
Table of Contents

Preface.....	vii
1. Introduction.....	1
Why Did They Need to Change Java Again?	1
What Is Functional Programming?	2
Example Domain	3
2. Lambda Expressions.....	5
Your First Lambda Expression	5
How to Spot a Lambda in a Haystack	6
Using Values	8
Functional Interfaces	9
Type Inference	11
Key Points	13
Exercises	14
3. Streams.....	17
From External Iteration to Internal Iteration	17
What's Actually Going On	20
Common Stream Operations	21
collect(toList())	22
map	22
filter	24
flatMap	25
max and min	26
A Common Pattern Appears	27
reduce	28
Putting Operations Together	30
Refactoring Legacy Code	31

Multiple Stream Calls	34
Higher-Order Functions	36
Good Use of Lambda Expressions	36
Key Points	37
Exercises	37
Advanced Exercises	39
4. Libraries.....	41
Using Lambda Expressions in Code	41
Primitives	42
Overload Resolution	45
@FunctionalInterface	47
Binary Interface Compatibility	47
Default Methods	48
Default Methods and Subclassing	49
Multiple Inheritance	52
The Three Rules	53
Tradeoffs	54
Static Methods on Interfaces	54
Optional	55
Key Points	56
Exercises	57
Open Exercises	58
5. Advanced Collections and Collectors.....	59
Method References	59
Element Ordering	60
Enter the Collector	62
Into Other Collections	62
To Values	63
Partitioning the Data	64
Grouping the Data	65
Strings	66
Composing Collectors	67
Refactoring and Custom Collectors	69
Reduction as a Collector	76
Collection Niceties	77
Key Points	78
Exercises	78
6. Data Parallelism.....	81
Parallelism Versus Concurrency	81

Why Is Parallelism Important?	83
Parallel Stream Operations	83
Simulations	85
Caveats	88
Performance	89
Parallel Array Operations	92
Key Points	94
Exercises	94
7. Testing, Debugging, and Refactoring.	97
Lambda Refactoring Candidates	97
In, Out, In, Out, Shake It All About	98
The Lonely Override	98
Behavioral Write Everything Twice	99
Unit Testing Lambda Expressions	102
Using Lambda Expressions in Test Doubles	105
Lazy Evaluation Versus Debugging	106
Logging and Printing	106
The Solution: peek	107
Midstream Breakpoints	107
Key Points	108
8. Design and Architectural Principles.	109
Lambda-Enabled Design Patterns	110
Command Pattern	110
Strategy Pattern	114
Observer Pattern	117
Template Method Pattern	119
Lambda-Enabled Domain-Specific Languages	123
A DSL in Java	124
How We Got There	125
Evaluation	127
Lambda-Enabled SOLID Principles	127
The Single Responsibility Principle	128
The Open/Closed Principle	130
The Dependency Inversion Principle	134
Further Reading	137
Key Points	137
9. Lambda-Enabled Concurrency.	139
Why Use Nonblocking I/O?	139
Callbacks	140

Message Passing Architectures	144
The Pyramid of Doom	145
Futures	147
Completable Futures	149
Reactive Programming	153
When and Where	155
Key Points	155
Exercises	156
10. Moving Forward.....	159
Index.....	161