Overload.java, 372
Overload2.java, 382
OverloadUnaryGlobal.cc, 504
OverloadUnaryMemb.cc, 505
OverrideAccessRestrict.cc, 653
OverrideAccessRestrict.java, 674
override definition for a function, 44
OverrideExceptionRestrict.cc, 654
OverrideReturnRestrict.cc, 652
overriding
  a function
    (C++), 46
    (Java), 47
  function
    (C++), 651
    (Java), 670
  restrictions (C++), 651
  restrictions (Java), 673

P

pack() (AWT/Swing), 800, 810, 813, 817, 819, 822, 828, 1000
package
diagram (UML), 588, 604
(Java), 3, 13, 55, 57, 71
qualified name, 56
padding
  (GNOME/GTK+), 838
variable
  ipadx (AWT/Swing), 827
  ipady (AWT/Swing), 827
paint() (AWT/Swing), 903–904, 916
paintBorder() (AWT/Swing), 903
paintChildren() (AWT/Swing), 903
paintComponent() (AWT/Swing), 854, 903–904, 942
paintEvent() (Qt), 918–919, 924
pair, 176
pallet, 798
pane (AWT/Swing)
  content, 797
  glass, 797–799
  layered, 797–798
  root, 797
Panel (AWT/Swing), 936, 948
parameterization, 549
parameterized
class
  (C++), 550, 555
  (Java), 568
function (C++), 550
  interface (Java), 573
  method (Java), 576
parameter list (UML), 596
PARAM tag (HTML), 939, 954
parseInt() (Java), 17–18, 939
pass argument to function, 345
  by pointer (C++), 345
  by reference (C++), 345
  by value
    (C++), 345
    of object reference (Java), 345
pass by pointer
  class type argument (C++), 351
  primitive type argument (C++), 347
pass by reference
  class type argument (C++), 352
  primitive type argument (C++), 349
pass by value
  class type argument
    (C++), 350
    (Java), 354
  of object reference (Java), 354
  primitive type argument
    (C++), 346
    (Java), 353
PassClassTypeByPointer.cc, 351
PassClassTypeByRef.cc, 352
PassClassTypeByValue.cc, 350
PassClassTypeByValue.java, 354
passing arguments
  (C++), 346
  (Java), 353
PassPassByPointer.cc, 347
PassPassByRef.cc, 349
PassPassByValue.cc, 347
PassPassByValue.java, 353
peek() (C++), 15–16, 246
peer class (AWT/Swing), 793
peered component (AWT/Swing), 794
peerless component (AWT/Swing), 793
performance penalty
  for polymorphic function (C++), 650
Photoshop, 795
pipe (C++), 23
PipedInputStream (Java), 259, 988, 990
PipedOutputStream (Java), 259, 988, 990
PlainDocument (AWT/Swing), 854
Point (AWT/Swing), 915
pointer
  to C++ function, 374
  to class member (C++), 463
  type (C++), 297
PointerDirectToMember.cc, 464
polyline, 924, 932
Polymorph.cc, 45
polymorphic
  behavior, 638
  type (C++), 646, 781
PolymorphicTypes.cc, 782
polymorphism, 3, 29, 31, 682, 728, 918
  in simulated OO in C, 98
Polymorph.java, 48
pop_back() (C++), 154, 174
pop() (C++), 174–175
pop_front() (C++), 168
port, 1027–1028
number, 1027–1028
Portable Operating System Interface, 1002
POSIX, 1002
  thread attributes, 1003
  threads, 964
pos() (Qt), 924
precedence of an operator, 497
precision
  of output (C++), 252
  of SQL data types, 1061
preemptive scheduling, 964
PretendGiant.cc, 520
previous() (Java), 186
PRIMARY KEY (SQL), 1061
  NOT NULL, 1060–1061
primitive type, 211
printf() (C), 6–7, 98, 796
println() (Java), 13, 265
PrintObj.cc, 53
