An Introduction to Object-Oriented Programming

Timothy Budd
Oregon State University

Addison-Wesley Publishing Company
Reading, Massachusetts · Menlo Park, California · New York
Don Mills, Ontario · Wokingham, England · Amsterdam · Bonn
Sydney · Singapore · Tokyo · Madrid · San Juan · Milan · Paris

CONTENTS

Preface *xi*

Organization of the Book xii
Necessary Background xiv
Acknowledgments xiv
Obtaining the Source Code xv

1
Thinking Object-Oriented *1*

1.1 A New Paradigm 2
1.2 A Way of Viewing the World 3
Computation as Simulation 8
1.3 Coping with Complexity 10
Abstraction Mechanisms 11
1.4 Reusable Software 14
1.5 Summary 15
Further Reading 15

2
Responsibility-Driven Design *19*

2.1 CRC Cards 20
2.2 Discovering Classes 23
2.3 Designing from Scenarios 24
A Case Study — A Teller Machine 25
Data Managers 28
2.4 Discovering Inheritance 31
2.5 Common Design Flaws 33

3
Classes and Methods *37*

3.1 Encapsulation 37
3.2 Example: A Playing Card 38
3.3 Interface and Implementation 42

3.4 Classes and Methods 42
Classes and Methods in Object Pascal 43
Classes and Methods in Smalltalk 45
Classes and Methods in Objective-C 48
Classes and Methods in C++ 52

4
Messages, Instances, and Initialization 57
4.1 Message-Passing Syntax 57
Message Passing in Object Pascal 58
Message Passing in C++ 59
Message Passing in Smalltalk 60
Message Passing in Objective-C 61
4.2 Creation and Initialization 61
Stack Versus Heap Storage Allocation 61
Memory Recovery 62
Lifetimes of Values 63
Pointers 64
Immutable Creation 64
4.3 Mechanisms for Creation and Initialization 65
Creation and Initialization in C++ 65
Creation and Initialization in Object Pascal 68
Creation and Initialization in Smalltalk 70
Creation and Initialization in Objective-C 72

5
A Case Study: Eight Queens 75
5.1 The Eight Queens Puzzle 75
5.2 Eight Queens in Object Pascal 80
5.3 Eight Queens in C++ 81
5.4 Eight Queens in Objective-C 82
5.5 Eight Queens in Smalltalk 84

6
Inheritance 85
6.1 The Benefits of Inheritance 86
Software Reusability 86
Code Sharing 86
Consistency of Interface 87
Software Components 87
Rapid Prototyping 87

Polymorphism 87
Information Hiding 89
6.2 *The Cost of Inheritance 89*
Execution Speed 89
Program Size 89
Message-Passing Overhead 90
Program Complexity 90
6.3 *Heuristics for When to Subclass 90*
Specialization 91
Specification 92
Construction 92
Generalization 92
Extension 93
Limitation 94
Variance 95
Combination 95
6.4 *If It Ain't Broke 95*
6.5 *Tree Versus Forest 96*
6.6 *Composition Versus Construction 97*

7
Static and Dynamic Binding *103*
7.1 *Static and Dynamic Typing 103*
The Container Problem 105
7.2 *Static and Dynamic Method Binding 106*
Merits of Static Versus Dynamic Binding 107
7.3 *Binding in Object Pascal 109*
7.4 *Binding in Smalltalk 110*
7.5 *Binding in Objective-C 111*
7.6 *Binding in C++ 111*

8
Case Study: Mixed-Type Arithmetic *113*
8.1 *Coercive Generality 113*
8.2 *Double Dispatching 116*
8.3 *A Comparison 117*

9
Replacement and Refinement *121*
9.1 *Adding, Replacing, and Refining 121*
9.2 *Replacement 122*

Replacement in C++ 124
Replacement in Object Pascal 128
Replacement in Objective-C and Smalltalk 128
9.3 Refinement 128
Refinement in C++ 129
Refinement in Object Pascal 131
Refinement in Objective-C and Smalltalk 131

10
Case Study: Solitaire *135*
10.1 The class Card *135*
10.2 Inheritance and Overriding 136
10.3 Object-Oriented User Interfaces 144

