

# Contents

<b>Introduction</b>	<b>xxix</b>
---------------------	-------------

## **Part I: Fundamentals of Professional Development**

<b>Chapter 1: Introduction to Object-Oriented Programming</b>	<b>3</b>
---	----------

---

<b>What Is Object-Oriented Programming?</b>	<b>3</b>
OOP Advantages	4
A Real-World Example	4
<b>Understanding OOP Concepts</b>	<b>5</b>
<b>Classes</b>	<b>6</b>
Objects	6
Inheritance	15
Interfaces	25
Encapsulation	27
<b>Changes to OO in PHP6</b>	<b>29</b>
<b>Summary</b>	<b>30</b>

<b>Chapter 2: Unified Modeling Language (UML)</b>	<b>31</b>
---	-----------

---

<b>Requirements Gathering</b>	<b>31</b>
Interviewing the Client	32
Use Case Diagrams	33
<b>Class Diagrams</b>	<b>35</b>
Modeling the Domain	35
Relationships	37
Implementation	39
<b>Activity Diagrams</b>	<b>42</b>
<b>Sequence Diagrams</b>	<b>44</b>
<b>State Diagrams</b>	<b>46</b>
<b>Component and Deployment Diagrams</b>	<b>47</b>
<b>Summary</b>	<b>48</b>

# Contents

---

<b>Chapter 3: Putting Objects to Work</b>	<b>49</b>
<b>Creating the Contact Manager</b>	<b>49</b>
The Contact Manager UML Diagrams	50
The PropertyObject Class	56
The Contact Type Classes	58
The DataManager Class	62
The Entity, Individual, and Organization Classes	64
<b>Making Use of the System</b>	<b>71</b>
<b>Summary</b>	<b>74</b>
<b>Chapter 4: Design Patterns</b>	<b>75</b>
<b>The Composite Pattern</b>	<b>76</b>
Implementation	78
Considerations	82
<b>The Observer Pattern</b>	<b>83</b>
Widgets	84
Considerations	89
<b>The Decorator Pattern</b>	<b>89</b>
Implementation	92
Using the Decorator	93
Considerations	94
<b>The Facade Pattern</b>	<b>95</b>
<b>The Builder Pattern</b>	<b>96</b>
Implementation	97
Considerations	100
<b>Summary</b>	<b>100</b>
<b>Chapter 5: Collections</b>	<b>103</b>
<b>Purpose of the Collection Class</b>	<b>103</b>
<b>Designing the Collection Class</b>	<b>105</b>
<b>Collection Class Basics</b>	<b>106</b>
The addItem Method	106
The removeItem and getItem Methods	107
Other Methods	108
Using the Collection Class	108
<b>Implementing Lazy Instantiation</b>	<b>109</b>
Callbacks	110
The setLoadCallback Method in the Collection Class	115
<b>Using the Collection Class</b>	<b>119</b>
<b>Improving the Collection Class</b>	<b>125</b>
<b>Summary</b>	<b>126</b>

---

<b>Chapter 6: Database Abstraction with PDO</b>	<b>127</b>
<b>PHP and Databases</b>	<b>128</b>
PHP's Database Support	128
PHP and PostgreSQL	128
Recap	129
The Problem	132
Wrapper Classes	133
Summing Up	134
<b>Database Abstraction</b>	<b>134</b>
What's Needed	134
Introducing PDO	135
Getting Started with PDO	135
<b>Working with PDO</b>	<b>136</b>
The PDO Class	137
Executing Queries	138
The PDOStatement Class	139
Prepared Statements	139
Write-Only Statements	141
Transactions and Committals	142
Constants	143
Handling Errors	143
<b>Advanced PDO Techniques</b>	<b>145</b>
Large Objects	146
Database-Specific Functions	147
Persistent Connections	147
Stored Procedures	148
Singleton Instantiation	149
<b>Limitations of PDO</b>	<b>150</b>
Query Syntax	150
Feature Emulation	150
Non-SQL Data Sources	151
Beyond PDO	151
<b>Summary</b>	<b>152</b>

## **Part II: Learning Advanced Development Techniques**

---

<b>Chapter 7: Object Relation Mapping</b>	<b>155</b>
<b>Types of Classes</b>	<b>155</b>
Utility Classes	156
Business Classes	156