PrintObj.java, 52
print representation (Java), 52
printStackTrace() (Java), 410
PrintStream (Java), 13, 265
PrintWriter (Java), 265, 1034
priority-preemptive scheduling, 964
priority queue, a C++ container class, 149, 151,
  176
PriorityQueueOps.cc, 177
priority of a thread, 964
private
  base (C++), 618
  class derivation (C++), 664
  for access control, 34, 38, 71, 595
PrivateConstructor.cc, 420
Promo.cc, 224
promotion, 224
property-string (UML), 595–596
protected
  base (C++), 618
  class derivation (C++), 664
  for access control, 71, 595
  inheritance, 665
  prototype (C++), 346
  pseudorandom, 460
  pseudorandom numbers, 56
pthread_attr_init() (POSIX), 1003
pthread_attr_t (POSIX), 1002–1003
PTHREAD_CANCELLED (POSIX), 1007
pthread_cancel() (POSIX), 1013
pthread_cond_broadcast() (POSIX), 1014
pthread_cond_destroy() (POSIX), 1014
pthread_cond_init() (POSIX), 1013
pthread_cond_signal() (POSIX), 1014
pthread_cond_timedwait() (POSIX), 1014
pthread_cond_wait() (POSIX), 1014
PTHREAD_CREATE_DETACHED (POSIX), 1003
PTHREAD_CREATE_JOINABLE (POSIX), 1003
pthread_create() (POSIX), 1002
pthread_detach() (POSIX), 1004
pthread_exit() (POSIX), 1007
PTHREAD_EXPLICIT_SCHED (POSIX), 1004
PTHREAD_INHERIT_SCHED (POSIX), 1004
pthread_mutexattr_getkind_np() (POSIX), 1011
pthread_mutexattr_setkind_np (POSIX), 1011
pthread_mutex_destroy() (POSIX), 1013
PTHREAD_MUTEX_FAST_NP (POSIX), 1010
pthread_mutex_lock() (POSIX), 1010, 1014
PTHREAD_MUTEX_RECURSIVE_NP (POSIX), 1011
pthread_mutex_unlock() (POSIX), 1010, 1014
pthreads.h (POSIX), 1002
public
  base (C++), 618
  class, 11
  class derivation (C++), 617
    for access control, 34, 38, 71, 595
pull-down menu, 881
pure virtual (C++), 74–75, 643, 660, 727
push() (C++), 174–175
push_back() (C++), 153–154, 174
push_front() (C++), 168
put() (C++), 21, 247
putback() (C++), 246
putc() (C), 19, 21
put position (C++), 240
Q
qapplication.h (Qt), 831–832, 834, 836
QApplication (Qt), 795, 801, 859
  for default initializations, 795
QBrush (Qt), 920
QButton (Qt), 795
QColor (Qt), 921
QDialog (Qt), 795, 836
qdialog.h (Qt), 832, 836
QDns (Qt), 1038
QFileDialog (Qt), 795
QFrame (Qt), 795
QFtp (Qt), 1038
QGrid (Qt), 830, 834
qgrid.h (Qt), 834
QGridLayout (Qt), 830, 835, 960
QGridLayoutTest.cc, 836
QGridTest.cc, 834
QGroupBox (Qt), 795
QBox (Qt), 795, 830
qbbox.h (Qt), 831
QBoxLayout (Qt), 830
QHBoxLayoutTest.cc, 832
QBoxLayoutTest.cc, 831
QLabel (Qt), 795
QLayout (Qt), 795, 830
qlayout.h (Qt), 832, 836, 864
QLineEdit (Qt), 836
QMainWidget (Qt), 802
QMatrix (Qt), 919
QMimeData (Qt), 919
QMultiLineEdit (Qt), 886
QMutex (Qt), 1019
QNetworkProtocol (Qt), 1038
Q_OBJECT (Qt), 862, 870
QObject (Qt), 795, 859, 870
qpainter.h (Qt), 864
QPainter (Qt), 918, 921
QPaintEvent (Qt), 918
QPoint (Qt), 924
QPopupMenu (Qt), 888
qpushbutton.h (Qt), 831–832, 834, 836
QPushButton (Qt), 804, 836, 960
QServerSocket (Qt), 1038, 1044
qsizepolicy.h (Qt), 867
QSizePolicy (Qt), 866
QSocket (Qt), 795, 1038
qsort()
  (C), 116
  (C++), 199, 375
Qsort.cc, 117
QStatusBar (Qt), 795
QString (Qt), 1038
Qt
  a class, 921
  a GUI toolkit, 790–791, 793, 795
