

CONTENTS

FOREWORD *xxii*

Chapter 1: THE WORLD OF JAVA 1

- THE NAME 2
- THE INTERNET 3
- PORTABILITY 3
- ACCESSIBILITY 4
- ROLES OF JAVA 4
- HOUSE OF JAVA 4
- JAVA BYTECODE 7
- JAVA: THE LANGUAGE 8
- THE INTERNET, HTML, AND JAVA APPLETS 10
 - HTML as an Internet User-Interface Language 14
 - Java Applets 16
- JAVA APPLICATIONS 18
- JAVASCRIPT 18
- JAVASERVER PAGES 20

CONTENTS

ENTERPRISE JAVA BEANS	20
JAVA VERSUS COBOL	21
SUMMARY	22
REFERENCES	23
<i>Chapter 2: JAVA FLOW OF CONTROL</i>	25
JAVA'S OUTER SKIN	26
COMPILING	26
Compiling COBOL	26
Compiling Java	27
FLOW OF CONTROL	28
COBOL Flow of Control	28
Java's Flow of Control	30
Calling all COBOL Programs	44
Calling All Java Methods	45
Java Overload! Java Overload!	46
Prototyping Java Methods—Not!	47
Defining Java Methods	48
CODE REUSE	49
COBOL Code Reuse	49
Java Code Reuse through Packages	50
STATIC DATA	57
COBOL Static Data Items	57
Java Static Variables	58
Java Static Methods	60
NAME SCOPING AND QUALIFICATION	62
COBOL Scoping and Qualification	62
Java Scoping and Qualification	63
NAME ACCESS RIGHTS	63
COBOL Access Rights	64
Java Access Rights	64
Getters and Setters	66
CLASS CONSTRUCTORS IN JAVA	66
CALLING ALL PROGRAMS	69
Calling a COBOL Program: the Mainline	69
Calling a Java Program: the “main” Method	69
Writing to the Command Line in COBOL	73
Writing to the Command Line in Java	74
SUMMARY	77