# Contents

---

<b>Business Objects in Detail</b>	<b>157</b>
The Design	157
Prototyping	158
Adding Data Binding	160
A Working Business Object	162
Ensuring Requirements Are Met	166
Understanding How It Works	169
Limitations	172
<b>A Smarter Approach</b>	<b>173</b>
Lazy Instantiation	173
Property Monitoring	173
Reusability	173
<b>DataBoundObject Class</b>	<b>173</b>
Design	174
The Code	175
Example Implementation	178
Understanding How It Works	180
Benefits	184
<b>Summary</b>	<b>185</b>
<b>Chapter 8: Event-Driven Programming</b>	<b>187</b>
<hr/>	
<b>Understanding Events</b>	<b>187</b>
<b>Using OOP to Handle Events</b>	<b>188</b>
Designing an Event-Driven Solution	189
Implementing the Solution	191
Implementing Security	196
Pausing for Thought	198
<b>Summary</b>	<b>199</b>
<b>Chapter 9: Logging and Debugging</b>	<b>201</b>
<hr/>	
<b>Creating a Logging Mechanism</b>	<b>201</b>
Simple File Logging	201
Example File System Layout	202
The Logger Class	203
Extending the Logger Class	208
<b>Creating a Debugging Mechanism</b>	<b>219</b>
<b>Summary</b>	<b>223</b>

---

<b>Chapter 10: Writing and Using Web Services</b>	<b>225</b>
<b>SOAP</b>	<b>226</b>
SOAP and PHP6	226
The PHP6 SOAP Extension	227
<b>Making a SOAP Client</b>	<b>230</b>
Looking Under the Hood	232
Exception Handling in the SOAP Client	239
<b>Making a SOAP Server</b>	<b>240</b>
Creating and Registering Functions	240
Creating the WSDL Document	241
<b>REST</b>	<b>243</b>
REST and PHP6	244
<b>Implementing the REST Client</b>	<b>244</b>
<b>Making a REST Server</b>	<b>245</b>
<b>Summary</b>	<b>246</b>
<b>Chapter 11: Communicating with Users</b>	<b>247</b>
<b>Why Communicate?</b>	<b>247</b>
Reasons to Communicate with Users	248
Thinking Outside the Web Browser	250
<b>Types of Communication</b>	<b>250</b>
All Communications Have . . .	250
Not All Communications Have . . .	251
What About Recipients?	251
<b>Communication As a Class Hierarchy</b>	<b>251</b>
The Recipient Class: A Quick Test of OOP-Like Thinking	252
The Communication Class	255
<b>E-mailing Your Users</b>	<b>259</b>
Building a Test Version	259
Getting the Message Across	263
Being Smart with Templates	268
Using MIME	270
<b>Other Communication Subclasses</b>	<b>270</b>
SMS Text Messaging	270
Fax	271
<b>Other Considerations</b>	<b>271</b>
Blocking Activity	271
Deliverability	272
<b>Summary</b>	<b>274</b>

## **Chapter 12: Sessions and Authentication** **275**

---

<b>Introduction to Sessions</b>	<b>276</b>
How HTTP Works	276
Definition of a Session	278
Perpetuation of a Session	279
Session Security	281
Good Session Practice	285
<b>How PHP Implements Sessions</b>	<b>287</b>
Basic PHP Sessions	288
Limitations of Basic PHP Sessions	290
<b>Creating an Authentication Class</b>	<b>290</b>
Connecting PHP Session Management to a Database	290
Introducing the HTTPSession Class	292
Database Schema	292
The Code: HTTPSession.phpm	293
The Code: Testing the HTTPSession Class	297
How the HTTPSession Class Works	300
Performance Considerations	302
Putting It All Together	304
<b>Summary</b>	<b>304</b>

## **Chapter 13: Application Architecture** **305**

---

<b>Introducing MVC</b>	<b>306</b>
The Model	306
The View	307
The Controllers	307
MVC in Web Applications	308
MVC in PHP	309
<b>The MVC Mini Toolkit</b>	<b>311</b>
Introducing the Toolkit	311
Using the Toolkit	324
A Word about PRG	331
The Toolkit in the Wild	332
<b>True Templating</b>	<b>333</b>
Recapping Native PHP Templating	333
The Pitfalls of Native Templating	333
True Templating with Smarty Templates	334
Installing Smarty	334
Using Smarty	335