QThread (Qt), 1018
QT_THREAD_SUPPORT (Qt), 1019
quantum, 967
queue
  a C++ container class, 149, 151, 175
  a C++ header file, 8
Query (Mysql++), 1075–1076
query() (Mysql++), 1076
QueueOps.cc, 175
quick-sort, 116, 197, 315
INDEX

Q_UINT16 (Qt), 1038
QUIT (SQL), 1067
quit(), (Qt), 804, 859
QUrlOperator (Qt), 1038
QVBox (Qt), 795, 830, 834
QStringLayout (Qt), 830, 834
QWaitCondition (Qt), 1022
QWidget (Qt), 795, 919, 923
QWidget.h (Qt), 864

R
racecondition, 1006, 967
Random (Java), 460
random() (Java), 55–56, 460
RandomAccessFile (Java), 259, 266, 273
RandomAccessIterator (C++), 566–568
range
checking (Java), 134
violation (Java), 135
rangeCheck() (C++), 529
rbegin() (C++), 565
RCP, 1028
read() (Java), 17–18, 24, 27, 884
Reader (Java), 259, 1029, 1033
read for binary I/O (C++), 247
ReadIntFromFile.java, 271
readInt() (Java), 271
readLine() (Java), 205, 1029
readObject() (Java), 280
read-only collection (Java), 200
read-only iterator (Java), 1077
readString(), 18
ReadStringFromFile.java, 273
readUTF() (Java), 273
readyRead() (Qt), 1038, 1046
realizing a component (AWT/Swing), 797, 1000
REAL (SQL), 1061
realtime scheduling, 1004
recursive mutex (POSIX), 1011
red-black tree, 191
reference, 38, 329
(C++), 292
initialization, 330
_handler thread (Java), 965
Reference.cc, 338
Reference2.cc, 339
ReferenceClassType.cc, 331
reflexive association (UML), 593
registering
a callback, 844
a listener (AWT/Swing), 847
image with tracker (AWT/Swing), 915
reinterpret_cast (C++), 234, 781
reliable connection-based stream protocol, 1028
Remote Call Procedure, 1028
remove()
(C++), 170
Java), 183, 186, 190, 206
removeAll()
(AWT/Swing), 943
for emptying a Java container, 183
removeElement() (Java), 196
remove_if() (C++), 206
removeUpdate() (AWT/Swing), 854
render() (C++), 565
RenderGraphics.c, 928
RenderGraphics.cc, 921
rendering, 903
repaint()
(AWT/Swing), 904, 910, 941, 1000
(Qt), 919
repeated inheritance (C++), 729, 751
reset(), 920
RESULT_QUERY (SQL), 1076
resize() (C++), 124, 153, 157
resizing behavior, 825
responsibility (UML), 592
restart() (Java), 941
restrictions on overriding function
(C++), 651
(Java), 673
Result (Mysql++), 1075–1076
result() (Java), 1071
resurrecting unreferenced objects, 449
result command (SQL), 1062
return-type (UML), 596
revalidate() (AWT/Swing), 943, 1000
reverse() (C++), 567
reverseIterator (C++), 568
right_attach (GNOME/GTK+), 840
right bitwise shift operator, 16
rigid area (AWT/Swing), 815
rlogin, 1028
Robot.cc, 427
Robot.java, 434
role
(UML), 588, 592
based class hierarchy, 768
playing class, 721, 768
INDEX

RolePlayers.cc, 771
root pane (AWT/Swing), 797
rotate() (Qt), 920
rotating a shape, 910
RotatingRect.java, 911
rounding-modes for floating-point, 227
round
  -robin scheduling, 1004, 964
  -toward-zero mode, 227
RTTI (C++), 234, 646, 690, 700, 780–781
run()
  (Java), 966
  (Qt), 1018
Runnable (Java), 969
runnable state, 967, 970
running state, 970
RuntimeException (Java), 408
run-time type identification (C++), 234, 646, 690, 780
  700
S
safe_mysqld, 1058
SansSerif, 908
