A Web-Based Introduction to Programming

A Web-Based Introduction to Programming

Essential Algorithms, Syntax, and Control Structures Using PHP, HTML, and MariaDB/MySQL

Fourth Edition

Mike O'Kane

Carolina Academic Press

Durham, North Carolina

Library of Congress Cataloging-in-Publication Data

Names: O'Kane, Mike, 1953- author.
Title: A web-based introduction to programming : essential algorithms, syntax, and control structures using PHP, HTML, and MariaDB/MySQL / Mike O'Kane.
Description: Durham, North Carolina : Carolina Academic Press, [2017] | Includes bibliographical references and index.
Identifiers: LCCN 2017017694 | ISBN 9781531002749 (alk. paper)
Subjects: LCSH: Computer software--Development. | Internet programming. | Computer programming--Web-based instruction. | PHP (Computer program language) | XHTML (Document markup language)
Classification: LCC QA76.76.D47 O43 2017 | DDC 005.3--dc23
LC record available at https://lccn.loc.gov/2017017694

eISBN 978-1-53100-707-2

Carolina Academic Press, LLC
700 Kent Street
Durham, North Carolina 27701
Telephone (919) 489-7486
Fax (919) 493-5668

www.cap-press.com

Printed in the United States of America.

To my dear mother and father,
thank you for the love and light that you bestowed on us.

Contents

Preface xvii
Acknowledgments xxv
About the Author xxvii

Chapter 1 · Introducing Computer Programming 3
Introduction 3
What Is a Computer Program? 4
What Do Programmers Do? 5
The Software Development Life Cycle 9
The Importance of Writing and Communicating 9
What Are Programming Languages? 10
Compilers and Interpreters 11
So Many Languages! 12
Standalone and Network Applications 12
Markup Languages 13
Combining Markup and Programming Languages 13
Summary 14
Chapter 1 Review Questions 15

Chapter 2 · Client/Server Applications—Getting Started 19
Introduction 19
Client/Server Design in Web Applications 20
Working with Files and Folders 21
Locating Files and Folders on Computers Running a Windows Operating System 22
Locating Files and Folders on the Internet 24
Internet Naming Conventions for Files and Folders 25
Working with a Local Web Server 26
What Languages Will I Use? 27
What Software Will I Need? 28
Installing a Text Editor 29
Installing One or More Web Browsers 29
Installing Your Web Server 30

Using Your Web Server 30
Using URLs with Your Web Server 31
Always Use URLs to Run Your Web Applications! 33
Where to Save Your Work Files 35
The Importance of Frequent Backups 35
Creating an HTML Document 36
Creating a PHP program 38
Creating an Interactive HTML and PHP Program 39
Summary 43
Chapter 2 Review Questions 44
Chapter 2 Code Exercises 47

Chapter 3 · Program Design — From Requirements to Algorithms 49
Introduction 49
What Are Instructions? 50
Common Characteristics of Instructions 50
Sequence, Selection and Repetition Structures 54
A Programming Example 55
Creating an Input, Processing, Output (IPO) chart 56
Designing the User Interface 56
Developing an Algorithm 57
A Smoking Calculator 61
Coding the Application 62
Summary 62
Chapter 3 Review Questions 64
Chapter 3 Code Exercises 69

Chapter 4 · Basics of Markup — Creating a User Interface with HTML 73
Introduction 73
A Short History of HTML 75
Introducing HTML Tags 76
Ignoring White Space 79
More HTML Tags 79
Introducing HTML Tables 82
Using HTML Tables to Layout Web Pages 86
Other HTML Tags 87
Deprecated HTML Tags 87
Introducing Style Sheets 88
Multiple Styles for a Single Tag 89
Selecting Colors for Fonts and Backgrounds 90
Referencing a Style Sheet in Your HTML Document 91
Applying a Style Sheet to Multiple Pages 91
Interactive User Interfaces 92
Creating HTML Forms 92
Using HTML Forms to Obtain User Input 95
Using HTML Tables to Line Up Prompt and Input Boxes 98

Problems with Form Submission 99
Drop Down Lists 99
Combining Textboxes and Drop Down Lists 101
Other Types of Input 103
Stylesheets and Forms 103
Summary 103
Chapter 4 Review Questions 105
Chapter 4 Code Exercises 109

