

Stephen G. Kochan



Updated
for Xcode 4.5
and iOS 6

Programming in Objective-C

Fifth Edition

Developer's Library



Programming in Objective-C

Fifth Edition

Programming in Objective-C

Table of Contents

Contents

1 Introduction

What You Will Learn from This Book

How This Book Is Organized

Support

Acknowledgments

Preface to the Fifth Edition

I: The Objective-C Language

2 Programming in Objective-C

Compiling and Running Programs

Explanation of Your First Program

Displaying the Values of Variables

Summary

Exercises

3 Classes, Objects, and Methods

What Is an Object, Anyway?

Instances and Methods

An Objective-C Class for Working with Fractions

The @interface Section

The @implementation Section

The program Section

Accessing Instance Variables and Data Encapsulation

Summary

Table of Contents

Exercises

4 Data Types and Expressions

Data Types and Constants

Arithmetic Expressions

Assignment Operators

A Calculator Class

Exercises

5 Program Looping

The for Statement

The while Statement

The do Statement

The break Statement

The continue Statement

Summary

Exercises

6 Making Decisions

The if Statement

The switch Statement

Boolean Variables

The Conditional Operator

Exercises

7 More on Classes

Separate Interface and Implementation Files

Synthesized Accessor Methods

Accessing Properties Using the Dot Operator

Multiple Arguments to Methods

Local Variables

The self Keyword

Allocating and Returning Objects from Methods

Table of Contents

Exercises

8 Inheritance

It All Begins at the Root

Extension through Inheritance: Adding New Methods

Overriding Methods

Abstract Classes

Exercises

9 Polymorphism, Dynamic Typing, and Dynamic Binding

Polymorphism: Same Name, Different Class

Dynamic Binding and the id Type

Compile Time Versus Runtime Checking

The id Data Type and Static Typing

Asking Questions about Classes

Exception Handling Using @try

Exercises

10 More on Variables and Data Types

Initializing Objects

Scope Revisited

Enumerated Data Types

The typedef Statement

Data Type Conversions

Bit Operators

Exercises

11 Categories and Protocols

Categories

Class Extensions

Protocols and Delegation

Composite Objects

Exercises

Table of Contents

12 The Preprocessor

- The #define Statement

- The #import Statement

- Conditional Compilation

- Exercises

13 Underlying C Language Features

- Arrays

- Functions

- Blocks

- Structures

- Pointers

- They're Not Objects!

- Miscellaneous Language Features

- How Things Work

- Exercises

II: The Foundation Framework

14 Introduction to the Foundation Framework

- Foundation Documentation

15 Numbers, Strings, and Collections

- Number Objects

- String Objects

- Array Objects

- Dictionary Objects

- Set Objects

- Exercises

16 Working with Files

- Managing Files and Directories: NSFileManager

- Working with Paths: NSPathUtilities.h

Table of Contents

Basic File Operations: NSFileHandle

The NSURL Class

The NSBundle Class

Exercises

17 Memory Management and Automatic Reference Counting

Automatic Garbage Collection

Manual Reference Counting

The Event Loop and Memory Allocation

Summary of Manual Memory Management Rules

Automatic Reference Counting

Strong Variables

Weak Variables

@autoreleasepool Blocks

Method Names and Non-ARC Compiled Code

18 Copying Objects

The copy and mutableCopy Methods

Shallow Versus Deep Copying

Implementing the <NSCopying> Protocol

Copying Objects in Setter and Getter Methods

Exercises

19 Archiving

Archiving with XML Property Lists

Archiving with NSKeyedArchiver

Writing Encoding and Decoding Methods

Using NSData to Create Custom Archives

Using the Archiver to Copy Objects

Exercises

III: Cocoa, Cocoa Touch, and the iOS SDK

Table of Contents

20 Introduction to Cocoa and Cocoa Touch

Framework Layers

Cocoa Touch

21 Writing iOS Applications

The iOS SDK

Your First iPhone Application

An iPhone Fraction Calculator

Summary

Exercises

Appendixes

A: Glossary

A

B

C

D

E

F

G

I

L

M

N

O

P

R

S

U

X

Z

Table of Contents

B: Address Book Example Source Code

Index