SCALE_DEFAULT (AWT/Swing), 916
SCALE_FAST (AWT/Swing), 916
scale() (Qt), 920
SCALE_REPLICATE (AWT/Swing), 916
SCALE_SMOOTH (AWT/Swing), 916
scaling an image, 916
scanf() (C), 15
SCHED_FIFO (POSIX), 1004
SCHED_OTHER (POSIX), 1004
schedparam (POSIX), 1004
schedpolicy (POSIX), 1004
SCHED RR (POSIX), 1004
scheduling
  a thread, 964, 967, 970
  policy, 1004
  priority, 1004
schema, 1073
scientif c, a stream manipulator (C++), 254
scope, 54, 61, 283
  (C++), 303
  (Java), 305
block
  (C++), 303
  (Java), 305
fi le (C++), 303
for thread scheduling contention (POSIX), 1004
function (C++), 303
function-prototype (C++), 303
namespace (C++), 303
of a global identifi er (C++), 304
operator (C++), 36, 61, 89, 304, 619
for nested type, 89
ScopeTest.java, 306
Scrollable (AWT/Swing), 940, 942
scrollbar, 873, 940, 942
search
  for array element (Java), 315
  for character (Java), 139
  for substring (C++), 120
  for substring (Java), 139
second (C++), 179
security, applet, 936, 956
seek() (Java), 266–267
seekp() (C++), 240
seekg() (C++), 240
SELECT (SQL), 1060, 1062, 1064
SelfRef.cc, 436
self-reference
  (C++), 435
  (Java), 439
sequence
  container
    (C++), 150, 173, 565
    (Java), 183
  adapter (C++), 173
diagram (UML), 597
sequential I/O, 266
Serializable (Java), 75–76, 274, 279
serial version UID (Java), 279
server, 1027
  -based database, 1057
ServerSocket (Java), 1031
Set, a Java container class, 182–183, 189
set
  a C++ container class, 149, 180
  a C++ header fi le, 8
  functions (SQL), 1069
  (Java), 182–183, 189
setBorder() (AWT/Swing), 810, 813, 819
setCoalesce() (Java), 941
setColor() (AWT/Swing), 905, 908
setMaximumSize()
  (AWT/Swing), 815
  (Qt), 833
setMinimumSize()
  (AWT/Swing), 815
setMnemonic() (AWT/Swing), 850
SetOps.cc, 180
SetOps.java, 190
setPen() (Qt), 921
setPrecision(), a stream manipulator (C++), 252
setPreferredSize() (AWT/Swing), 815, 937
setSize() (AWT/Swing), 800, 937
setSizePolicy() (Qt), 837, 866
setSocket() (Qt), 967
setSpacing() (Qt), 831
setToolTipText() (AWT/Swing), 849
setWorldMatrix() (Qt), 920
short, 213, 220
short-circuit evaluation (C++), 543
show() (AWT/Swing), 1000
showDocument() (Java), 956
showpoint, a stream manipulator (C++), 254
shuffle, 150, 166
shuffle() (Java), 316
side effect, 347
SIGNALL (Qt), 859
signal, 870
(C++), 804
emitted by button, 808
signaling object, 1013
signal_name (GNOME/GTK+), 880
signals and slots (Qt), 844
SignalSlotLCD.cc, 860
signature function, 369
signed char (C++), 213–214
int (C++), 213, 221
Silly.cc, 376
Simple Mail Transfer Protocol, 1028
SimulatedOO.c, 98
simulated oo in C, 2, 790–791
Singleton.cc, 417
Singleton (Java), 200
Singleton.java, 416
size() (C++), 124, 175, 180
(Java), 190
sizeHint() (Qt), 833, 836–837
sizeof, 6–7, 98, 117, 125
sizePolicy() (Qt), 837, 866
smallInt (C++), 110, 120
sizeType (C++), 116, 119, 122
Sketch.c, 933
Sketch.cc, 924
smoothing, free-form, 923, 932
skipws, a stream manipulator (C++), 255
Sleepws.cc, 256
sleep state, 964, 970
SlideShowApplet.html, 946
SlideShowApplet (Java), 940
SlideShowApplet.java, 943
SLOT (Qt), 859
