What's Inside

Foreword 1

Introduction 5

Prerequisites 6
Learning Java 6
Goals 7
Online documentation.......... 8
Chapters 9
Exercises 15
Multimedia CD ROM.......... 15
Source code 16
Coding standards 17
Java versions 18
Seminars and mentoring.... 19
Errors 19
Note on the cover design.... 20
Acknowledgements 21

1: Introduction to objects 25

The progress of abstraction 26
An object has an interface... 28
The hidden implementation 30
Reusing the implementation 31
Inheritance: reusing the interface........... 32
Overriding base-class functionality 33
Is-a vs. is-like-a relationships ... 34
Interchangeable objects with polymorphism............ 35
Dynamic binding...................... 37
Abstract base classes and interfaces.......................... 37

Object landscapes and lifetimes 38
Collections and iterators 39
The singly-rooted hierarchy 41
Collection libraries and support for easy collection use............... 42
The housekeeping dilemma: who should clean up? 44
Exception handling: dealing with errors 46
Multithreading.................. 47
Persistence......................... 48
Java and the Internet 48
What is the Web? 48
Client-side programming 51
Server-side programming 57
A separate arena: applications .. 58
Analysis and Design 58
Staying on course 59
Phase 0: Let's make a plan........ 59
Phase 1: What are we making? 60
Phase 2: How will we build it?................ 61
Phase 3: Let's build it!............... 62
Phase 4: Iteration...................... 63
Plans pay off 64
Java vs. C++? 65

2: Everything is an object 67
You manipulate objects with handles 68
You must create all the objects 69
Where storage lives 69
Special case: primitive types 70
Arrays in Java 72
You never need to destroy an object............... 73
Scoping.................................... 73
Scope of objects 74
Creating new data types: class................ 75
Fields and methods 75
Methods, arguments and return values 77
The argument list 78
Building a Java program 79
Name visibility 80
Using other components.......... 80
The **static** keyword 81
Your first Java program 83
Comments and embedded documentation 86
Comment documentation 87
Syntax...................................... 88
Embedded HTML....................... 89
@see: referring to other classes 89
Class documentation tags 90
Variable documentation tags 90
Method documentation tags 91
Documentation example 92
Coding style 93
Summary 93
Exercises 93

3: Controlling program flow 95
Using Java operators......... 95
Precedence 96
Assignment............................... 96
Mathematical operators.......... 100
Auto increment and decrement 102
Relational operators................ 103
Logical operators 105
Bitwise operators 108
Shift operators........................ 109
Ternary if-else operator 113
The comma operator 114
String operator +.................. 114
Common pitfalls when using operators....................... 115
Casting operators 115
Java has no "sizeof"................ 119
Precedence revisited 119
A compendium of operators... 119
Execution control 130
true and false 130
if-else 131
Iteration 132
do-while................................... 133
for.. 133
break and continue 135
switch 141
Summary 146
Exercises 146

4: Initialization and cleanup 147
Guaranteed initialization with the constructor 148
Method overloading 150
Distinguishing overloaded methods 152
Overloading with primitives... 153
Overloading on return values.......................... 157
Default constructors 158
The **this** keyword.................. 159
Cleanup: finalization and garbage collection...... 162
What is **finalize**( ) for? 163
You must perform cleanup 164
Member initialization....... 168
Specifying initialization.......... 170
Constructor initialization 171
Array initialization 179
Multidimensional arrays........ 184
Summary 186
Exercises 187

5: Hiding the implementation 189
package: the library unit 190
Creating unique package names....................... 193
A custom tool library 196
Using imports to change behavior 199
Package caveat 201
Java access specifiers 202
"Friendly"............................... 202
public: interface access........... 203
private: you can't touch that! 205
protected: "sort of friendly" 206
Interface and implementation 208
Class access 209
Summary 212
Exercises 214

6: Reusing classes 217
Composition syntax 218
Inheritance syntax............ 221
Initializing the base class 224
Combining composition and inheritance................ 226
Guaranteeing proper cleanup 228
Name hiding........................... 231
Choosing composition vs. inheritance 233
protected........................ 234
Incremental development.................... 235
Upcasting 236
Why "upcasting"? 237
The **final** keyword........... 238
Final data 238
Final methods......................... 243
Final classes............................ 243
Final caution 244
Initialization and class loading 245
Initialization with inheritance 246
Summary 248
Exercises 248