11
Inheritance and Types *151*
11.1 Memory Layout 151
Minimum Static Space Allocation 153
Maximum Static Space Allocation 155
Dynamic Memory Allocation 156
11.2 Assignment 157
Assignment in C++ 158
Assignment in Object Pascal 160
Assignment in Smalltalk 160
Assignment in Objective-C 161
11.3 Equality 161
Equality in Objective-C and Object Pascal 163
Equality in Smalltalk 163
Equality in C++ 164
11.4 Type Conversion 164

12
Multiple Inheritance *169*
12.1 Incomparable Complex Numbers 169
12.2 Walking Menus 171
12.3 Name Ambiguity 173
Inheritance from Common Ancestors 174
12.4 Multiple Inheritance in C++ 175

13
Polymorphism *185*
13.1 Polymorphic Objects 188

13.2 Overloading 188
13.3 Overriding 192
13.4 Deferred Methods 194
Deferred Methods in C++ 195
Deferred Methods in Object Pascal 195
Deferred Methods in Objective-C 196
Deferred Methods in Smalltalk 196
13.5 Polymorphism 197
13.6 Efficiency and Polymorphism 199

14
Case Study: Smalltalk Collections 201
14.1 Implementation Approaches 203
14.2 Operations 204
Loops and Iteration 205
Injection 207
Computing the Number of Elements 207
Printing 208
Selection 208
Testing Elements 209
Retrieving Elements 210
Inserting Elements 210
14.3 The Yo-Yo Problem 211

15
Visibility and Dependency 215
15.1 Coupling and Cohesion 216
Varieties of Coupling 216
Varieties of Cohesion 217
Coupling and Cohesion in Object Oriented Systems 219
15.2 The Law of Demeter 219
15.3 Active Values 221
15.4 Subclass Clients and User Clients 222
15.5 Construction and Inheritance 224
15.6 Control of Access and Visibility 225
Visibility in Smalltalk 225
Visibility in Object Pascal 225
Visibility in C++ 226
Visibility in Objective-C 231

16
Case Study: A Billiards Game 233
16.1 A Simple Window Interface 233

16.2 A Graphical Universe 236
16.3 Graphical Components of the Billiards Game 238
16.4 The Main Program 239

17
Case Study: Container Classes in C++ *243*
17.1 The Basic Elements 243
17.2 Linked Lists 247
17.3 Sets 247
17.4 Ordered Collections 252
17.5 Tables 253

18
A Second Look at Classes *259*
18.1 The Nature of Classes 259
Classes as Types 259
Classes as Objects 263
18.2 Class Data 269
Class Variables in Smalltalk 270
Class Variables in C++ 270
Class Variables in Objective-C 271
18.3 Are Classes Necessary? 272
18.4 Parameterized Classes 276

19
Implementation *281*
19.1 Compilers 282
Virtual Method Tables 283
Name Encoding 285
Dispatch Tables 286
19.2 Interpreters 288
19.3 Advanced Reading 290

20
Further Information *293*
20.1 C++ 293
20.2 Objective-C 294
20.3 Smalltalk 295
20.4 Object Pascal 296
Turbo Pascal 296

20.5 *Other Languages 297*
20.6 *Selection of an Object-Oriented Language 297*
20.7 *Further Topics 298*
Programming Environments 298
Generic Dispatching 299
Persistence 300
Object Databases 300
20.8 *Research References 300*

A
Source for Eight Queens Puzzle *301*
A.1 *Eight Queens in Object Pascal 301*
A.2 *Eight Queens in C++ 303*
A.3 *Eight Queens in Objective-C 305*
A.4 *Eight Queens in Smalltalk 307*

B
Source for Solitaire Game *311*
B.1 *File solitaire.cc 311*
B.2 *File game.h 316*
B.3 *File card.h 317*
B.4 *File cardview.h 318*
B.5 *File cardview.cc 319*
B.6 *File pile.h 321*
B.7 *File pile.cc 325*

C
Source for Billiards Game *333*
C.1 *File simplewindow.p 333*
C.2 *File graph.p 341*
C.3 *File components.p 344*
C.4 *File billiard.p 348*

D
Source for Container Classes *353*
D.1 *File collect.h 353*
D.2 *File collect.cc 357*
D.3 *File ctest.cc 364*

E
Glossary *371*

Bibliography *381*

Index *391*