Chapter 3: JAVA SYNTAX ····· 79

 I WANT TO BE FREE ····· 79

 COMMENTS ····· 80

 JavaDoc Comments ····· 82

 VARIABLE NAMING ····· 88

 COMPARING ANATOMIES ····· 89

 Anatomy of a COBOL Program ····· 89

 Anatomy of a Java Class ····· 90

 STATEMENTS ····· 92

 Expression Statement ····· 92

 Block Statement ····· 93

 Empty Statement ····· 95

 Labeled Statement ····· 96

 EXPRESSIONS AND OPERATORS ····· 98

 Computational Expressions ····· 99

 Fetch Expressions ····· 115

 OTHER MATHEMATICAL FUNCTIONS ····· 116

 SUMMARY ····· 119

Chapter 4: STRUCTURED OPERATIONS AND STATEMENTS ····· 121

 ALTERNATE PATHS: DECISION STRUCTURES ····· 124

 If, Then What? Else, of Course ····· 124

 Too Many IFs? Time to Switch! ····· 128

 The Conditional Operator, ?: ····· 132

 LOOP STRUCTURES—THEY GET AROUND ····· 133

 Going “for” a Loop: The for Statement ····· 134

 Looping for a While ····· 143

 Looping Until Done ····· 147

 TRANSFER STRUCTURES ····· 150

 If You Continue, We Will Need a Break ····· 150

 Go to Where? ····· 156

 Return to Caller ····· 156

 SUMMARY ····· 159

Chapter 5: DATA TYPES AND VARIABLES ····· 161

 REVIEW OF COBOL DATA CATEGORIES ····· 162

 INTRODUCTION TO JAVA TYPES ····· 164

 DATA DECLARATION ····· 168

 Defining Data Items in COBOL ····· 168

CONTENTS

Defining Variables in Java	168
VARIABLE SCOPE, LIFETIME, AND EXTERNAL ACCESS	172
Variable Scope	172
Variable Lifetime	174
Variable External Access	175
Java Variables: An Example	176
LITERALS BY DATA TYPE	178
Figurative Constants	181
NUMERIC DATA TYPE RANGES	181
Float Versus Decimal	183
BOOLEAN DATA TYPE	187
CASTING AND NUMERIC CONVERSIONS	188
JAVA DATA-TYPE CLASS WRAPPERS	192
SUMMARY	195
<i>Chapter 6: ARRAYS, VECTORS, AND HASHTABLES</i>	197
ARRAYS	198
Declaring Arrays	200
Creating or Instantiating Arrays	202
Compile-Time Initialization of Arrays	203
Runtime Initialization of Arrays	204
Initializing Multidimensional Arrays	206
Arrays Are Objects	211
Unbalanced Arrays	213
ARRAYS OF OBJECTS	214
HELPER METHODS FOR ARRAYS	217
VECTORS	221
HASHTABLES	229
PARAMETER PASSING	233
USING ARRAYS INSTEAD OF PRIMITIVES TO SIMULATE PASS-BY-REFERENCE	237
SUMMARY	239
<i>Chapter 7: STRING MANIPULATION</i>	241
JAVA STRING BASICS AND PITFALLS	242
STRINGS IN COBOL	244
STRING MANIPULATION	247
Concatenation	247
Substring	249
Searching for a Substring	254

Trimming Blanks	256
Determining String Lengths	258
Translating Characters	259
Translating Strings	260
Translating Case	263
Class Condition	264
Parsing Strings	266
PERFORMANCE CONSIDERATIONS: THE STRINGBUFFER CLASS	266
Trimming Leading and Trailing Blanks in Java	271
TOKENIZING STRINGS: THE STRINGTOKENIZER CLASS	274
STRING CLASS: MORE METHODS	276
SUMMARY	277
<i>Chapter 8: DATE AND TIME MANIPULATION</i>	279
COBOL DATE AND TIME REVIEW	279
COBOL DATE AND TIME ENHANCEMENTS	281
OS/390 COBOL Date and Time Enhancements	283
ILE COBOL on OS/400 Date and Time Enhancements	285
Finally	288
JAVA DATE AND TIME SUPPORT	289
Raw Time	290
If You Have the Time, I Need a Date	293
Comparing Dates	296
DATE AND TIME MANIPULATION	297
Date/Time Math and Duration	297
Extracting Parts of a Date or Time	300
FORMATTING DATE AND TIME	303
Formatting Date Data Items	303
Formatting Time Data Items	306
The Default Formatter	308
LOOK BEFORE YOU LEAP	311
TIME ZONES	312
SUMMARY	314
<i>Chapter 9: AN OBJECT ORIENTATION</i>	315
OO TERMINOLOGY	316
Encapsulation	317
Inheritance	319
Deep Inheritance	326
Polymorphism	327

CONTENTS

INHERITING MORE	334
Protected Access	335
Constructor Access	336
Preventing Inheritance	336
Enforcing Inheritance	337
Constructors and Inheritance	345
Multiple Inheritance	346
Java-Supplied Interface Examples	358
O2O—OBJECT TO OBJECT: CASTING OBJECTS	363
Why Cast?	364
The Rules of Object Casting	365
COMMON METHODS TO OVERRIDE	365
THE OO WAY OF LIFE	366
In Search of Objects	367
Reuse Considerations	371
Other OO Terms	372
OO Design Tips	373
SUMMARY	375
REFERENCES	376
<i>Chapter 10: EXCEPTIONS</i>	377
COBOL EXCEPTION AND ERROR-HANDLING: THE OS/390 LE AND OS/400 ILE EXCEPTION MODEL	377
EXCEPTIONS IN JAVA	379
Exception Objects in Java	379
Sending Java Exceptions—The throw Statement	386
Monitoring for Java Exceptions—The try/catch Blocks	388
Who Throws What	391
SUMMARY	393
<i>Chapter 11: THREADS</i>	395
SYNCHRONOUS VERSUS ASYNCHRONOUS	395
THREADS VERSUS JOBS	396
CALLING A METHOD ASYNCHRONOUSLY: SPAWNING THREADS	398
An Example without Using Threads	398
Spawning Option 1: Extend Thread	401
Spawning Option 2: Implement Runnable	405
STOPPING A THREAD	407
STOPPING MULTIPLE THREADS—THREAD GROUPS	413
ENDING PROGRAMS WITH JAVA RUNNING THREADS	415
Daemon Threads	415

