



# Course Description

SDT Corporate Plaza, 252 Ponce de Leon Suite 1000, Hato Rey, Puerto Rico, 00918  
Tel. (787) 641-7714 [www.sdtpr.com](http://www.sdtpr.com) [info@sistemadt.com](mailto:info@sistemadt.com)

**Publisher:** Official CIW Curriculum

**Code:** CCN05CEOOADPR303

**ISBN:** 1593020457

**Version:** 3.1

**Course Length:** 32 hours

**Audience:** Database developers, Internet application developers, database architects, middleware programmers, database administrators, Java developers, and client/server developers.

**Prerequisites:** Students must have CIW Foundations certification or equivalent experience, and have completed the Java Programming series or have equivalent experience with an object-oriented programming language.

**Download:**

[Table of Contents](#)

[Course Description](#)

[Sample Chapter](#)

## **What's in *Object Oriented Analysis and Design* #58; Instructor Guide:**

*Object-oriented Analysis and Design* is a 30-hours course that teaches object-oriented analysis and design techniques using UML in the context of the Unified Software Development Process. The course provides an introduction to object-oriented theory and the software development life cycle. Students learn proper analysis and design procedures, and their roles in the development process. Students gain hands-on experience with all phases of the development process: requirements, analysis, design, construction, and testing.

Instructor materials include ¼ instructor notes in the margins, activity worksheets, and implementation table/syllabus, and a customizable PowerPoint slide presentation.

## **Why *Object Oriented Analysis and Design* #58; Instructor Guide**

- Leads to CIW certification.
- CIW provides objective validation of critical Internet skills.
- CIW certification is a worldwide credential that establishes an individual as an expert in Internet technologies.

# Object-Oriented Analysis and Design Series Course: Object-Oriented Analysis and Design (March 2003)

---

*Object-Oriented Analysis and Design* is a course that teaches object-oriented analysis and design techniques using UML in the context of the Unified Software Development Process. The course provides an introduction to object-oriented theory and the software development life cycle. Students learn proper analysis and design procedures, and their roles in the development process. Students gain hands-on experience with all phases of the development process: requirements, analysis, design, construction, and testing.

---

## Topics

### Section I — The Object Paradigm

#### Introduction to Software Engineering

Evolution of Software Engineering  
Object Technology  
Advantages and Disadvantages of an Object-Oriented Approach  
Analysis and Design

#### Understanding the Object Paradigm

Classes  
Encapsulation  
Abstraction  
Object Relationships  
Application Objects

#### Inheritance and Polymorphism

Inheriting Attributes and Methods  
Extending Components with Inheritance  
Polymorphism  
Abstract Classes  
Multiple Inheritance  
Subtyping vs. Subclassing

#### Software Development Life Cycle

Software Development Life Cycle Model  
Waterfall Life Cycle Model  
V-Shaped Life Cycle Model  
Incremental Life Cycle Model  
Spiral Life Cycle Model

### Section II — Tools of Analysis and Design

#### The Unified Software Development Process

Software Development Process  
The Unified Process  
Life Cycle of the Unified Process  
Cycles, Phases, Iterations, Core Workflows, and Workflows and Iterations

#### Unified Modeling Language (UML)

Models  
Views

#### Computer-Aided Software Engineering (CASE)

Introduction to CASE  
Selecting a CASE Tool

### Section III — The Requirements Workflow

#### The Requirements Workflow

Introduction to Requirements Capture  
Activities of the Requirements Workflow  
Requirements and the Unified Process

#### Use-Case Modeling

Introduction to Use-Case Diagrams  
Developing a Use-Case Model

#### Activity Diagrams

Branches and Merges  
Forks and Joins  
Library System Activity Diagrams

#### Interface Design and Prototyping

User Interface Design  
User Interface Ergonomics  
User Interface Prototyping  
Specifying System Interfaces

### Section IV — The Analysis Workflow

#### The Analysis Workflow

Introduction to Analysis  
Analysis Classes  
Use-Case Realizations – Analysis  
Analysis and the Unified Process