slot (Qt), 804, 844, 870, 887
SMALLINT (SQL), 1061
SmallInt (C++), 529
SmallIntWithIncrementDecrement.cc, 531
smart pointer (C++), 519
SmartPtr.h, 527
SmartPtrInitial.cc, 523
SmartPtrWithOwnership.cc, 525
SMTP network service, 1028
society of objects, 1
Socket (Java), 1028–1029, 1038
Socket, 1027–1028, 1038
number, 1027–1028
SortedLine (Qt), 921
sort() (C++), 116, 162, 171, 182, 206, 375, 538, 567, 692
(Java), 55, 137, 145, 185–186, 205, 315
pointer types (C++), 542
stable, 117
SortedMap, a Java container class, 184, 191
SortedSet, a Java container class, 183
SortHeap (C++), 567
SortPointerType.cc, 542
SortTiming.java, 55
SortWithFunctor.cc, 539
SortWithLessThan.cc, 540
SOUTH (AWT/Swing), 810, 827
SOUTHEAST (AWT/Swing), 827
SOUTHWEST (AWT/Swing), 827
SpecialInt.cc, 437
SpecialInt.java, 440
specialization, 769
especific cation perspective (UML), 592
specificity, 369
splicing one list into another
(C++), 171
(Java), 185–186
SQL, 1055, 1057, 1059
stream, a C++ header file, 8, 257
stable sorting, 117, 197, 315
Stack, a Java container class, 184
stack, 299
INDEX

a C++ container class, 149, 151, 174
a C++ header file, 8
Java), 184
stacking order, 842
StackOps.cc, 174
StackOverflowError (Java), 408
standard
conversion, 224–225
error stream, 239, 21, 23
input stream, 239
namespace (C++), 9
output, 8
stream, 23, 239
StandardConvert.cc, 225
Standard Template Library (C++), 8–9
start()
for applets (AWT/Swing), 903, 937, 991
for QThread (Qt), 1018
for Thread (Java), 966
for Timer (Java), 941
startsWith() (Java), 141
state
diagram (UML), 604
of input stream object (C++), 23
transition (UML), 606
state() (Qt), 1039
statechart diagram (UML), 588, 606
concurrent, 608
Statement (Java), 1070
static, 19, 595, 690
class member
(C++), 422
Java), 430
const (C++), 422
initialization, 422
destruction (C++), 429
final (Java), 430, 690
initialization, 430
for continuous storage (C++), 362
for file scope (C++), 367
initialization (C++), 429
member, 81
initialization (C++), 423, 426
initialization (Java), 433
nested class (Java), 89, 847
result set (Mysql++), 1077
variable (C++), 361
binding (C++), 647
Static1.cc, 422
Static1.java, 430
Static2.cc, 423
Static2.java, 431
StaticBinding.cc, 647
static_cast (C++), 117, 232, 618, 781
StaticStorage.cc, 362
status message in sequence diagram (UML), 598
stdafx.h (C), 110
stderr, standard error stream (C), 19, 21
stdexcept, a C++ header file, 8–9
stdin, standard input stream (C), 14
stdio.h, a C header file, 21
stdlib.h, a C header file, 20
std, standard namespace (C++), 9, 22, 386
stereotype (UML), 596
STL (C++), 8–9
stop()
for applets (AWT/Swing), 937
for Timer (Java), 941
store() (Mysql++), 1076
storage allocation
String (Java), 132
strcat() (C), 110
strcncpy() (C), 109, 563
strcpy() (C), 98, 110, 507
stream buffering
(C++), 258
Java), 264
function (C++), 252
hierarchy
(C++), 237
Java), 259
manipulator (C++), 252
streambuf (C++), 258
a header file, 8
_strerror() (C), 1003
stretch (Qt), 833
String (Java), 18
String
Java), 129
storage allocation, 132
string
(C), 108
(C++), 33, 113
Java), 129
a C++ header file, 8–9, 113, 116
