Stephen G. Kochan

Updated for Xcode 4 and iOS

Programming in Objective-C

Third Edition

Developer's Library

Table of Contents

1 Introduction 1

What You Will Learn from This Book 2

How This Book Is Organized 3

Support 5

Acknowledgments 5

Part I The Objective-C 2.0 Language

2 Programming in Objective-C 7

Compiling and Running Programs 7

Using Xcode 8

Using Terminal 15

Explanation of Your First Program 17

Displaying the Values of Variables 21

Summary 23

3 Classes, Objects, and Methods 27

What Is an Object, Anyway? 27

Instances and Methods 28

An Objective-C Class for Working with Fractions 30

The `@interface` Section 32

Choosing Names 33

Instance Variables 35

Class and Instance Methods 35

The `@implementation` Section 37

The `program` Section 38

Accessing Instance Variables and Data Encapsulation 45

Summary 48

4 Data Types and Expressions 51

Data Types and Constants 51

Type `int` 51

Type `float` 52

Type `char` 52

Qualifiers: `long`, `long`, `long`, `short`, `unsigned`, and `signed` 53
Type `id` 54
Arithmetic Expressions 55
Operator Precedence 55
Integer Arithmetic and the Unary Minus Operator 58
The Modulus Operator 60
Integer and Floating-Point Conversions 61
The Type Cast Operator 62
Assignment Operators 63
A Calculator Class 64

5 Program Looping 69
The `for` Statement 70
Keyboard Input 76
Nested `for` Loops 78
`for` Loop Variants 80
The `while` Statement 81
The `do` Statement 85
The `break` Statement 87
The `continue` Statement 87
Summary 88

6 Making Decisions 91
The `if` Statement 91
The `if-else` Construct 95
Compound Relational Tests 98
Nested `if` Statements 101
The `else if` Construct 102
The `switch` Statement 111
Boolean Variables 114
The Conditional Operator 118

7 More on Classes 123
Separate Interface and Implementation Files 123
Synthesized Accessor Methods 128
Accessing Properties Using the Dot Operator 129

Multiple Arguments to Methods 130
Methods Without Argument Names 132
Operations on Fractions 133
Local Variables 135
Method Arguments 136
The `static` Keyword 137
The `self` Keyword 140
Allocating and Returning Objects from Methods 141
Extending Class Definitions and the Interface File 146

8 Inheritance 149
It All Begins at the Root 149
Finding the Right Method 153
Extension Through Inheritance: Adding New Methods 154
A Point Class and Memory Allocation 157
The `@class` Directive 159
Classes Owning Their Objects 163
Overriding Methods 167
Which Method Is Selected? 169
Overriding the `dealloc` Method and the Keyword `super` 171
Extension Through Inheritance: Adding New Instance Variables 173
Abstract Classes 175

9 Polymorphism, Dynamic Typing, and Dynamic Binding 179
Polymorphism: Same Name, Different Class 179
Dynamic Binding and the `id` Type 182
Compile Time Versus Runtime Checking 184
The `id` Data Type and Static Typing 185
Argument and Return Types with Dynamic Typing 186
Asking Questions About Classes 187
Exception Handling Using `@try` 191