#### Analysis Modeling

Collaboration Diagrams  
Flow of Events and Special Requirements  
Class-Responsibility-Collaboration Cards  
Class Analysis

### Section V — The Design Workflow

#### The Design Workflow

Design Model  
Use-Case Realizations – Design  
Design and the Unified Process

#### Architecture Modeling

Package Diagrams  
Deployment Diagrams

#### Class Diagrams

Association, Aggregation, Composition, and Generalization  
Check Out Asset Class Diagram

#### Sequence Diagrams

Return Values, Message Conditions, Deletion, Multiplicity, and Return Stack

#### Statechart Diagrams

States, Transitions, and Superstates

#### Design Quality Issues

Elements of Good Design  
Object-Oriented Design Metrics  
Chidamber and Kemerer  
Metrics for Object-Oriented Design  
Designing for Reuse

#### The Model-View-Controller Paradigm (MVC)

Model-View-Controller  
Building MVC Classes  
Hybrid MVC

#### Refactoring

Identifying New Methods  
Identifying Methods That Can Be Moved, Inheritance Opportunities  
Clarifying Variable Names

### Section VI — The Implementation Workflow

#### The Implementation Workflow

Identifying Components  
Integration Build Planning  
Generating Code from Design Classes  
Unit Testing  
Implementation and the Unified Process

### Section VII — The Test Workflow

#### The Test Workflow

Test Cases, Procedures and Components  
Test and the Unified Process

## **Object-Oriented Analysis and Design Series Course: Object-Oriented Analysis and Design (March 2003) Continued**

---

### **Target Audience**

Database developers and administrators, Internet application developers, middleware programmers, Java developers, and client/server developers.

### **Job Responsibilities**

Develop n-tier database and legacy connectivity solutions for Web applications using Java, Java APIs, Java Database Connectivity solutions, middleware tools, and distributed object models.

### **Prerequisites**

Students must have CIW Foundations certification or equivalent experience, and have completed the Java Programming series (*Java Programming Fundamentals*) or have equivalent experience with an object-oriented programming language.

### **Duration**

30 hours



# OBJECT-ORIENTED ANALYSIS AND DESIGN: INSTRUCTOR GUIDE TABLE OF CONTENTS

Course Description .....	xiii
ProsoftTraining Courseware .....	xiii
Course Objectives .....	xv
Classroom Setup .....	xv
System Requirements .....	xv
Conventions and Graphics Used in This Book .....	xvii
Classroom Setup Guide .....	Classroom Setup Guide-1

## **Section I: The Object Paradigm**

---

<b>Lesson 1: Introduction to Software Engineering</b> .....	<b>1-1</b>
Pre-Assessment Questions .....	1-2
Evolution of Software Engineering .....	1-3
Object Technology .....	1-4
Advantages of an Object-Oriented Approach .....	1-5
Disadvantages of an Object-Oriented Approach .....	1-6
Analysis and Design .....	1-6
Lesson 1 Review .....	1-8
Lesson 1 Instructor Section .....	1-9
<b>Lesson 2: Understanding the Object Paradigm</b> .....	<b>2-1</b>
Pre-Assessment Questions .....	2-2
Classes .....	2-3
Encapsulation .....	2-3
Abstraction .....	2-6
Object Relationships .....	2-6
Application Objects .....	2-7
Lesson 2 Review .....	2-10
Lesson 2 Instructor Section .....	2-11
<b>Lesson 3: Inheritance and Polymorphism</b> .....	<b>3-1</b>
Pre-Assessment Questions .....	3-2
Inheritance .....	3-3
Inheriting Attributes .....	3-3
Inheriting Methods .....	3-4
Extending Components with Inheritance .....	3-5

Polymorphism .....	3-6
Abstract Classes .....	3-7
Multiple Inheritance .....	3-8
Subtyping vs. Subclassing .....	3-10
Lesson 3 Review .....	3-12
Lesson 3 Instructor Section .....	3-14