7: Polymorphism 251

- Upcasting 252
  - Why upcast? 253
- The twist 255
  - Method call binding 255
  - Producing the right behavior 256
  - Extensibility 259
- Overriding vs. overloading 263
- Abstract classes and methods 264
- Interfaces 268
  - "Multiple inheritance" in Java 272
  - Extending an interface with inheritance 274
  - Grouping constants 275
  - Initializing fields in interfaces 277
- Inner classes 278
  - Inner classes and upcasting 280
  - Inner classes in methods and scopes 282
  - The link to the outer class 288
  - **static** inner classes 291
  - Referring to the outer class object 293
  - Inheriting from inner classes 294
  - Can inner classes be overridden? 295
  - Inner class identifiers 297
  - Why inner classes: control frameworks 298
- Constructors and polymorphism 306
  - Order of constructor calls 306
  - Inheritance and **finalize( )** 308
  - Behavior of polymorphic methods inside constructors 312
- Designing with inheritance 315
  - Pure inheritance vs. extension 316
  - Downcasting and run-time type identification 318
- Summary 320
- Exercises 321

8: Holding your objects 323

- Arrays 324
  - Arrays are first-class objects 325
  - Returning an array 329
- Collections 331
  - Disadvantage: unknown type 331
- Enumerators (iterators) 337
- Types of collections 341
  - **Vector** 341
  - **BitSet** 342
  - **Stack** 344
  - **Hashtable** 345
  - Enumerators revisited 353
- Sorting 354
- The generic collection library 360
- The new collections 361
  - Using **Collections** 366
  - Using **Lists** 369
  - Using **Sets** 374
  - Using **Maps** 376
  - Choosing an implementation 379
  - Unsupported operations 388
  - Sorting and searching 391
  - Utilities 397
- Summary 400
- Exercises 401

9: Error handling with exceptions 403

- Basic exceptions 405
  - Exception arguments 406
- Catching an exception 406
  - The **try** block 407
  - Exception handlers 407
  - The exception specification 409
  - Catching any exception 410
  - Rethrowing an exception 412
- Standard Java exceptions 415
  - The special case of **RuntimeException** 416
- Creating your own exceptions 418
- Exception restrictions 421
- Performing cleanup with finally 425
  - What's **finally** for? 427
  - Pitfall: the lost exception 429
- Constructors 431
- Exception matching 435
  - Exception guidelines 436
- Summary 437
- Exercises 437

10: The Java IO system 439

- Input and output 440
  - Types of **InputStream** 440
  - Types of **OutputStream** 442
- Adding attributes and useful interfaces 444
  - Reading from an **InputStream** with **FilterInputStream** 445
  - Writing to an **OutputStream** with **FilterOutputStream** 446
- Off by itself: RandomAccessFile 448
- The **File** class 449
  - A directory lister 449
  - Checking for and creating directories 454
- Typical uses of IO streams 456
  - Input streams 460
  - Output streams 462
  - Shorthand for file manipulation 463
  - Reading from standard input 465
  - Piped streams 466
- **StreamTokenizer** 467
  - **StringTokenizer** 470
- Java 1.1 IO streams 473
  - Sources and sinks of data 474
  - Modifying stream behavior 474
  - Unchanged Classes 476
  - An example 476
  - Redirecting standard IO 480
- Compression 482
  - Simple compression with GZIP 483
  - Multi-file storage with Zip 484
  - The Java archive (jar) utility 487
- Object serialization 489
  - Finding the class 493
  - Controlling serialization 495
  - Using persistence 504
- Summary 512
- Exercises 513

11: Run-time type identification 515

- The need for RTTI 516
  - The **Class** object 518
  - Checking before a cast 521
- RTTI syntax 528
- Reflection: run-time class information 531
  - A class method extractor 532
- Summary 537
- Exercises 538

12: Passing and returning objects 541

- Passing handles around 542
  - Aliasing 543
- Making local copies 545
  - Pass by value 546
  - Cloning objects 546
  - Adding cloneability to a class 548

- Successful cloning 550
- The effect of **Object.clone( )** ... 552
- Cloning a composed object 555
- A deep copy with **Vector** 557
- Deep copy via serialization 559
- Adding cloneability further down a hierarchy 561
- Why this strange design? 562

Controlling cloneability 563

- The copy-constructor 568

Read-only classes 572

- Creating read-only classes 574
- The drawback to immutability 575
- Immutable **Strings** 577
- The **String** and **StringBuffer** classes 581
- **Strings** are special 584

Summary 584

Exercises 586

13: Creating windows and applets 587

Why use the AWT? 589

The basic applet 590

- Testing applets 593
- A more graphical example 594
- Demonstrating the framework methods 595

Making a button 596

Capturing an event 596

Text fields 600

Text areas 601

Labels 602

Check boxes 604

Radio buttons 606

Drop-down lists 607

List boxes 609

- **handleEvent( )** 610

Controlling layout 613

- **FlowLayout** 613
- **BorderLayout** 614
- **GridLayout** 615
- **CardLayout** 616
- **GridBagLayout** 618