Java's Own Threads	415
THINKING ASYNCHRONOUS—DANGER LURKS	416
Multi-Threaded Danger—An Inventory Example	418
Threaded Application Design Alternatives	425
MORE ON SYNCHRONIZED THREADS	425
Synchronizing Producer/Consumer Relationships	427
Deadly Embrace	429
THREAD PRIORITIES	430
LOOSE THREADS	431
TIMING THREADED APPLICATIONS	432
SUMMARY	435
<i>Chapter 12: USER INTERFACE</i>	<i>437</i>
USER INTERFACES IN JAVA: OPTIONS, OPTIONS, OPTIONS	438
Java GUIs	439
What About WYSIWYG Tools?	440
AN OVERVIEW OF JAVA AWT	440
Basic Components: Buttons, Lists, Text, and More	441
Containers: Frames, Dialogs, and Panels	445
Adding Components to Containers: Layout Managers	449
Advanced Components: Menu Bars, Pull-Downs, and Popups	467
Processing User Input—Events	473
A Small AWT Example: A To-Do List	484
Error Checking	490
Error Messages	492
Applets Versus Applications	494
Graphics	499
Locales	499
A BRIEF SWING	504
Swinging Harder and Faster	509
A Small Swing Example: A To-Do List	511
Swing Wrap	512
SUMMARY	513
<i>Chapter 13: DATABASE ACCESS</i>	<i>515</i>
ACCESSING DATA FROM COBOL	515
Data on OS/390	516
Data on OS/400	516
Data and COBOL	517
COBOL RECORD-LEVEL DATABASE ACCESS	519
COBOL Commitment Control on OS/390	520

CONTENTS

COBOL Commitment Control on OS/400	521
ACCESSING DATA WITH SQL	522
Interactive SQL Versus Embedded SQL	524
Static SQL Versus Dynamic SQL	527
Embedded SQL Versus SQL CLI APIs	528
Local SQL Versus Distributed SQL	529
SQL Views and Indexes	530
SQL Languages: DDL Versus DML	530
WHY THE INTRODUCTION TO SQL?	532
ACCESSING DATA WITH JAVA	532
JDBC Drivers for DB2/390	534
JDBC Drivers for DB2/400	535
Java JDBC Programming	536
A Database Query Example	559
More on Data Types	561
Null-Value Considerations in JDBC	562
Commitment Control in JDBC	562
Retrieving Database Metadata in JDBC	563
Good Stuff: Stored Procedures, Triggers, and RI	568
Stored Procedure Calls in JDBC	569
WHAT'S NEW IN JDBC 2.0	572
Data Type Enhancements	572
ResultSet Enhancements	573
Something New: Batch Updates	574
JDBC 2.0 and DB2	575
Optional JDBC 2.0 Enhancements	575
EMBEDDED SQL IN JAVA: SQLJ	576
Writing Embedded SQL in Java	577
Preparing and Compiling Embedded SQL in Java	581
SUMMARY	582
REFERENCES	583
<i>Chapter 14: MORE JAVA</i>	585
INNER CLASSES	586
ANONYMOUS INNER CLASSES	588
STATIC INITIALIZERS	590
JAVA NATIVE INTERFACE (JNI)	591
JAVA REMOTE METHOD INVOCATION (RMI)	593
JAVA OBJECT SERIALIZATION	597
JAVA-SUPPLIED PACKAGES	601
COMMON "HOW DO I?" QUESTIONS	604