constant, 108
constructor
(C++), 507
Java), 131
joining of
(C), 110
(C++), 111
Java), 138
literal, 108, 211
::npos (C++), 120
stream, 257
StringBuffer (Java), 38, 129
StringCharIndexing.cc, 114
StringFind.cc, 121
StringFind.java, 140
string.h, a C header file, 107, 109
StringIndexOutOfBoundsException (Java), 134
StringInsert.java, 141
StringLiteralUniqueness.java, 131
StringSize.cc, 125
StringSizeOf.cc, 127
StringSort.java, 137
StringTokenizer (Java), 26, 192, 205
strlen() (C), 109, 507
struct (C), 98
struct (C++), 336
access privilege, 337
tag, 336
StructInit.cc, 337
Structured Query Language, 1055, 1057
StyledDocument (AWT/Swing), 854
subclass, 2, 31
(C++), 42
copy assignment operator, 617
copy constructor, 617
operator overloading, 617
(Java), 47
subscript operator (C++), 150
substr() (C++), 122
substring() (Java), 141
superclass, 31
(C++), 42
(Java), 47
super() (Java), 670
superstate (UML), 608
suppressing name mangling (C++), 386
swap() (C++), 129
Swap.cc, 356
Swap.java, 355
SwapWithPointer.cc, 357
Swing (Java), 790–791
SwingUtilities (AWT/Swing), 993, 1000
switch, 305
sync() (C++), 258
SynchedFileIO.java, 981
SynchedSwaps.c, 1011
SynchedSwaps.cc, 1019
SynchedSwaps.java, 979
synchronization
bar (UML), 611
(C++), 1019
(Java), 979
pthreads (POSIX), 1010
synchronized
collection (Java), 200
container (Java), 185
synchronizedList() (Java), 200
synchronous method invocation, 601
System (Java), 13, 56, 957
system thread group (Java), 965

T
tab character, 215
TableLayoutPanelTest.c, 841
TCP, 1028
server socket, 1038
socket, 1038
tellg() (C++), 240, 248
tellp() (C++), 240
telnet, 1027, 1053
daemon, 1027
telnetd, 1027
template
class, 547
class (C++), 83
declaration (C++)
general syntax, 563
non-type parameter, 563
type parameter, 563
parameter, 555
parameter (C++)
default value, 564
list, 563
specialization (C++), 560
TemplateX.cc, 83
templatization, 549
templatized
class, 550, 568
function, 550
program, 548
terminate() (C++), 393
TermIO.c, 14
TermIO.cc, 15
TermIO.java, 17
TextFilePosition2.cc, 241
TextFilePosition.cc, 240
Text, 1059
TextArea (AWT/Swing), 882, 884
text string, drawing of, 919
TFTP, 1028
TFTP network service, 1028
this, for self-reference
(C++), 435, 453
(Java), 439
this() (Java), 421, 669
Thread (Java), 1018, 941, 966, 968
thread
group
tree (Java), 965
interference, 971
priority, 942, 964
safety, 1000
scheduling
first-in first-out, 1004
non-realtime, 1004
INDEX

priority-preemptive, 964
round-robin, 1004, 964
state
  blocked, 970
  born, 967, 970
  runnable, 967, 970
  running, 970
  sleeping, 970
  waiting, 970
synchronization
  (C++), 1019
  (Java), 979
(PPOSIX)), 1010
ThreadBasic.c, 1005
ThreadBasic.java, 966
ThreadBasicWithJoin.c, 1007
ThreadBasicWithJoin.java, 968
ThreadBasicWithRunnable.java, 969
ThreadGroup (Java), 965
Throwable, 965
Throwables (Java), 1001
throw clause (C++), 389
throw clause (Java), 404
throws clause (Java), 399
time order in a sequence diagram (UML), 601
timer, 965
threads
  timeslicing of, 964, 967, 972
TIME (SQL), 1061
  WITH TIME ZONE, 1061
TIMESTAMP (SQL), 1061