Alternatives to **action** 619

Applet restrictions 625

- Applet advantages 626

Windowed applications 627

- Menus 627
- Dialog boxes 631

The new AWT 638

- The new event model 639
- Event and listener types 641
- Making windows and applets with the Java 1.1 AWT 646
- Revisiting the earlier examples 650
- Binding events dynamically 670
- Separating business logic from UI logic 672
- Recommended coding approaches 675

Java 1.1 UI APIs 692

- Desktop colors 692
- Printing 693
- The clipboard 701

Visual programming and Beans 704

- What is a Bean? 705
- Extracting **BeanInfo** with the **Introspector** 708
- A more sophisticated Bean 714
- Packaging a Bean 718
- More complex Bean support ... 719
- More to Beans 720

Introduction to Swing 721

- Benefits of Swing 722
- Easy conversion 722
- A display framework 724
- Tool tips 725
- Borders 725
- Buttons 726
- Button groups 728
- Icons 730
- Menus 732
- Popup menus 737
- List boxes and combo boxes 738
- Sliders and progress bars 739
- Trees 740
- Tables 743
- Tabbed Panes 745
- The Swing message box 747
- More to Swing 747

Summary 748

Exercises 749

14: Multiple threads 751

Responsive user interfaces 752

- Inheriting from **Thread** 755
- Threading for a responsive interface 757
- Combining the thread with the main class 761
- Making many threads 764
- Daemon threads 767

Sharing limited resources 769

- Improperly accessing resources 770
- How Java shares resources..... 775
- Java Beans revisited 780

Blocking 785

- Becoming blocked 786
- Deadlock 797

Priorities 801

- Thread groups 806

Runnable revisited 814

- Too many threads 817

Summary 820

Exercises 822

15: Network programming 825

Identifying a machine 826

- Servers and clients 828
- Port: a unique place within the machine 829

Sockets 830

- A simple server and client....... 831

Serving multiple clients 837

Datagrams 842

A Web application 850

- The server application 851
- The **NameSender** applet 857
- Problems with this approach 863

Connecting Java to CGI 864

- Encoding data for CGI 865
- The applet 867
- The CGI program in C++ 872
- What about POST? 882

Connecting to databases with JDBC 887

- Getting the example to work .. 890
- A GUI version of the lookup program 894
- Why the JDBC API seems so complex 897

Remote methods 897

- Remote interfaces 898
- Implementing the remote interface 899
- Creating stubs and skeletons .. 902
- Using the remote object 903
- Alternatives to RMI 904

Summary 905

Exercises 905

16: Design patterns 907

The pattern concept 908

- The singleton 909
- Classifying patterns 911

The observer pattern 911

Simulating the trash recycler 915

Improving the design 919

- "Make more objects" 919
- A pattern for prototyping creation 922

Abstracting usage 932

Multiple dispatching 936

- Implementing the double dispatch 937

The "visitor" pattern 944

RTTI considered harmful? 952
Summary 955
Exercises 956

17: Projects 957

Text processing 957
Extracting code listings 958
Checking capitalization style 974
A method lookup tool 983
Complexity theory 988
Summary 995
Exercises 996

A: Using non-Java code 997

The Java Native Interface 998
Calling a native method 999
Accessing JNI functions: The **JNIEnv** argument 1002
Passing and using Java objects 1003
JNI and Java exceptions 1006
JNI and threading 1007
Using a pre-existing code base 1007
The Microsoft way 1007
J/Direct 1008
The **@dll.import** directive 1009
The com.ms.win32 package 1011
Marshaling 1012
Writing callback functions.... 1013
Other J/Direct features 1015
Raw Native Interface (RNI) 1015
RNI Summary 1017
Java/COM integration 1017
COM Fundamentals 1019
MS Java/COM Integration.... 1022
Developing COM servers in Java 1023
Developing COM clients in Java 1025
ActiveX/Beans integration 1027
A note about native methods and applets 1027
CORBA 1028
CORBA Fundamentals 1028
An example 1030
Java Applets and CORBA 1035
CORBA vs. RMI 1036
Summary 1036

B: Comparing C++ and Java 1039

C: Java programming guidelines 1051

D: Performance 1057

Basic approach 1057
Locating the bottleneck... 1058
1. Install your own instrumentation 1058
2. JDK profiling [2] 1059
3. Special tools 1059
Tips for measuring performance 1059
Speedup techniques 1060
Generic approaches 1060
Language dependent approaches 1060
Specific situations 1061
References 1064
Performance tools 1064
Web sites 1064
Articles 1064
Java specific books 1065
General books 1065

E: A bit about garbage collection 1067

F: Recommended reading 1073

Index 1075