Chapter 5 · Creating a Working Program — Basics of PHP 115
Introduction 115
Why PHP? 116
Working with HTML and PHP 116
Important Features of Client/Server Programs 121
Receiving Input from a Form — wage2.php 122
Processing the Smoking Survey — smoking.php 126
PHP — General Guidelines and Syntax 129
Arithmetic Expressions 133
Using Arithmetic Functions 134
White Space in PHP Files 137
Generating Character Strings from PHP 137
Including Double Quotes in Character Strings 138
Using Multiple PHP Sections 139
Using the number_format() Function to Display Numbers to a Specific Number of Places 140
Including Calls to PHP Functions inside PHP Print Statements 140
String Concatenation and the Concatenation Operator 141
The PHP Echo Statement 143
Finding Syntax Errors 143
Finding Logical Errors 143
Summary 144
Chapter 5 Review Questions 145
Chapter 5 Code Exercises 150

Chapter 6 · Persistence — Saving and Retrieving Data 153
Introduction 153
The Difference Between Persistent and Transient Data 154
Files and Databases 156
Working with a Text File 157
Closing a Text File 158
Reading Data from a Text File 159
PHP Functions to Read Data from a Text File 160
Writing Data to a Text File 163
PHP Functions to Write Data to a Text File 164
Be Careful to Avoid Security Holes! 168

Using Escape Characters 168
Escape Characters and HTML Tags 169
Using PHP to Append Data to Files 170
PHP Functions to Append Data to a Text File 171
Processing Files that Contain Complete Records on Each Line 173
PHP Functions to Parse a Delimited Character String 174
Processing a File with Multiple Records 177
Appending Records to a File 179
Working with Multiple Files 183
Summary 184
Chapter 6 Review Questions 185
Chapter 6 Code Exercises 190

Chapter 7 · Programs that Choose — Introducing Selection Structures 195
Introduction 195
Introducing IF and IF..ELSE Structures 198
Introducing Flow Charts 199
Boolean Expressions and Relational Operators 202
Selection Using the IF Structure 203
Testing Threshold Values 206
Selection Using the IF..ELSE Structure 207
When to Use Braces in IF..ELSE Statements 211
Creating a Program with Multiple but Independent Selection Structures 212
Comparing Strings — Testing for a Correct Password 214
Ignoring the Case of a Character String 218
Providing a Selective Response 219
Using Selection to Construct a Line of Output 221
Summary 224
Chapter 7 Review Questions 225
Chapter 7 Code Exercises 232

Chapter 8 · Multiple Selection, Nesting, ANDs and ORs 237
Introduction 237
Introducing the Logical Operators AND and OR 238
Introducing the NOT Operator 241
Validating User Input 242
Using A Nested Selection Structure to Validate Input 243
Designing Applications with Nested Selection Structures 247
Use of Braces in Nested Selection Structures 248
Chaining Related Selection Structures 249
Additional Input Validation Using Chained Selection Structures 252
More about Input Validation: Using the trim() Function 255
When to Use AND or OR? Be Careful with Your Logic! 256
The Challenge of Software Testing 257
A Special Case: The Switch Statement 258

More Examples in the Samples Folder 258
Some Words of Encouragement 259
Summary 260
Chapter 8 Review Questions 261
Chapter 8 Code Exercises 267

Chapter 9 · Programs that Count — Harnessing the Power of Repetition 271
Introduction 271
Controlling a Loop by Counting 273
Coding a FOR Loop in PHP 273
General Syntax of a FOR Loop 276
Including the Counting Variable in Your Loop Statements 277
Using a Variable to Control the Loop Condition 278
Converting from Celsius to Fahrenheit 280
Changing the Increment Value 282
Using Loops with HTML Tables 284
Allowing the User to Control the Loop 285
Improving Processing Efficiency 288
Using Loops to "Crunch Numbers" 289
Using a Loop to Accumulate a Total 289
Finding the Total and Average from a File of Numbers 291
Finding the Highest and Lowest Values in a Series 293
Performing Multiple Operations on a File of Numbers 294
Nesting IF..ELSE Structures to Customize Output from a Loop 296
Loops within Loops — Creating a Bar Chart 300
Selecting from a List of Data Files 304
Summary 305
Chapter 9 Review Questions 306
Chapter 9 Code Exercises 312

Chapter 10 · "While NOT End-Of-File" — Introducing Event-Controlled Loops 317
Introduction 317
Characteristics of WHILE Loops 318
The Structure of WHILE Loops 321
An Algorithm to Process Files of Unknown Length 321
Using a WHILE Loop to Process a File of Scores 324
Including Selection Structures Inside a WHILE Loop 329
Using a WHILE Loop to Count, Sum and Average Data 331
Using a WHILE Loop to Process a File of Records 334
Processing Weekly Wages from a File of Timesheet Records 336
Processing Selected Records from a File of Timesheet Records 338
Processing Selected Fields from a File of Records 341
Processing a File of Survey Data 344
Using DO..WHILE or REPEAT..UNTIL Loops 348

