

CONTENTS

1	The Information Management Dilemma	1
	The Problem	2
	Modern Corporations Are Headed Toward Disaster	3
	What Does the Customer Want?	3
	Why Object-Oriented Technology Is Important to Developers	4
2	Managing Complexity: Analysis and Design	6
	Abstraction Mechanism	8
	Functions	8
	Modules	9
	Abstract Data Types	10
	Classes/Objects	11
	Message Passing	11
	Generalization/Specialization and Polymorphism	12
	Additional Relationships	12
	Associations	13
	Aggregation	14
	Behavior	16
	Static Behavior	16
	Dynamic Behavior	16
	Rules	18
	Complex Systems	19
3	Object-Oriented Programming	21
	What Is Object-Oriented Programming?	22
	Not a Silver Bullet	22
	New Paradigm	22
	Basic Object-Oriented Programming Concepts	22

Object-Oriented Programming Languages	27
Object-Based Programming	27
Class-Based Programming	27
Object-Oriented Programming	27
Advanced Object-Oriented (OO) Programming	28
Leading-Edge Object-Oriented Programming	28
Why C++?	28
Ways of Organizing Reality	29
The Simulation Model of Computation	31
The Object-Oriented Way of Organizing Reality	32
4 Finding the Objects	40
Object-Oriented Analysis: Model of Reality	41
Building the OO Model	41
Identification of Objects	42
Current Techniques	44
Using the Things to Be Modeled	45
Using the Definitions of Objects and Classes	45
Using Object Decomposition	45
Using Generalization	46
Using Subclasses	46
Using Object-Oriented Domain Analysis	47
Reusing an Application Framework	47
Reusing Class Hierarchies	48
Reusing Individual Objects and Classes	48
Using Subassemblies	48
Using Personal Experience	49
Traditional Techniques	50
Using Nouns	50
Using Traditional Data Flow Diagrams	51
Using Class-Responsibility-Collaboration (CRC) Cards	52
Recommended Approaches	53
Example	55
5 Identifying Responsibilities	58
What Is an Object?	59
What Is an Attribute?	60
Descriptive Attributes	60
Naming Attributes	61
What Is a Service?	61
What Is a Method?	61
Identifying Attributes	62
Specifying Attributes	64

Identifying Services	65
Specifying Services	66
Recommended Approach	66
Example	67
6 Specifying Static Behavior	71
What Is Behavior?	72
Techniques for Specifying Static Behavior	73
Techniques for Specifying Control	75
Techniques for Documenting Control	76
Process Flow Diagrams	76
Object Interaction Diagram (Collaboration Diagram)	76
Event Trace Diagram (Sequence Diagram)	77
The Use-Case Model	77
Techniques for Documenting Static Behavior	78
Preconditions and Postconditions	78
Flowcharting	78
Data Flow Diagrams	78
Structured English	78
Recommended Approach	79
Example	80
7 Dynamic Behavior	85
Introduction	86
Techniques for Identifying Dynamic Behavior	87
Common Lifecycle Forms	89
Models for Capturing the Lifecycle	89
Identifying and Specifying Events	91
Use Case and Scenario	91
Event Trace Diagram	92
Example	92
Specifying Dynamic Behavior	94
Event List	94
State Transition Table	97
Documenting Dynamic Behavior	101
State Diagrams	101
Recommended Approach	103
8 Identifying Relationships	105
Accessing Another Object's Services	106
Relationships	106
Generalization	107
Identifying and Specifying Generalization/Specialization	110

Object Aggregation	111
Classification of Aggregation	112
Assembly-parts (Component-Integral Composition)	113
Material-Object Composition	114
Portion-Object Composition	115
Place-Area Composition	116
Collection-Members Composition	116
Container-Content (Member-Bunch Composition)	117
Member-Partnership Composition	117
Objects and Aggregation Relationships	118
Links Between Objects	118
Identifying and Specifying Links and Aggregations	120
Managing Relationships	121
Documenting Relationships	123
Recommended Approach	125
Example	126
9 Rules	129
Introduction	130
Rules	131
Identifying Declarative Statements	131
Specifying and Documenting Rules	133
Mapping Rules to the Proper Object-Oriented Concepts	134
Documenting the Rules Using UML	136
Implementing Rules	136
Recommended Approach	137
10 The Model	139
Concepts	140
Concepts and Object-Oriented Model	141
Class	141
Association	141
Class Aggregation	142
Generalization/Specialization	142
Polymorphism	143
Instantiation	144
Refining the Model	144
Subsystems	145
Domain	145
Bridge	146
Organizing Subsystems	146
Horizontal Layers	147