**Lesson 4: Software Development Life Cycle 4-1**

Pre-Assessment Questions.....	4-2
Software Development Life Cycle Model .....	4-3
Waterfall Life Cycle Model.....	4-4
V-Shaped Life Cycle Model.....	4-5
Incremental Life Cycle Model .....	4-6
Spiral Life Cycle Model .....	4-7
Lesson 4 Review .....	4-9
Lesson 4 Instructor Section .....	4-10

**Section II: Tools of Analysis and Design**

---

**Lesson 5: The Unified Software Development Process 5-1**

Pre-Assessment Questions.....	5-2
Software Development Process.....	5-3
The Unified Process .....	5-3
Life Cycle of the Unified Process.....	5-4
Unified Process Cycles .....	5-4
Unified Process Phases .....	5-5
Unified Process Iterations .....	5-6
Core Workflows of the Unified Process .....	5-7
Unified Process Workflows and Iterations.....	5-9
Lesson 5 Review .....	5-10
Lesson 5 Instructor Section .....	5-11

**Lesson 6: Unified Modeling Language 6-1**

Pre-Assessment Questions.....	6-2
Introduction to UML .....	6-3
UML Models .....	6-3
UML Views .....	6-4
Lesson 6 Review .....	6-6
Lesson 6 Instructor Section .....	6-7

**Lesson 7: Computer-Aided Software Engineering (CASE) 7-1**

Pre-Assessment Questions.....	7-2
Introduction to CASE.....	7-3
Selecting a CASE Tool.....	7-3
Lesson 7 Review .....	7-6
Lesson 7 Instructor Section .....	7-7

**Section III: The Requirements Workflow**

---

**Lesson 8: The Requirements Workflow 8-1**

Pre-Assessment Questions.....	8-2
Introduction to Requirements Capture.....	8-3
Activities of the Requirements Workflow .....	8-3
Requirements and the Unified Process.....	8-6
Lesson 8 Review .....	8-8
Lesson 8 Instructor Section .....	8-9

**Lesson 9: Use-Case Modeling 9-1**

Pre-Assessment Questions.....	9-2
Introduction to Use-Case Diagrams .....	9-3
Developing a Use-Case Model .....	9-6
Lesson 9 Review .....	9-17
Lesson 9 Instructor Section .....	9-18

**Lesson 10: Activity Diagrams 10-1**

Pre-Assessment Questions.....	10-2
Introduction to Activity Diagrams .....	10-3
Branches and Merges in Activity Diagrams .....	10-3
Forks and Joins in Activity Diagrams.....	10-4
Library System Activity Diagrams .....	10-5
Lesson 10 Review .....	10-13
Lesson 10 Instructor Section .....	10-14

**Lesson 11: Interface Design and Prototyping 11-1**

Pre-Assessment Questions.....	11-2
User Interface Design .....	11-3
User Interface Ergonomics.....	11-3
User Interface Prototyping .....	11-6
Specifying System Interfaces.....	11-14
Lesson 11 Review .....	11-16
Lesson 11 Instructor Section .....	11-18

**Section IV: The Analysis Workflow**

---

**Lesson 12: The Analysis Workflow 12-1**

Pre-Assessment Questions.....	12-2
Introduction to Analysis .....	12-3
Analysis Classes.....	12-3
Use-Case Realizations—Analysis.....	12-5
Analysis and the Unified Process .....	12-6
Lesson 12 Review .....	12-7
Lesson 12 Instructor Section .....	12-8

**Lesson 13: Analysis Modeling 13-1**

Pre-Assessment Questions.....	13-2
Collaboration Diagrams .....	13-3
Flow of Events and Special Requirements .....	13-6
Class-Responsibility-Collaboration Cards .....	13-6
Class Analysis .....	13-7
Lesson 13 Review .....	13-10

Lesson 13 Instructor Section .....	13-11
------------------------------------	-------