How Do I Determine the Working Directory?	604
How Do I Change the Working Directory?	605
How Do I List Files and Directories?	606
How Do I List Drives?	611
How Do I Create a Directory?	612
How Do I Create or Open a Flat File and Write to It?	613
How Do I Read a (Flat) File?	614
How Do I Call Another Program or Command?	615
How Do I Save and Remember User Profile Information?	618
JAVA BEANS	620
Java Bean Features	621
Java Bean Introspection	622
Thinking Java Beans	623
Java Bean Example	624
Java Bean Grinds	625
WEBSHERE: OF SERVLETS, JAVASERVER	
PAGES, ENTERPRISE JAVA BEANS, AND MORE	626
Servlets	627
JavaServer Pages	634
Enterprise Java Beans (EJBs)	642
XML	644
VISUALAGE FOR JAVA	647
Versions of VisualAge for Java	648
The VisualAge for Java Repository	649
Creating, Editing, Running, and Debugging Inside VisualAge for Java	650
The VisualAge for Java Visual Composition Editor	652
Other Functions in VisualAge for Java	654
SUMMARY	656
<i>Appendix A: COBOL FOR OS/390 FLOW OF CONTROL</i>	<i>657</i>
IBM COBOL FOR OS/390 & VM REVIEW	657
Typical COBOL Application Structure	658
New Packaging Option: Dynamic Link Libraries (DLLs)	660
COBOL FOR OS/390 FLOW OF CONTROL	663
Distributing Your COBOL DLLs	664
SUMMARY	665
<i>Appendix B: ILE COBOL FLOW OF CONTROL</i>	<i>667</i>
ILE COBOL FOR OS/400 REVIEW	667
TRADITIONAL COBOL/400 APPLICATIONS	669
NEW ILE COBOL APPLICATIONS	670

CONTENTS

ILE COBOL Program Objects and Modules	671
ILE Activation Groups	674
COMPILING COBOL	674
ILE COBOL FLOW OF CONTROL	675
Running a COBOL Program	675
Distributing Your COBOL Service Programs	676
SUMMARY	677
<i>Appendix C: OBJECT-ORIENTED COBOL ON OS/390</i>	679
JAVA OBJECT MODEL REVIEW	680
OVERVIEW OF OO COBOL	682
DEFINING CLASSES IN OO COBOL	684
Identification Division	684
Environment Division	684
Data Division	685
Procedure Division	685
Example	686
DEFINING METHODS IN OO COBOL	686
Identification Division	686
Environment Division	687
Data Division	687
Procedure Division	687
USING CLASSES IN OO COBOL	688
Instantiating OO COBOL Objects	689
Invoking OO COBOL Methods	690
Freeing or Deallocating OO COBOL Objects	690
SUMMARY COMPARISON TO JAVA	691
<i>Appendix D: USING JAVA ON OS/390</i>	693
INSTALLING THE COMPONENTS	695
THE OS/390 JAVA DEVELOPMENT KIT	696
Using the Hierarchical File System and IShell Interpreter	697
Working with Java on OS/390	699
Setting CLASSPATH in OMVS Shell	701
Java Remote AWT	703
OF GATEWAYS, CONNECTORS, PACKAGES, AND MORE	704
Common Connector Framework (CCF)	706
Enterprise Access Builders	708
A Little More on JRIO	713

Creating New CICS Transactions with Java	714
CICS and CORBA	722
VISUALAGE FOR JAVA ENTERPRISE EDITION FOR OS/390	725
<i>Appendix E: USING JAVA ON OS/400</i>	<i>727</i>
INSTALLING THE COMPONENTS	728
THE AS/400 DEVELOPER KIT FOR JAVA	728
Using the Integrated File System and QShell Interpreter	729
Working with Java on OS/400	732
Setting CLASSPATH in QShell	733
Using CL Commands for Java, and the Java Transformer	735
Java Remote AWT	736
AS/400 TOOLBOX FOR JAVA	737
Getting the AS/400 Toolbox for Java Classes	741
Using the AS/400 Toolbox for Java Classes	742
CommandCall	743
ProgramCall	743
DataQueue	747
Record-Level Database Access	753
Other AS/400 Toolbox for Java Functionality	758
VISUALAGE FOR JAVA ENTERPRISE TOOLKIT FOR AS/400	758
Common Connector Framework in VisualAge for Java	760
<i>Appendix F: SOFTWARE LOADING INSTRUCTIONS</i>	<i>761</i>
TO-DO LIST SWING GUI EXAMPLE	762
VISUALAGE FOR JAVA ON YOUR CD	762
<i>Bibliography</i>	<i>765</i>
WEB SITES ON JAVA AND RELATED TOPICS	765
BOOKS ON JAVA AND RELATED TOPICS	768
<i>Index</i>	<i>771</i>
<i>About the Authors</i>	<i>815</i>