Vertical Partitions	147
Combination	148
Identifying Subsystems	148
Recommended Approach	148
Example	149
11 Design	156
Introduction	157
System Design	158
Subsystems	158
Architectural Frameworks	159
Detailed Design	162
Class Design	162
Association Design	164
Generalization and Inheritance	165
Delegation	166
Orlando Treaty	167
Multiple Inheritance	168
12 C++ Fundamentals	172
History	173
Programming Elements	173
Keywords	174
Identifiers	174
Literals	175
Operators	175
Punctuators	175
Native Data Types	176
Basic Data Types	177
Constant Values	177
Symbolic Variables	178
Pointer Types	178
Constant Types	179
Reference Types	180
Enumeration Types	180
Array Types	180
Typedef Names	181
What Is a Statement?	181
Expressions	182
Compound Statements	182
Statement Flow Control	182
If statement	183
For statement	183
What Is a Function?	184

Function Invocation	185
Function Definition	186
Function Prototype	187
Inlining	187
Storage Class	188
Auto	188
Extern	188
Register	188
Static	189
13 Implementing Class	191
Components of a Class	192
Class Definition	192
Class Body	193
Visibility	193
Data Members	193
Member Functions	195
Generalization Using Inheritance	196
Recommended Approach	197
Example	198
14 Implementing Static Behavior	200
Function Definition	201
Return Type	202
Return Statement	202
Function Argument List	203
Passing Arguments	205
Pass-by-Value	205
Reference or Pointer Argument	206
Return Type as Reference or Pointer	208
Casting	209
Const and Defaults	210
Const	210
Default Initializers	212
Identifier Scope	212
Recommended Approach	213
Definition in the .h file	213
Definition in the .C file	214
15 Instantiating and Deleting Objects	217
Introduction	218
Constructors	218
Destructors	221

Using Constructors and Destructors Properly	224
Generalization and Constructors	225
Recommended Approach	226
Creating an Object	226
Destroying an Object	227
Coding Guidelines	228
Constructors	228
Destructors	229
16 Implementing Generalization/Specialization	230
Inheritance	31
Specifying a Derived Class	231
Inheriting from a Derived Class and Implementing Association	233
Adding Polymorphism	235
Abstract Class	238
Multiple Inheritance	239
Virtual Destructors	245
Derived Class Visibility	246
17 Implementing More Relationships	249
Introduction	250
Implementing Association	250
Implementing Attributes of an Association	251
Implementing Aggregation	251
Pointers	251
Arrays	253
Friends	255
Static Members	256
Implementing Association	257
Binary Association	257
Many-to-One Association	257
Many-to-Many Association	259
Implementing Friends	259
Class as a Friend	260
Function as a Friend	260
Implementing a One-to-Many Association Using a Friend	261
Implementing Aggregation	263
Buried Pointers	263
Embedded Objects	264
Implementing Static Members	265
Recommended Approach	267

18	Case Study: Breakout	268
	Requirements	269
	Acquiring Domain Expertise	270
	Expert's Knowledge	271
	Mechanics of Breakout	271
	Time Sequencing in Simulations	271
	Provided Technology Services	272
	Detailed Class Descriptions	275
	Geometric Subsystem Classes	275
	Display Subsystem Classes	277
	Collision Subsystem Classes	284
	UML Description of the Subsystems	286
	Geometric Subsystem	286
	Display Subsystem	286
	Collision Subsystem	286
19	Case Study: Team 1	289
	Step 1: Finding the Objects	290
	Step 2: Identifying Responsibilities	290
	Step 3: Specifying Behaviors	302
	Step 4: Specifying Relationships	311
	Step 5: Refinement	315
20	Case Study: Team 2	318
	Step 1: Finding The Objects	319
	Step 2: Identifying Responsibilities	320
	Step 3: Specifying Behaviors	338
	Step 4: Specifying Relationships	347
	Step 5: Refinement	354
21	Case Study: Design & Implementation	374
	Design	375
	Implementing Class	384
	Implementing Static Behavior	398
	Instantiating Objects	401
	Implementing Inheritance	405
	Implementing Relationships	426
	Bibliography	436
	Index	439