---

**Section V: The Design Workflow**

---

**Lesson 14: The Design Workflow** **14-1**

Pre-Assessment Questions.....	14-2
Design Model.....	14-3
Use-Case Realizations—Design .....	14-4
Design and the Unified Process.....	14-6
Lesson 14 Review .....	14-7
Lesson 14 Instructor Section .....	14-9

**Lesson 15: Architecture Modeling** **15-1**

Pre-Assessment Questions.....	15-2
Introduction to Architecture Modeling.....	15-3
Package Diagrams .....	15-3
Deployment Diagrams .....	15-6
Lesson 15 Review .....	15-8
Lesson 15 Instructor Section .....	15-10

**Lesson 16: Class Diagrams** **16-1**

Pre-Assessment Questions.....	16-2
Introduction to Class Diagrams .....	16-3
Association.....	16-3
Aggregation .....	16-4
Composition .....	16-6
Generalization .....	16-7
Check Out Asset Class Diagram .....	16-9
Lesson 16 Review .....	16-11
Lesson 16 Instructor Section .....	16-12

**Lesson 17: Sequence Diagrams** **17-1**

Pre-Assessment Questions.....	17-2
Introduction to Sequence Diagrams .....	17-3
Return Values in Sequence Diagrams .....	17-3
Message Conditions in Sequence Diagrams.....	17-4
Deletion in Sequence Diagrams .....	17-5
Multiplicity in Sequence Diagrams.....	17-5
Return Stack in Sequence Diagrams.....	17-7
Lesson 17 Review .....	17-8
Lesson 17 Instructor Section .....	17-9

**Lesson 18: State Diagrams** **18-1**

Pre-Assessment Questions.....	18-2
Introduction to State Diagrams.....	18-3
States.....	18-3
State Transitions .....	18-3
Superstates .....	18-4
Lesson 18 Review .....	18-8
Lesson 18 Instructor Section .....	18-9

<b>Lesson 19: Design Quality Issues</b>	<b>19-1</b>
Pre-Assessment Questions.....	19-2
Elements of Good Design.....	19-3
Object-Oriented Design Metrics .....	19-4
Chidamber and Kemerer Metrics .....	19-4
Metrics for Object-Oriented Design (MOOD).....	19-6
Designing for Reuse.....	19-8
Lesson 19 Review .....	19-10
Lesson 19 Instructor Section .....	19-11

<b>Lesson 20: The Model-View-Controller Paradigm</b>	<b>20-1</b>
Pre-Assessment Questions.....	20-2
Introduction to Model-View-Controller .....	20-3
Model-View-Controller Architecture .....	20-4
Building Model-View-Controller Classes.....	20-4
Hybrid MVC.....	20-8
Lesson 20 Review .....	20-10
Lesson 20 Instructor Section .....	20-11

<b>Lesson 21: Refactoring</b>	<b>21-1</b>
Pre-Assessment Questions.....	21-2
Introduction to Refactoring.....	21-3
Identifying New Methods.....	21-3
Identifying Methods That Can Be Moved .....	21-5
Identifying Inheritance Opportunities .....	21-7
Clarifying Variable Names.....	21-8
Lesson 21 Review .....	21-9
Lesson 21 Instructor Section .....	21-10

---

## Section VI: The Implementation Workflow

<b>Lesson 22: The Implementation Workflow</b>	<b>22-1</b>
Pre-Assessment Questions.....	22-2
The Implementation Workflow .....	22-3
Identifying Components.....	22-3
Integration Build Planning.....	22-4
Generating Code from Design Classes.....	22-5
Unit Testing.....	22-6
Implementation and the Unified Process .....	22-7
Lesson 22 Review .....	22-9
Lesson 22 Instructor Section .....	22-10

---

## Section VII: The Test Workflow

<b>Lesson 23: The Test Workflow</b>	<b>23-1</b>
Pre-Assessment Questions.....	23-2
The Test Workflow .....	23-3
Test Cases.....	23-3
Test Procedures .....	23-4
Test Components.....	23-5
Test and the Unified Process.....	23-5