Summary 348
Chapter 10 Review Questions 349
Chapter 10 Code Exercises 357

Chapter 11 · Structured Data — Working with Arrays 361
Introduction 361
What Is an Array? 362
Working with Array Elements 363
Extending an Array 364
Displaying Array Values 364
Receiving Scores into an Array from an HTML Form 365
Arrays of Strings 368
How Large Is the Array? 369
Why Do Array Indices Begin with 0 and Not 1? 369
Using FOR Loops with Arrays 370
Using the sizeof() Function to Control a FOR Loop 371
Summing and Averaging the Values in an Array 371
Counting Selected Values in an Array 372
Multiple Operations on an Array 372
Reading Data from a File into an Array 374
Reading Data into an Array from a File of Unknown Length 377
Using [] with no Index Value 378
Reading Selected Data from a File into an Array 379
Reading Data from a File into Multiple Arrays 380
Reading Selected Data from a File of Records into an Array 381
More About the explode() and list() Functions 382
A Special Loop for Processing Arrays — FOREACH 384
Multi-Dimensional Arrays 385
Summary 385
Chapter 11 Review Questions 387
Chapter 11 Code Exercises 392

Chapter 12 · Associative Arrays 397
Introduction 397
Using a Variable to Reference the Key of an Associative Array 398
Using Associative Arrays as Lookups 399
Using the array() Function to Create Associative Arrays 400
Associative Arrays and the FOREACH Loop 401
More about the $_POST Array 402
Using the isset() Function to Combine a Web Form with the Form Processing Code in a Single Page 404
Web Sessions and the $_SESSION Array 406
Adding Code to Manage a Web Session 408
Creating, Initializing and Modifying Session Variables 409
Validating $_SESSION and $_POST Arrays 412

Revisiting the Same Page in a Web Session 414
Summary 418
Chapter 12 Review Questions 419
Chapter 12 Code Exercises 421

Chapter 13 · Program Modularity — Working with Functions 425
Introduction 425
Using Functions 427
Understanding Function Arguments 428
Receiving Values from a Function 429
Researching Available Functions 430
Reasons to Use Pre-Defined Functions 430
Using die() or exit() to Terminate an Application 431
Creating Your Own Functions 432
Where Do I Put My Functions? 435
Creating a Library of Functions 437
Including Functions from External Files 438
Using the Same Functions in Different Programs 439
Functions Calling Functions 442
Learning to Think Beyond Specific Applications 445
More about Include Files 449
Summary 450
Chapter 13 Review Questions 452
Chapter 13 Code Exercises 457

Chapter 14 · Connecting to a Database — Working with MySQL 461
Introduction 462
What Is a Relational Database? 462
The Relational Database Management System (RDBMS) 462
Structured Query Language — MySQL 463
Starting Your MySQL Server 464
Configuring MySQL for Use with This Textbook 464
Three Ways to Work with MySQL 465
Working with PHP and MySQL 465
Using PHP to Open and Close a Connection to a MySQL Server 465
Using the MySQL SELECT Query 467
Selecting Specific Records 468
Relational Operators in MySQL 469
The Logical Operators AND and OR 470
Ordering Your Query Results 471
Viewing Your Query Results 472
Using an HTML Table to Display the Query Results 474
Putting It All Together 474
Using Input from an HTML Form to Construct a Query 476
Processing Queries with a Single Result 476

Performing Calculations with the Result Set 477
Performing Aggregate Operations on MySQL Queries 478
Performing JOIN Operations on Multiple Tables 479
Using INSERT to Add Records to a Table 481
Using UPDATE to Modify a Record 481
Removing a Record 482
Storing MySQL Connection Data in an Include File 483
Creating, Dropping, and Altering Databases and Tables 484
Summary 484
Chapter 14 Review Questions 487
Chapter 14 Code Exercises 491

Chapter 15 · Introduction to Object-Oriented Programming 493
Introduction 494
What is an Object? 495
Creating and Using Instances of a Class 496
Using Employee Objects in an Application 497
Defining an Object 500
Coding the Object Class 501
Creating and Using Instances of an Object Class 506
The Class Constructor Method 506
Method Overloading 507
Why do Objects Matter? 508
Object Design and Inheritance 509
Abstract Classes and Methods 517
Method Over-riding 518
Polymorphism 518
OOP and Databases 518
OOP Development 519
OOP Languages 520
Summary 520
Chapter 15 Review Questions 521
Chapter 15 Code Exercises 524