  WITH TIME ZONE, 1061
toArray () (Java), 183, 201, 316
token, 192, 211
toolbar, 798, 897
Toolkit (AWT/Swing), 794, 915
top() (C++), 174
top_attach (GNOME/GDK+), 840
top_level
  containers, 793, 795–796
  window, 796, 804
toString() (Java), 52
tower of GUI API’s, 792
transient
  Java, 281
transition, statechart diagram (UML), 606
transitivity, 605
translate()
  (AWT/Swing), 905
  (Qt), 920
translating Unicode into UTF-8, 270
Transmission Control Protocol, 1028
transparent pixels, 793
TreeMap, a Java container class, 184, 191
TreeSet, a Java container class, 189
trigger (UML), 612
trimToSize() (Java), 197
Trivial File Transfer Protocol, 1028
trunc (C++), 239
try–catch
  (C++), 389
  (Java), 18, 24, 401
TryCatch.cc, 390
TryCatch.java, 400
two’s complement representation, 230, 236
  type, 114
    conversion
      explicit, 223
      explicit (C++), 232, 234
      explicit (Java), 234
      for primitive types, 223
      implicit, 223
      implicit (C++), 224, 232
      implicit (Java), 228
typedef (C++), 61, 65–67, 69, 84, 88, 113, 169,
  237
typeid() (C++), 786
typeinfo, a C++ header file, 8–9
type
  inheritance (C++), 665
  parameter, 555
    constraining of (Java), 579
template declaration of (C++), 563
type
  typedef (C++), 61, 65–67, 69, 84, 88, 113, 169,
  237
typeid() (C++), 786

U
UDP, 1028
ugetc() (C), 14
UML, 3, 588
unary operator, 504
unchecked exception (Java), 399, 408
unexpect( ) (C++), 393
unget() (C++), 246
ungetc() (C), 15
Unicode, 134, 136, 213, 218, 237, 1029
Unified Modeling Language, 3, 588
UNIQUE (SQL), 1062
unique() (C++), 171
Unix, 791–792, 794–795
  emulation in Windows, 12
UnknownHostException (Java), 1029
unmodifiableList() (Java), 200
unnamed namespace (C++), 69, 304
unsigned
  char (C++), 213–214
  int, 110, 116
  int (C++), 213
  long int, 110
UnsupportedEncodingException (Java), 134
UnsyncedFileIO.java, 976
UnsyncedSwaps.c, 1008
UnsyncedSwaps.java, 972
upcasting (C++), 628–629
UPDATE (SQL), 1060, 1062, 1066
update()
(AWT/Swing), 904
(Qt), 919
URL, 939, 943, 953, 1028, 1038
(Java), 956
usage patterns for exception handling
(C++), 391
(Java), 402
US-ASCII, 133
USE (SQL), 1063
use case (UML), 588, 608
  diagram, 588
  extend relationship, 590
  generalization, 590
  include relationship, 590
User1.cc, 35
User2.cc, 35
User3.cc, 36
UserConv.cc, 534
UserConvConstructor.cc, 535
UserConvExplicit.cc, 536
User Datagram Protocol, 1028
User.java, 39
using (C++)
  declaration, 64, 303
  directive, 9, 22, 61, 64, 303
UTF-8, 133
  encoding, 270
UTF-16, 133
UTF-16BE, 133
UTF-16LE, 133–134
utility, a C++ header file, 8, 176
V
tvalarray
  a C++ container class, 150, 152
  a C++ header file, 8
VARCHAR(n) (SQL), 1061
variable
  default initialization, 283
    (C++), 286
    (Java), 293
Vector, a Java container class, 184, 194
  vector
    (C++), 567
    (Java), 184, 194
  a C++ container class, 148, 150, 152, 174
  a C++ header file, 8, 156
  initialization (C++), 157, 166
  migration, 149
VectorBasic.cc, 155
VectorForClassType.cc, 164
VectorFrontBackResize.cc, 157
VectorInitArray.cc, 167
