

CONTENTS

<i>Chapter 1: THE WORLD OF JAVA</i>	1
WHAT IS JAVA?	3
WHAT ABOUT THAT OO THING?	3
WHY THE HYPE?	4
The Name	4
The Internet	5
Portability	6
Accessibility	6
THE ROLES OF JAVA	6
THE HOUSE OF JAVA	7
JAVA BYTECODE	10
JAVA: THE LANGUAGE	11
THE INTERNET, HTML, AND JAVA APPLETS	13
HTML as an Internet user-interface language	17
Java as an Internet user-interface language	19
JAVA APPLICATIONS	20
JAVASCRIPT	21
JAVASERVER PAGES	22

ENTERPRISE JAVABEANS	23
JAVA VERSUS RPG	23
SUMMARY	24
References	25
<i>Chapter 2: THE JAVA ONION</i>	27
RPG IV VERSUS RPG III APPLICATIONS	28
THE OUTER SKIN: RPG VERSUS JAVA	32
RPG’s outer skin	32
Java’s outer skin	32
Compiling RPG	32
Compiling Java	33
About RPG IV procedures	34
RPG IV flow of control	35
About Java methods	36
Java’s flow of control	37
COMPARING ANATOMIES	38
RPG syntax	39
Prototyping RPG procedures	40
Defining RPG procedures	41
Calling RPG procedures	43
JAVA SYNTAX	44
Java class syntax	44
Java variable syntax	46
Java method syntax	47
An example of a Java class	48
Getter and setter methods in Java	50
ALLOCATING MEMORY IN RPG IV	50
USING JAVA CLASSES	53
Accessing methods	55
Accessing variables	58
This is special	59
Java overload! Java overload!	61
STATIC MEMBERS	61
RPG static fields	62
Java static variables	63
Java static methods	65
CONSOLE-ATION	69
Calling a Java program: The “main” method	69

Command-line parameters in RPG	71
Command-line parameters in Java	72
Writing to the command line in RPG	72
Writing to the command line in Java	74
Class constructors in Java	78
CODE REUSE AND DISTRIBUTION	82
Distributing reusable RPG code as service programs	82
Java code reuse through packages	84
JAVA ACCESSOR RIGHTS	91
SUMMARY	93
<i>Chapter 3: JAVA'S LANGUAGE AND SYNTAX</i>	<i>95</i>
I WANT TO BE FREE! THE JAVA BASICS	95
COMMENTS	97
Java comments	98
JavaDoc comments	99
VARIABLE NAMING	105
The naming syntax for fields and variables	106
Keywords	106
COMPARING ANATOMIES	107
The anatomy of an RPG program	107
The anatomy of a Java Class	108
STATEMENTS	110
The expression statement	110
The block statement	111
The empty statement	113
The labeled statement	114
EXPRESSIONS AND OPERATORS	116
Computational expressions	117
Fetch expressions	136
OTHER MATHEMATICAL FUNCTIONS	137
SUMMARY	142
<i>Chapter 4: STRUCTURED OPERATIONS AND STATEMENTS</i>	<i>143</i>
ALTERNATE PATHS: DECISION STRUCTURES	146
If, then what? Else, of course	146
Too many IFs? Time to switch!	151
The conditional operator, ?:	154
Loop structures—They get around	155

Going “for” a loop—Just “do” it	156
A FOR loop in RPG—Yes, in RPG	164
Looping for a while	165
Looping until done	170
TRANSFER STRUCTURES	171
If you continue, I will need a break	171
Go to where?	175
Return to caller	176
SUMMARY	178
<i>Chapter 5: DATA TYPES AND VARIABLES</i>	<i>181</i>
A REVIEW OF RPG TYPES	182
AN INTRODUCTION TO JAVA TYPES	186
DATA DECLARATION	189
Defining fields in RPG	190
Defining variables in Java	195
VARIABLE SCOPE, LIFETIME, AND EXTERNAL ACCESS	199
Variable scope	199
Variable lifetime	200
Variable external access	201
Variables: An example	202
LITERALS BY DATA TYPE	205
NUMERIC DATA TYPE RANGES	207
FLOAT VERSUS PACKED	209
THE BOOLEAN DATA TYPE	214
CASTING AND NUMERIC CONVERSIONS	215
JAVA DATA-TYPE CLASS WRAPPERS	220
SUMMARY	223
<i>Chapter 6: ARRAYS AND VECTORS</i>	<i>225</i>
OBJECT REFERENCE OPERATORS	227
ARRAYS	228
Declaring arrays	230
Creating or instantiating arrays	232
Compile-time initialization of arrays	233
Runtime initialization of arrays	236
Initializing multi-dimensional arrays	238
Arrays are objects	241
Pre-runtime arrays in RPG	243