---

Advanced Smarty	340
When to Use Smarty vs. Traditional Templating	342
<b>A Spot of Modeling Work</b>	<b>343</b>
Introducing Unit Testing	343
Introduction to PHPUnit	346
Why Bother?	351
<b>Summary</b>	<b>352</b>
<b>Chapter 14: PHP Application Frameworks</b>	<b>355</b>
<hr/>	
<b>Introduction to Frameworks</b>	<b>355</b>
Application Structure	356
Separation of Code and Display Logic	356
Validation	357
URL Rewriting	357
Form Persistence	358
Object Relation Mapping	358
Database Abstraction	359
Utility Classes	359
<b>Use Case Scenario</b>	<b>359</b>
Underlying Database	360
<b>An Example Application</b>	<b>361</b>
Introducing Ulysses	361
Installing Ulysses	361
Working with Ulysses	363
Setting Up	363
Business Objects	364
Servlets	366
Processing User Input	371
Persistence	372
Validation	373
A Quick Check	374
Saving Input	374
Heading Home	375
A Quick Summary	375
<b>Some Other Contenders</b>	<b>377</b>
Prado	377
Zend Framework	378
<b>Deployment Considerations</b>	<b>379</b>
<b>Summary</b>	<b>380</b>

## **Part III: A Real-World Case Study**

### **Chapter 15: Project Overview 383**

---

<b>Widget World</b>	<b>384</b>
<b>The Widget World Landscape</b>	<b>386</b>
The Technical Layer	386
The Financial Layer	386
The Political Layer	386
You	387
Is It Really About Technology?	387
<b>The Development Approach</b>	<b>387</b>
What This Means to You	389
The Technology	391
<b>Summary</b>	<b>392</b>

### **Chapter 16: Project Management 393**

---

<b>Do Your Homework</b>	<b>394</b>
Why Is the Project Happening?	394
Who Is the Project For?	394
What Is the History of the Project?	396
What Are the Anticipated Prerequisites of the Project?	397
<b>Receiving the Formal Brief</b>	<b>397</b>
Business Requirements	398
Scope	399
Timelines	400
Budget	401
Commercial Terms	403
Future Plans	404
Look and Feel	404
Technology	405
Support	405
What Now?	405
<b>Constructing the Pitch</b>	<b>406</b>
Pitches Versus Quotes	406
Pitches Versus Specifications	406
Who to Involve When Pitching	407
When to Go the Extra Mile	408
When to Say No	408
Structuring the Pitch	409

---

<b>Choosing Your People</b>	<b>410</b>
Project Manager	410
Account Manager	411
Lead Architect	411
Software Architects and Engineers	412
Client-Side Developers	412
Information Architects	412
Senior Designers	413
Studio Artists	413
Doubling Up of Roles	413
Working Practices	413
Your Client's Role	414
<b>Summary</b>	<b>414</b>
<b>Chapter 17: Project Planning</b>	<b>415</b>

---

<b>Choosing a Process</b>	<b>415</b>
The Waterfall Process	416
The Spiral Process	417
Making the Decision	419
<b>Common Process Practices</b>	<b>419</b>
Specification Phase	419
Design Phase	422
Build Phase	424
Testing Phase	424
Handover	425
<b>Programming Methodologies and Practices</b>	<b>425</b>
Test-Driven Development	425
Extreme Programming	426
<b>Change Management</b>	<b>428</b>
Specification Revisions	428
Specification Changes That Arise After Sign-Off	429
Disputes Arising from Differences in Interpretation	429
Bugs Reported by the Client	429
<b>Summary</b>	<b>430</b>
<b>Chapter 18: Systems Architecture</b>	<b>431</b>

---

<b>What Is Systems Architecture?</b>	<b>431</b>
Why Is It Important?	432
What Needs to be done?	432

# Contents

---

<b>Determining Business Requirements</b>	<b>433</b>
<b>Determining Hardware Requirements</b>	<b>434</b>
Web Servers	434
Database Servers	434
Load Balancing	435
Firewall	436
<b>Understanding the Physical Environment and Connectivity</b>	<b>436</b>
Calculating CIR	437
Calculating Monthly Transfer	438
Rack Space Considerations	438
Network Considerations	438
Storage Considerations	439
<b>Managing Maintenance</b>	<b>439</b>
<b>Summary</b>	<b>439</b>
<b>Chapter 19: Building the Application</b>	<b>441</b>
<hr/>	
<b>Starting the Project: Monday</b>	<b>441</b>
Getting an Earful	442
Story Weight Estimation	444
Release Planning	450
<b>Starting the Work</b>	<b>451</b>
Outlining the Details of Story 9	451
Writing Tests	452
PHPUnit	453
Creating the Login Screen	460
The Next Story	464
Re-estimating	474
<b>Cleaning Up</b>	<b>475</b>
Refactoring Code	476
<b>Finishing the Iteration</b>	<b>483</b>
Story 14: Changing the Week Recalls a Previous Week	483
Story 15: Per-Week Items on the Customer Contact Report	484
<b>Travel Expense Report</b>	<b>492</b>
Travel Expense Item	494
Travel Expense Week	496
The Spike	499
More Travel Expense Week Tests	500
Satisfying the Travel Expense Week Tests	502
<b>The Finalized Travel Expense Report</b>	<b>514</b>
<b>Mock Objects</b>	<b>529</b>
<b>Summary</b>	<b>533</b>