Lesson 23 Review .....	23-7
Lesson 23 Instructor Section .....	23-8

<b>Course Assessment</b>	<b>Course Assessment-1</b>
<b>Appendixes</b>	<b>Appendixes-1</b>
<b>Glossary</b>	<b>Glossary-1</b>
<b>Index</b>	<b>Index-1</b>
<b>Supplemental CD-ROM Contents</b>	<b>Supplemental CD-ROM Contents-1</b>
<b>Handouts: Activities</b>	<b>Handouts: Activities-1</b>
<b>Handouts: Optional Labs</b>	<b>Handouts: Optional Labs-1</b>
<b>Handouts: Quizzes</b>	<b>Handouts: Quizzes-1</b>
<b>Handout: Course Assessment</b>	<b>Handout: Course Assessment-1</b>

**List of Labs**

Lab 1-1: Using an object-oriented approach.....	1-7
Lab 2-1: Using the object paradigm .....	2-8
Lab 3-1: Using inheritance and polymorphism.....	3-11
Lab 7-1: Using ArgoUML.....	7-4
Lab 8-1: Creating a candidate requirements document .....	8-6
Lab 9-1: Developing a use-case model.....	9-16
Lab 10-1: Creating activity diagrams .....	10-12
Lab 11-1: Defining interface requirements .....	11-14
Lab 13-1: Creating analysis models .....	13-8
Lab 15-1: Creating package diagrams .....	15-5
Lab 15-2: Creating deployment diagrams.....	15-7
Lab 16-1: Creating class diagrams .....	16-9
Lab 17-1: Creating sequence diagrams .....	17-7
Lab 18-1: Creating state diagrams .....	18-6
Lab 19-1: Applying design metrics .....	19-7
Lab 22-1: Performing implementation .....	22-7
Lab 23-1: Performing test modeling .....	23-6

**List of Activities**

Activity 1-1: Identifying objects .....	1-10
Activity 2-1: Identifying class relationships .....	2-12
Activity 3-1: Identifying class relationships .....	3-15
Activity 8-1: Identifying nonfunctional requirements .....	8-10
Activity 9-1: Identifying include and extend relationships.....	9-19
Activity 10-1: Evaluating activity diagrams .....	10-15
Activity 11-1: Defining system interfaces.....	11-19
Activity 12-1: Classifying analysis classes.....	12-9
Activity 15-1: Identifying communications links .....	15-11
Activity 16-1: Identifying aggregation and composition.....	16-13
Activity 19-1: Applying design metrics .....	19-12
Activity 23-1: Identifying white-box and black-box test cases .....	23-9

## List of Optional Labs

Optional Lab 13-1: Evaluating collaboration diagrams .....	13-12
Optional Lab 17-1: Creating interaction diagrams .....	17-10
Optional Lab 18-1: Creating state diagrams .....	18-10
Optional Lab 21-1: Applying refactoring techniques .....	21-11
Optional Lab 22-1: Realizing design .....	22-11

## List of Quizzes

Lesson 1 Quiz .....	1-11
Lesson 2 Quiz .....	2-13
Lesson 3 Quiz .....	3-15
Lesson 4 Quiz .....	4-11
Lesson 5 Quiz .....	5-12
Lesson 6 Quiz .....	6-8
Lesson 7 Quiz .....	7-8
Lesson 8 Quiz .....	8-11
Lesson 9 Quiz .....	9-20
Lesson 10 Quiz .....	10-16
Lesson 11 Quiz .....	11-20
Lesson 12 Quiz .....	12-9
Lesson 13 Quiz .....	13-13
Lesson 14 Quiz .....	14-10
Lesson 15 Quiz .....	15-12
Lesson 16 Quiz .....	16-14
Lesson 17 Quiz .....	17-11
Lesson 18 Quiz .....	18-10
Lesson 19 Quiz .....	19-14
Lesson 20 Quiz .....	20-12
Lesson 21 Quiz .....	21-13
Lesson 22 Quiz .....	22-12
Lesson 23 Quiz .....	23-11