Unbalanced arrays ·····	248
Arrays of objects ·····	249
Helper methods for arrays ·····	252
VECTORS ·····	256
The Registration class ·····	260
The AttendeeList Class ·····	261
The Attendee class ·····	263
Simulating vectors in RPG ·····	264
HASHTABLES ·····	271
PARAMETER PASSING ·····	278
Parameter passing in RPG IV ·····	278
Using arrays instead of primitives for simulated pass-by-reference ·····	281
SUMMARY ·····	283
<i>Chapter 7: STRING MANIPULATION</i> ·····	285
JAVA STRING BASICS AND PITFALLS ·····	286
STRINGS IN RPG ·····	288
STRING MANIPULATION ·····	292
Concatenation ·····	292
Substrings ·····	294
Searching for a substring ·····	296
Trimming blanks ·····	298
Determining string lengths ·····	299
Translating characters ·····	300
Translating case ·····	303
Checking for characters ·····	305
PERFORMANCE CONSIDERATIONS: THE STRINGBUFFER CLASS ·····	308
IMPLEMENTING TRIMR AND TRIML IN JAVA ·····	312
TOKENIZING STRINGS: THE STRINGTOKENIZER CLASS ·····	316
THE STRING CLASS: MORE METHODS ·····	317
SUMMARY ·····	318
<i>Chapter 8: DATE AND TIME MANIPULATION</i> ·····	321
RPG IV DATE AND TIME ENHANCEMENTS ·····	322
JAVA DATE AND TIME SUPPORT ·····	322
RAW TIME ·····	323
GET ME A DATE, AND GIVE ME THE TIME! ·····	327
DATE AND TIME MANIPULATION ·····	330
Date/time math and duration ·····	330

Comparing dates · · · · ·	334
Extracting parts of a date or time · · · · ·	335
FORMATTING DATE FIELDS · · · · ·	338
FORMATTING TIME FIELDS · · · · ·	344
THE DEFAULT FORMATTER · · · · ·	349
TESTING DATE AND TIME · · · · ·	351
LOOK BEFORE YOU LEAP · · · · ·	354
TIME ZONES · · · · ·	355
SUMMARY · · · · ·	356
<i>Chapter 9: AN OBJECT ORIENTATION</i> · · · · ·	359
THE THREE TENETS OF OO · · · · ·	361
Encapsulation · · · · ·	361
Inheritance · · · · ·	362
Polymorphism · · · · ·	372
INHERITING MORE OR LESS · · · · ·	379
Preventing inheritance · · · · ·	379
Enforcing inheritance · · · · ·	380
THE FACTORY METHOD-DESIGN PATTERN · · · · ·	382
INTERFACES · · · · ·	387
Interfaces versus inheritance · · · · ·	398
Interfaces are callbacks · · · · ·	400
Interfaces and constants · · · · ·	401
Java-supplied interface examples · · · · ·	401
O2O—OBJECT TO OBJECT: CASTING OBJECTS · · · · ·	409
COMMON METHODS TO OVERRIDE · · · · ·	411
THE OO WAY OF LIFE · · · · ·	412
In search of objects · · · · ·	412
OO design tips · · · · ·	415
SUMMARY · · · · ·	418
REFERENCES · · · · ·	419
<i>Chapter 10: EXCEPTIONS</i> · · · · ·	421
SYSTEM I EXCEPTION MODEL OVERVIEW · · · · ·	421
The OPM exception model · · · · ·	422
The ILE exception model · · · · ·	423
RPG III exception-handling · · · · ·	423
ILE RPG (RPG IV) exception-handling · · · · ·	424
New MONITOR operation in RPG V5R1 · · · · ·	425

EXCEPTIONS IN JAVA	426
The Java exception model at a glance	427
Exception objects in Java	428
Searching Java documentation for classes	431
Sending Java exceptions with the throw operator	433
Monitoring for Java exceptions with try/catch blocks	437
SUMMARY	443
<i>Chapter 11: THREADS</i>	<i>445</i>
SYNCHRONOUS VERSUS ASYNCHRONOUS	445
THREADS VERSUS JOBS	446
CALLING A METHOD ASYNCHRONOUSLY: SPAWNING THREADS	448
Calling without using threads	448
Using threads, option 1: The extends thread	451
Using threads, option 2: implements Runnable	454
STOPPING A THREAD	457
Stopping multiple threads: The ThreadGroups class	462
Ending programs with Java running threads	464
DAEMON THREADS	465
THINKING ASYNCHRONOUS—DANGER LURKS...	465
The role of threads in a complex multi-user application	466
How single objects can have multiple threads of execution	467
Multithreaded danger: An inventory example	467
The solution: Synchronizing asynchronous execution	472
Fine-grained synchronization	474
Threaded application design alternatives	475
MORE ON THE SYNCHRONIZED KEYWORD	476
Synchronizing producer/consumer relationships with wait/notify	478
Deadly embrace	480
THREAD PRIORITIES	481
LOOSE THREADS	482
TIMING THREADED APPLICATIONS	483
SUMMARY	485
<i>Chapter 12: USER INTERFACE</i>	<i>487</i>
USER INTERFACES IN RPG IV	487
USER INTERFACES IN JAVA: OPTIONS, OPTIONS, OPTIONS	488
JAVA GUI	490
AN OVERVIEW OF JAVA SWING	491