Chapter 16 · Where to Go from Here . . . 527
Introduction 527
Moving Forward with PHP and HTML 528
More about PHP 530
PHP and Other Languages 530
XHTML and XML 532
Client Side Processing with Javascript and Ajax 533
The Importance of OOP 534
IDE's, Modeling Languages and Frameworks 534
Client/Server and Server/Server Programming 535
Mobile Applications 535

Digital Media 536
Graphical User Interfaces and Interface Design 536
Web Design and Content Management 536
Database Programming and SQL 536
In Summary: Follow Your Heart! 537

Appendix A · Data Representation and Formats 539
Introduction 539
Storing Data in Bits and Bytes 539
How Multimedia Data Is Represented in Binary 540
How Numeric Values Are Represented in Binary 541
How Plain Text Is Represented in Binary 542
How Source Code and Markup Code Is Represented in Binary 543
How Program Instructions Are Represented in Binary 544
How Memory Addresses Are Represented in Binary 544
What Else Can Be Represented in Binary? 544

Appendix B · Files, Folders, Addressing Schemes, and Command Line Arguments 545
File Types and File Extensions 545
Disk and Disk Drives 546
Files and File Folders (Directories) 547
Naming Files and Folders 548
File Addresses in Windows and on the Web 548
Relative Addresses in Windows 549
Relative Addresses on the Internet 550
Using Relative Web Addresses in HTML Code 550
Managing Files at the Command Line 551
Introduction to MS DOS Commands 551
Recalling Previous Commands 556
Use Double Quotes when Paths Include Spaces 556
Printing the Contents of the Console Window 556
Creating Batch Files 556
Unix Commands 557

Appendix C · Installing and Running Your Standalone Web Server 559
Using an Active Web Server 559
Problems Using Your Web Server 559
Advanced Users 560

Appendix D · Debugging Your Code 561
Problems Viewing Your HTML or PHP Programs 561
Problems with HTML Layout 563
Locating PHP Syntax Errors 564
Common PHP Syntax Errors 564

Common Logical Errors 566

Appendix E · More about HTML and CSS 569
Useful HTML References 569
Useful CSS References 569
Inline Styles and Internal Style Sheets 570
Deprecated HTML Tags 570
Frequently Asked Questions Regarding HTML Tags 571

Appendix F · More about PHP Functions and Data Types 575
Useful PHP References 575
More about PHP Functions and Data Types 575
Standard PHP Array Functions 576
Standard PHP File Functions 577
Standard PHP Math Functions 578
Standard PHP String Functions 579
PHP Data Types 580

Appendix G · Additional PHP Operators and Control Structures 581
Shortcut Operators 581
Switch Structure 581
Another Loop Structure: DO..WHILE 583
Multi-Dimensional Arrays 584
Ragged Arrays 586
Multi-Dimensional Associative Arrays 586

Index 589

Preface

The problem I have tried to solve with this textbook is, quite simply, how to effectively introduce general programming concepts to students who have never programmed before. Perhaps like me, you have found yourself frustrated by textbooks that try to cover too much too fast, make inappropriate assumptions about what a student already knows, or take sudden leaps in complexity when providing examples and exercises.

I believe that the purpose of an introductory programming course is to help students gain confidence and develop their understanding of basic logic, syntax, and problem-solving. They do not need to learn all aspects of a language or even learn best practices—these are topics for the next course level. The question is: how to provide the kind of hands-on experience that supports active learning without overwhelming the beginning student with too much syntactical and programmatic detail?

I have tried many approaches over the years before settling on a Web-based approach, using PHP, CSS, and HTML code to develop small, interactive Web applications. This approach has proved very successful. Many students report how much they enjoy the course, how much they have learned, and how well the material has served them in subsequent courses and in their professional life. I also hear from many students who tell me that the course positively changed their opinion of programming as a career or subject of interest, which is most gratifying.

Some instructors may have concerns that my coverage of the PHP and HTML is insufficient. The book uses a small number of HTML tags, CSS rules, and PHP functions, and employs some arbitrary conventions to simplify the code and keep the focus on basic concepts common to most languages. For example, PHP **print** statements are used rather than **echo** statements, and these statements **always** include parentheses and double quotes so that the syntax is more consistent with the output statements of most other languages. The last chapter ("Where to Go From Here") clarifies which practices are standard and which are particular to the textbook.