VectorInsertEraseSort.cc, 160
VectorListOps.java, 196
VectorOps.java, 195
VIRTUAL (AWT/Swing), 826
viewport view, 816
virtual
  base (C++), 237, 731, 737, 752
  destructor (C++), 655, 727, 781
  function (C++), 44, 617, 630, 651, 918
  pure (C++), 74–75, 643, 727
  table (C++), 648
    pointer, 649
    terminal, 791
VirtualBase.cc, 732
VirtualBaseAssign.cc, 740
VirtualBaseCopyConstruct.cc, 737
VirtualDestructor.cc, 655
VirtualPrint1.cc, 641
VirtualPrint2.cc, 644
visibility, 43, 47, 595
  private, 595
  protected, 595
  public, 595
visual representation, 587
VSPACE, 938
vtbl (C++), 648
vtpr (C++), 649
VtprConcealed.cc, 649
W
wait
  list, 984
  state, 964, 970
wait()
  (C++), 1018
  (Java), 983–984
waitForAll() (AWT/Swing), 915
waitForID() (AWT/Swing), 915
wait–notify (Java), 983
wait–signal (POSIX), 1014
wait–wake (Qt), 1022
wakeAll() (Qt), 1022
wakeOne() (Qt), 1022
weightx (AWT/Swing), 826
weighty (AWT/Swing), 826
WEST (AWT/Swing), 810, 827
WHERE (SQL), 1062
white-space characters, 15–16, 212
widening java, 228
widening primitive conversion (Java), 228
widening type conversion (Java), 371
widget, 795
Win32, 793
  API, 791–792
WINDOW_ACTIVATED (AWT/Swing), 852
WindowAdapter (AWT/Swing), 680, 850
WINDOW_CLOSED (AWT/Swing), 852
windowClosing() (AWT/Swing), 850
INDEX

WindowClosing (AWT/Swing), 852
WindowClosing() (AWT/Swing), 852
WindowDeactivated (AWT/Swing), 852
WindowDeiconified (AWT/Swing), 852
WindowEvent (AWT/Swing), 852
WindowIconified (AWT/Swing), 852
WindowListener (AWT/Swing), 847
window manager, 842
WindowOpened (AWT/Swing), 852
Windows platform, 794
WindowWithButtons2.java, 850
WindowWithButtons.java, 848
WindowWithHelloButton.c, 871
WindowWithMenu.c, 898
WindowWithMenu.cc, 889
WindowWithMenu.h (Qt), 889
WindowWithMenu.java, 884
wistream (C++), 258
wostream (C++), 258
wrapper class (Java), 185, 194
WriteBinaryIntsToFile.cc, 250
writeBytes() (Java), 269
writeChars() (Java), 264, 270
writeDouble() (Java), 264
writeFloat() (Java), 264
write for binary I/O (C++), 247
writeInt() (Java), 264
WriteIntToFile.java, 261
write() (Java), 24, 262–263, 269
writeObject() (Java), 280
Writer (Java), 259, 1029, 1033
WriteStringToFile.java, 267
writeUTF() (Java), 270
wxWindows (C++), 791

x

Xlib, 791–792
Xt, 791
X window system, 790
Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that

Your essential comparative approach to learning C++ and Java

Programming with Objects: A Comparative Presentation of Object-Oriented Programming with C++ and Java, a comparative presentation of object-oriented programming with two of the most popular programming languages of today, teaches vital skills and techniques for the Internet age. Based on highly successful courses taught by the author, this book answers the need for a comprehensive educational program on the subject of object-oriented programming. A Comparative Presentation of Object-Oriented Programming with C++ and Java. (A new book authored by Avi Kak and published by John-Wiley) (The book was released by the publisher on March 21, 2003.) The second printing came out in mid 2005.