BASIC GUI COMPONENTS IN JAVA	494
Common inherited methods	496
Common non-inherited methods	501
Model view controller	508
Events	509
CONTAINERS: JFRAMES, JDIALOGS, AND JPANELS	515
Adding components to containers: Layout managers	519
Nesting layout managers: The JPanel class	531
A simple form class	536
EFFECTIVE USE OF JTEXTFIELDS	543
Component sizing and layout managers	543
Restricting input in a JTextField	544
MENU BARS AND POP-UPS	545
ADVANCED SWING FUNCTIONALITY	550
Default push button	550
Toolbars	550
Actions	551
Multiple-document interface	551
More components	551
More containers	552
Look and feel	552
Predefined dialogs	553
Message and simple input dialogs	556
APPLETS VERSUS APPLICATIONS	557
Applets and Swing	558
TRANSLATION-READY USING RESOURCE BUNDLES	562
Locales	562
Translated strings—Resource bundles	564
SUMMARY	566
<i>Chapter 13: DATABASE ACCESS</i>	<i>567</i>
DATABASE ACCESS IN RPG IV	567
RPG DATABASE ACCESS UNDER COMMITMENT CONTROL	569
ACCESSING DATA WITH SQL	569
Interactive SQL versus embedded SQL	572
Static SQL versus dynamic SQL	573
Dynamic SQL versus CLI APIs	574
Local versus distributed SQL	575
SQL terminology	576

SQL languages: DDL versus DML	577
What you need for SQL on the AS/400	579
WHY THE INTRODUCTION TO SQL?	579
ACCESSING DATA WITH JAVA	579
JDBC DRIVERS FOR DB2/400	582
JAVA JDBC PROGRAMMING	583
Steps 1 and 2: Registering and connecting	585
Step 3: Creating SQL statement objects	595
Step 4: Running the SQL statements	597
Step 5: Retrieving the information in result sets	599
Step 6: Close, close, close	617
More on data types	617
Null-value considerations in JDBC	619
Commitment control in JDBC	620
Retrieving database metadata in JDBC	621
THE GOOD STUFF: STORED PROCEDURES, TRIGGERS, AND RI	624
STORED PROCEDURE CALLS IN JDBC	625
WHAT'S NEW IN JDBC 2.0	627
Datatype enhancements	627
ResultSet enhancements	628
Something new: Batch updates	630
Optional JDBC 2.0 enhancements	630
WHAT'S NEW IN JDBC 3.0	631
EMBEDDED SQL IN JAVA: SQLJ	631
Writing embedded SQL in Java	633
Preparing and compiling embedded SQL in Java	636
SUMMARY	637
REFERENCES	637
<i>Chapter 14: MORE JAVA</i>	<i>639</i>
INNER CLASSES	640
ANONYMOUS INNER CLASSES	642
STATIC INITIALIZERS	644
JAVA NATIVE INTERFACE (JNI)	645
JAVA REMOTE METHOD INVOCATION (RMI)	647
JAVA OBJECT SERIALIZATION	652
J2SE 5.0 LANGUAGE CHANGES	656
Generics	656
Enhanced for Loop	657

Autoboxing/Unboxing	657
Enumerations	658
Variable Arguments List	659
Static Import	659
Annotations	659
JAVA-SUPPLIED PACKAGES	660
COMMON “HOW DO I?” QUESTIONS	664
How do I determine the working directory?	664
How do I change the working directory?	665
How do I list files and directories?	667
How do I list drives?	669
How do I create a directory?	670
How do I create or open a flat file and write to it?	670
How do I read a flat file?	672
How do I call another program or command?	673
How do I save and restore user-profile information?	676
JAVABEANS	678
Properties	678
Bound properties	681
Events	684
Serializing beans	687
The BeanInfo class	689
Beans and tools	690
More beans	692
NEXT STEPS: OF SERVLETS, JSPS, EJBS, AND MORE	692
SUMMARY	692
<i>APPENDIX A</i>	<i>695</i>
INSTALLING THE COMPONENTS	696
THE ISERIES IBM DEVELOPER KIT FOR JAVA	697
Using the Integrated File System and QShell Interpreter	697
Working with Java on OS/400	700
Setting your classpath in QShell	702
Using CL commands for Java and the Java Transformer	703
ISERIES IBM TOOLBOX FOR JAVA	705
Getting the iSeries IBM Toolbox for Java classes	710
Using the iSeries IBM Toolbox for Java classes	711
The CommandCall class	713
The ProgramCall class	714

PCML for easy calling of programs and service programs	723
The DataQueue class	727
Record-level database access	736
OTHER AS/400 TOOLBOX FOR JAVA FUNCTIONALITY	744
<i>APPENDIX B</i>	745
RPG CALLING JAVA	746
Mapping data types	754
Java exceptions when calling Java from RPG	756
Java Virtual Machine considerations	756
Classpath considerations	757
CALLING RPG PROCEDURES AS RPG NATIVE METHODS FROM JAVA	757
FINALLY	761
<i>Appendix C: OBTAINING THE CODE SAMPLES</i>	763
<i>ABOUT THE AUTHORS</i>	765
<i>INDEX</i>	767