10 More on Variables and Data Types 197

Initializing Objects 197

Scope Revisited 200

Directives for Controlling Instance Variable Scope 200

External Variables 201

Static Variables 203

Enumerated Data Types 205

The `typedef` Statement 208

Data Type Conversions 209

Conversion Rules 210

Bit Operators 211

The Bitwise AND Operator 213

The Bitwise Inclusive-OR Operator 214

The Bitwise Exclusive-OR Operator 214

The Ones Complement Operator 215

The Left Shift Operator 216

The Right Shift Operator 217

11 Categories and Protocols 219

Categories 219

Some Notes About Categories 224

Protocols and Delegation 225

Delegation 228

Informal Protocols 228

Composite Objects 229

12 The Preprocessor 233

The `#define` Statement 233

More Advanced Types of Definitions 235

The `#import` Statement 240

Conditional Compilation 241

The `#ifdef`, `#endif`, `#else`, and `#ifndef` Statements 241

The `#if` and `#elif` Preprocessor Statements 243

The `#undef` Statement 244

13 Underlying C Language Features 247

Arrays 248

Initializing Array Elements 250

Character Arrays 251

Multidimensional Arrays 252

Functions 254

Arguments and Local Variables 255

Returning Function Results 257

Functions, Methods, and Arrays 260

Blocks 261

Structures 265

Initializing Structures 268

Structures Within Structures 269

Additional Details About Structures 271

Don't Forget About Object-Oriented Programming! 273

Pointers 274

Pointers and Structures 277

Pointers, Methods, and Functions 279

Pointers and Arrays 280

Operations on Pointers 290

Pointers and Memory Addresses 291

Unions 292

They're Not Objects! 295

Miscellaneous Language Features 295

Compound Literals 295

The `goto` Statement 296

The null Statement 296

The Comma Operator 297

The `sizeof` Operator 297

Command-Line Arguments 298

How Things Work 300

Fact #1: Instance Variables are Stored in Structures 300

Fact #2: An Object Variable is Really a Pointer 301

Fact #3: Methods are Functions, and Message Expressions are Function Calls 301
Fact #4: The `id` Type is a Generic Pointer Type 302

Part II The Foundation Framework

14 Introduction to the Foundation Framework 305
Foundation Documentation 305

15 Numbers, Strings, and Collections 309
Number Objects 309
A Quick Look at the Autorelease Pool 311
String Objects 314
More on the `NSLog` Function 314
The `description` Method 315
Mutable Versus Immutable Objects 316
Mutable Strings 322
Where Are All Those Objects Going? 326
Array Objects 328
Making an Address Book 332
Sorting Arrays 350
Dictionary Objects 356
Enumerating a Dictionary 357
Set Objects 360
`NSIndexSet` 363

16 Working with Files 369
Managing Files and Directories: `NSFileManager` 370
Working with the `NSData` Class 374
Working with Directories 376
Enumerating the Contents of a Directory 378
Working with Paths: `NSPathUtilities.h` 380
Common Methods for Working with Paths 383
Copying Files and Using the `NSProcessInfo` Class 385
Basic File Operations: `NSFileHandle` 389
The `NSURL` Class 393
The `NSBundle` Class 394

17 Memory Management 397
The Autorelease Pool 397
Reference Counting 398
Reference Counting and Strings 401
Instance Variables 403
An Autorelease Example 409
Summary of Memory-Management Rules 410
More on the Event Loop and Memory Allocation 411
Finding Memory Leaks 413
Garbage Collection 413

18 Copying Objects 417
The `copy` and `mutableCopy` Methods 418
Shallow Versus Deep Copying 420
Implementing the `<NSCopying>` Protocol 422
Copying Objects in Setter and Getter Methods 425

19 Archiving 429
Archiving with XML Property Lists 429
Archiving with `NSKeyedArchiver` 431
Writing Encoding and Decoding Methods 433
Using `NSData` to Create Custom Archives 440
Using the Archiver to Copy Objects 443

Part III Cocoa, Cocoa Touch, and the iOS SDK

20 Introduction to Cocoa and Cocoa Touch 445
Framework Layers 445
Cocoa Touch 446

21 Writing iOS Applications 449
The iOS SDK 449
Your First iPhone Application 449
Creating a New iPhone Application Project 452
Entering Your Code 455
Designing the Interface 458

An iPhone Fraction Calculator 464
Starting the New `Fraction_Calculator` Project 465
Defining the View Controller 468
The `Fraction` Class 472
A `Calculator` Class That Deals with Fractions 475
Designing the UI 477
Summary 478

Appendixes

A Glossary 481

B Address Book Source Code 489

Index 495