
Table of Contents

Preface.....	vii
1. Introduction.....	1
Overview of Concepts in Functional Programming	1
First-Class Functions	2
Pure Functions	2
Recursion	2
Immutable Variables	2
Nonstrict Evaluation	2
Statements	2
Pattern Matching	2
Functional Programming and Concurrency	3
Conclusion	3
2. First-Class Functions.....	5
Introduction to XXY	5
Functions as Objects	7
Refactoring Using If-Else Structures	8
Refactoring Using Function Objects to Extract Fields	10
Anonymous Functions	16
Lambda Functions	16
Closures	18
Higher-Order Functions	20
Refactoring get Functions by Using Groovy	22
Conclusion	23
3. Pure Functions.....	25
Output Depends on Input	25
Purifying Our Functions	29

Side Effects	33
Conclusion	37
Making the Switch to Groovy	38
4. Immutable Variables.....	43
Mutability	43
Immutability	48
Conclusion	54
5. Recursion.....	55
An Introduction to Recursion	56
Recursion	59
Tail Recursion	61
Refactoring Our countEnabledCustomersWithNoEnabledContacts Function	62
Conclusion	64
Introducing Scala	65
6. Strict and Nonstrict Evaluations.....	67
Strict Evaluation	68
Nonstrict (Lazy) Evaluation	69
Laziness Can Create Problems	73
Conclusion	76
7. Statements.....	79
Taking the Plunge	80
Simple Statements	80
Block Statements	82
Everything Is a Statement	84
Conclusion	92
8. Pattern Matching.....	93
Simple Matches	93
Simple Patterns	95
Extracting Lists	97
Extracting Objects	99
Converting to Pattern Matches	101
Conclusion	103
9. Functional OOP.....	105
Static Encapsulation	105
Objects As Containers	107
Code as Data	109

Conclusion	111
10. Conclusion.....	113
From Imperative to Functional	113
Introduce Higher-Order Functions	113
Convert Existing Methods into Pure Functions	114
Convert Loops to Tail/Recursive-Tail Methods	114
Convert Mutable Variables into Immutable Variables	115
What Next?	115
New Design Patterns	115
Message Passing for Concurrency	115
The Option Pattern (Extension of Null Object Pattern)	116
Object to Singleton Method Purity	117
Putting It All Together	117
Conclusion	125
Index.....	127