## List of Figures

Figure 1-1: Data flow diagram example .....	1-3
Figure 1-2: Order object .....	1-4
Figure 1-3: Student registration data flow diagram .....	1-7
Figure A1-1: Elements of inventory management system .....	1-10
Figure 2-1: Book class .....	2-6
Figure 2-2: The "has a" relationship .....	2-6
Figure 2-3: The "uses a" relationship .....	2-7
Figure 2-4: Interface objects .....	2-7
Figure 2-5: Control objects .....	2-8
Figure A2-1: Automobile diagram .....	2-12
Figure 3-1: Inheritance .....	3-3
Figure 3-2: Inheritance relationships .....	3-4
Figure 3-3: DueDate class .....	3-6
Figure 3-4: Multiple inheritance .....	3-9
Figure 3-5: Subtyping .....	3-10
Figure A3-1: Air traffic control system .....	3-15
Figure 4-1: General software development life cycle model .....	4-3
Figure 4-2: Waterfall life cycle model .....	4-4
Figure 4-3: V-Shaped life cycle model .....	4-5
Figure 4-4: Incremental life cycle model .....	4-6

Figure 4-5: Spiral life cycle model .....	4-8
Figure 5-1: Single cycle.....	5-4
Figure 5-2: Complete life cycle .....	5-5
Figure 5-3: Project risk across one cycle .....	5-5
Figure 5-4: Core workflows .....	5-7
Figure 5-5: Workflows and iterations .....	5-9
Figure 8-1: Candidate requirements document .....	8-4
Figure 8-2: Requirements and the Unified Process .....	8-6
Figure 9-1: Actor example.....	9-3
Figure 9-2: Use case example .....	9-3
Figure 9-3: Association relationship.....	9-4
Figure 9-4: Generalization relationship.....	9-5
Figure 9-5: Include relationship.....	9-5
Figure 9-6: Extend relationship .....	9-6
Figure 9-7: Use-Case Model.....	9-7
Figure A9-1: ATM System use-case diagram .....	9-19
Figure A9-2: Cell Phone use-case diagram .....	9-19
Figure A9-3: Microcassette Recorder use-case diagram .....	9-20
Figure 10-1: Check Out Asset activity diagram .....	10-3
Figure 10-2: Check Out Asset branch and merge .....	10-4
Figure 10-3: Check Out Asset fork and join .....	10-5
Figure 10-4: Add Asset to Database activity diagram.....	10-6
Figure 10-5: Delete Asset from Database activity diagram .....	10-7
Figure 10-6: Generate Report activity diagram .....	10-7
Figure 10-7: Check Out Asset activity diagram .....	10-8
Figure 10-8: Check In Asset activity diagram .....	10-9
Figure 10-9: Issue Library Card activity diagram .....	10-10
Figure 10-10: Access Account Information activity diagram .....	10-10
Figure 10-11: Generate Overdue Notices activity diagram .....	10-11
Figure 10-12: Pay Overdue Fine activity diagram .....	10-11
Figure A10-1: ATM Transaction activity diagram.....	10-15
Figure 11-1: Standard Windows toolbar.....	11-4
Figure 11-2: State visualization .....	11-5
Figure 11-3: Interface Requirements Document .....	11-7
Figure 12-1: UML class stereotypes .....	12-3
Figure 12-2: Alternative UML notation for entity class.....	12-3
Figure 12-3: Boundary class example .....	12-4
Figure 12-4: Entity class example .....	12-4
Figure 12-5: Control class example.....	12-5
Figure 12-6: Analysis and the Unified Process .....	12-6
Figure A12-1: Analysis classes.....	12-9
Figure 13-1: Collaboration diagram example.....	13-3
Figure 13-2: Sequence numbers and parameter passing.....	13-3
Figure 13-3: Return value example—collaboration diagram.....	13-4
Figure 13-4: Message condition example—collaboration diagram .....	13-4
Figure 13-5: Multiplicity example—collaboration diagram .....	13-5
Figure 13-6: Check Out Asset collaboration diagram.....	13-5
Figure 13-7: Example CRC card.....	13-7
Figure OL13-1: Check Out Asset collaboration diagram .....	13-12
Figure 14-1: Design subsystem example .....	14-3
Figure 14-2: UML class notation example .....	14-4
Figure 14-3: Design class notation.....	14-5
Figure 14-4: Design and the Unified Process.....	14-6
Figure 15-1: Package example .....	15-3
Figure 15-2: Package dependencies .....	15-4
Figure 15-3: Embedded packages .....	15-4
Figure 15-4: Library system package diagram.....	15-5