Intended Audience

The book is designed to serve:

- Instructors teaching introductory programming, programming logic and design, or Web programming courses, who want a textbook that engages students and provides a solid preparation for subsequent courses, but avoids overwhelming beginners with too much syntactical detail or program complexity.
- Traditional and online students taking a first course in programming, programming logic and design, or Web programming.
- Web designers, graphic artists, technical communicators, and others who find that their work increasingly requires some degree of programming expertise, and need an effective, hands-on introduction.
- Others who wish to learn the basics of programming, either for personal interest, or to explore the possibility of a career in this field.

Note that solutions to quizzes and exercises are only available to verified course instructors.

Approach

The book takes a fairly novel approach, allowing students to learn program logic and design by developing a large number of small Web-based applications. Students love working with the Web, and this approach has other important benefits:

- Important concepts such as client/server design, server-side processing, and interface-driven code modules can be introduced in the form of working applications, and then applied in hands-on exercises.
- Students not only learn the essential control structures and syntax of a programming language, but also learn to use a markup language (and associated style sheets), and a database query language to access and query a database. This makes sense in today's programming environment where these languages are routinely used in combination to develop a networked application.
- The material is relevant to students across a range of disciplines: Computer Science, Information Systems, Technical Communications, Network Systems, Digital Media, Web Technologies, Mobile Applications, Database Programming, and other technology-related fields.
- The focus on hands-on problem-solving and fundamental structures prepare students for next-level, language-specific courses such as PHP, Python, Java or C++, as well as Web design and database courses, without replicating a great deal of material, while the syntax covered here is generally consistent with these and other languages.

The book makes use of a programming language (PHP), a scripting language (HTML), a style sheet language (CSS), and a database query language (MariaDB or MySQL),

but does not attempt to provide a complete overview of these languages. Instead, students learn sufficient syntax to convert requirements into working applications using basic programming structures, arithmetic and logical expressions, user interfaces, functions, data files, and SQL queries. The focus remains on basic concepts, logic and design, algorithm development, and common programming procedures. The book provides context throughout, explaining why each topic is important, and referring students to related career paths.

Although the book focuses on Web-based applications, there is NO requirement for a network-based programming environment. The book uses a fully functional but standalone Apache Web server (the open source xampp distribution provided by the Apache Friends group) that students can install on a USB drive or home computer simply by unzipping a file. Students can begin programming in HTML, PHP and MariaDB or MySQL in literally minutes.

Features

Each chapter begins with clearly stated learning outcomes. Each topic is introduced using examples of simple program requirements that are first developed as algorithms and interfaces and then realized in working code. Code statements and control structures are explained step by step.

Different programming topics are treated in separate chapters. Even topics that are commonly combined, such as counting loops and event-controlled loops, have their own chapters so that students have the chance to develop and apply their understanding of each separately.

Each chapter includes quizzes that have been carefully developed to test the student's understanding of the chapter's learning outcomes. The questions have been tested extensively in the classroom.

Three different types of coding exercise are provided at the end of each chapter:

- **Fixit** exercises provide small programs that include a single error of some kind. These exercises help students improve their problem-solving ability, test their understanding of key concepts, and develop tracing and debugging skills.
- **Modify** exercises provide working programs that must be modified to perform a somewhat different or additional function. These exercises help students determine how and where to add new code, and test their ability to read and understand existing code.
- **Code completion** exercises allow students to apply concepts and tools covered in the chapter by developing new applications. These exercises test the student's ability to: understand requirements, develop algorithms, and produce working code. The code completion exercises follow consistent themes that are developed throughout the book, so that students can more readily appreciate the value of new functionalities that they learn in each chapter.

Templates for each exercise contain partially completed code so students don't waste time typing (and debugging) code that is not relevant to the problem at hand. The templates also help instructors to streamline the grading process.

The textbook comes with a standalone Web server that can be installed on a fixed or portable drive simply by unzipping a file (so students can bring the software with them to work on computers at any location).

The server installation includes textbook folders that contain all code samples and exercise templates. Students can complete the exercises simply by opening, editing, and saving the appropriate files. Assignments can be turned in simply by zipping and submitting the appropriate chapter folder.

The textbook appendices provide additional learning resources designed to: (a) help individual students with particular needs or interests (for example file/folder management, additional references, and help debugging code); and (b) deliver useful topics not included in the chapters (for example data representation, additional control structures, and multi-dimensional arrays).