---

<b>Chapter 20: Quality Assurance and Testing</b>	<b>535</b>
<b>Introduction to QA</b>	<b>535</b>
Why You Should Aim High	536
What Is Quality?	537
Measurable and Quantifiable Quality	538
<b>Testing</b>	<b>540</b>
Unit Testing	540
Functional Testing	541
Load Testing	542
Usability Testing	543
Fault Tracking	543
Effective Fault Management Using Mantis	544
Getting the Most out of Mantis	554
<b>Summary</b>	<b>554</b>
<b>Chapter 21: Deployment</b>	<b>555</b>
<b>Devising Development Environments</b>	<b>555</b>
The Studio Development Environment	556
The Studio Staging Environment	557
The Live Staging Environment	557
The Live Production Environment	559
Development Databases	559
<b>The Deployment Workflow</b>	<b>560</b>
Automated Version Control Repository Extraction	562
Using rsync	564
Keeping Servers in Sync with rsync	566
A Word About FTP	567
<b>Summary</b>	<b>567</b>
<b>Part IV: Pushing PHP to Its Limits</b>	
<b>Chapter 22: Content Management in the PHP World</b>	<b>571</b>
<b>What Is Content Management?</b>	<b>571</b>
<b>A Brief History of Content Management</b>	<b>573</b>
<b>Examples of PHP CMSs</b>	<b>575</b>
Drupal	575
Joomla!	578
ExpressionEngine	580
Frog CMS	583

# Contents

---

<b>Building Your Own System</b>	<b>585</b>
Content-Related Issues	586
Administrators and Privileges	588
Workflow	589
Templates	590
<b>Summary</b>	<b>591</b>
<b>Chapter 23: Handling High Traffic and High Availability</b>	<b>593</b>
<hr/>	
<b>Understanding the Terms</b>	<b>593</b>
High Availability	594
High Traffic	594
<b>Platforms</b>	<b>595</b>
Exterior Network	595
Interior Network	596
Hardware	597
Operating System	598
Web Server	598
Application Server	599
Database	599
Software Architecture	600
<b>Impact Factors</b>	<b>601</b>
Server Load	601
Component Failure	602
Network Load	602
Network Failure	602
<b>Mitigation Techniques</b>	<b>603</b>
Load Balancing	603
Geographic Balancing	605
NIC Teaming	606
Disk Fault Tolerance	606
Power Redundancy	607
Multi-segment Topologies	608
Database Clustering and Replication	610
Traffic Calculus	611
Caching	613
Content Generation	614
Asset Servers	615
Content Delivery Networks	615
<b>Real-World Examples</b>	<b>615</b>
Flickr	616
Wikipedia	616
<b>Summary</b>	<b>618</b>

---

<b>Chapter 24: PHP Advocacy</b>	<b>619</b>
<b>Low Overhead</b>	<b>619</b>
<b>Linguistic Simplicity</b>	<b>621</b>
Weakly Typed Languages	621
Reduced Punctuation	621
Commonly Understood Syntax	622
<b>Cross-Platform Support</b>	<b>622</b>
<b>Ubiquity</b>	<b>624</b>
<b>Power</b>	<b>624</b>
<b>Summary</b>	<b>625</b>
<b>Chapter 25: Your Career As a PHP Professional</b>	<b>627</b>
<b>Motivation</b>	<b>627</b>
<b>Your Career As a Developer</b>	<b>628</b>
More Than Web Development Skills	628
Soft Skills	629
Academic Skills	630
Community Skills	630
<b>Summary</b>	<b>631</b>
<b>Part V: Appendixes</b>	
<b>Appendix A: Version Control</b>	<b>635</b>
<b>Appendix B: PHP IDEs</b>	<b>647</b>
<b>Appendix C: Performance Tuning PHP</b>	<b>659</b>
<b>Appendix D: Best Practice PHP Installation</b>	<b>671</b>
<b>Index</b>	<b>681</b>

---