Figure 15-5: Deployment diagram example .....	15-6
Figure 15-6: Three-tier model .....	15-7
Figure A15-1: Intranet deployment diagram.....	15-11
Figure 16-1: UML class notation .....	16-3
Figure 16-2: Association example—class diagram .....	16-3
Figure 16-3: Navigability example—class diagram.....	16-4
Figure 16-4: Multiplicity example—class diagram .....	16-4
Figure 16-5: Aggregation example—class diagram.....	16-5
Figure 16-6: Composition example—class diagram .....	16-6
Figure 16-7: Alternative notation for composition.....	16-7
Figure 16-8: Generalization example—class diagram .....	16-8
Figure 16-9: The "implements" relationship .....	16-8
Figure 16-10: Check Out Asset class diagram.....	16-9
Figure A16-1: Class pair relationships .....	16-13
Figure 17-1: Sequence diagram example.....	17-3
Figure 17-2: Return values example—sequence diagram.....	17-4
Figure 17-3: Message condition example—sequence diagram .....	17-5
Figure 17-4: Check Out Asset sequence diagram .....	17-6
Figure OL17-1: Check Out Asset sequence diagram .....	17-10
Figure 18-1: State example .....	18-3
Figure 18-2: State diagram example .....	18-4
Figure 18-3: CheckOutController state diagram without superstate .....	18-5
Figure 18-4: CheckOutController state diagram with superstate .....	18-6
Figure 19-1: Coupling between object classes (CBO) example.....	19-4
Figure 19-2: Depth of inheritance tree (DIT) example .....	19-5
Figure 19-3: Number of children (NOC) example .....	19-5
Figure 19-4: Invisibility.....	19-6
Figure A19-1: Check Out Asset class diagram.....	19-12
Figure A19-2: CBO metric for Check Out Asset class diagram.....	19-12
Figure A19-3: DIT metric for Check Out Asset class diagram .....	19-13
Figure 20-1: Model-View-Controller architecture.....	20-4
Figure 20-2: MVC application¼pie chart view.....	20-5
Figure 20-3: MVC application¼bar graph view.....	20-7
Figure 20-4: Hybrid MVC architecture .....	20-8
Figure 20-5: Hybrid MVC application—slider .....	20-8
Figure 22-1: Implementation workflow.....	22-3
Figure 22-2: Deployment diagram.....	22-4
Figure 22-3: Design classes .....	22-5
Figure 22-4: Implementation and the Unified Process .....	22-7
Figure OL22-1: Check Out Asset class diagram .....	22-11
Figure 23-1: Test cases.....	23-3
Figure 23-2: Test and the Unified Process.....	23-5

**List of Tables**

Table 6-1: UML views .....	6-4
Table 7-1: CASE tools timeline.....	7-3
Table 9-1: Use-case relationships .....	9-4
Table 11-1: Backus-Naur Form.....	11-14
Table 23-1: Test case for the Check Out Asset use case .....	23-3

**Appendixes**

- Appendix A: Objectives and Locations\*
  - Appendix B: Works Consulted\*
- \* Appendix found on Supplemental CD-ROM