Textbook Web Site

The textbook Web site ensures that both students and instructors have access to the most current resources associated with this textbook. The Web site includes: everything you need to install and use the Web server; slide presentations; and hints and help for students working through each chapter. The Web site also provides support for verified instructors, including additional exercises, test banks, slide presentations, quiz solutions, code solutions, and other instructional resources. The Web site can be found at:

http://www.mikeokane.com/textbooks/WebTech/

Changes to the Fourth Edition

In addition to minor corrections and improvements, this fourth edition of the book includes: a new install of the xampp Apache Web server distribution with installation instructions for Windows, Mac OS, and Linux; revised file naming conventions that are more standard for current web development; a hopefully improved redesign of Chapters 7 and 8; additional materials and improvements to Chapter 13 (functions); references to both MySQL and MariaDB in Chapter 14 (the actual code and descriptions are identical); a new Chapter 15 that introduces Object-Oriented Programming.

Chapter Overview

Chapter 1: Introducing Computer Programming. Students learn the relationship between machine language and high-level languages, and review common tasks that computer programs typically perform. The work of a programmer is described, and the software development cycle is explained. The chapter highlights and briefly summarizes design approaches such as algorithm development, interface design, client/server design and object-oriented programming. Different programming languages are identified, and the distinction is made between interpreted and compiled languages, and between markup and programming languages. Standalone and network applications are also contrasted.

Chapter 2: Client/Server Applications — Getting Started. This chapter prepares students for the hands-on work they will perform in subsequent chapters. File types and local and Internet addressing schemes are explained. Instructions are provided to install, run, and test the required software. Students are shown how to create, store, and run a number of sample applications in order to become familiar with the process of using a text editor, saving files, running the Web server, and viewing the results in a Web browser.

Chapter 3: Program Design — from Requirements to Algorithms. The general characteristics and requirements of effective instructions are explored, using human and program examples. Students walk through the process of reviewing simple requirements, creating input, processing, and output (IPO) charts, designing the interface, and developing solution algorithms. The chapter introduces sequence, selection and control structures, variables and assignment operations, and arithmetic and logical expressions.

Chapter 4: Basics of Markup — Creating a User Interface with HTML. This chapter explains the significance of data rendering, and provides a brief overview and history of Hypertext Markup Language (HTML). Commonly used HTML tags are explained, and the student is shown how to apply these to create and organize simple Web pages. Cascading style sheets are introduced. Students are shown how to create HTML forms to obtain user input as a first step in developing interactive Web applications. HTML Tables are used to perform simple form layout.

Chapter 5: Creating a Working Program — Basics of PHP. This chapter teaches sufficient PHP language syntax to process user input received from HTML forms, perform simple arithmetic, and produce formatted output. In the process, students learn to code arithmetic expressions, use standard operators and functions, create and work with variables, and identify and fix both syntax and logical errors.

Chapter 6: Persistence — Saving and Retrieving Data. This chapter explains the difference between persistent and transient data, and introduces text file processing as well as basic database concepts. Students learn to: open, read, write, and close text files; work with multiple files; parse lines of data that contain multiple values separated by some kind of delimiter.

Chapter 7: Programs that Choose — Introducing Selection Structures. This chapter introduces selection control structures and demonstrates the use of algorithms to solve problems requiring simple selection. Students learn to use IF and IF..ELSE structures, Boolean expressions, relational operators, truth tables, simple string comparisons, and testing procedures.

Chapter 8: Multiple Selection, Nesting, ANDs and ORs. This chapter develops examples from Chapter 7 to handle problems associated with input validation and more complex requirements. Students explore the use of compound Boolean expressions, nested selection structures, chained IF..ELSEIF..ELSE selection structures, and multiple but independent selection structures.

Chapter 9: Programs that Count — Harnessing the Power of Repetition. This chapter introduces loop structures with a focus on count-controlled FOR loops. Students learn how to refer to the counting variable within the loop, and how to use loops to generate tables, crunch numbers, accumulate totals, find highest and lowest values in a series, select values from a file of records, and display bar charts.

Chapter 10: "While NOT End-Of-File" — Introducing Event-Controlled Loops. This chapter introduces WHILE loops and demonstrates the use of the priming read and the standard algorithm to process files of unknown length. The student is shown how WHILE loops can be used to perform various operations on a list of data values, and how a file of records can be processed and searched for specific records or field values.

Chapter 11: Structured Data — Working with Arrays. This chapter introduces numerically-indexed arrays, and shows how arrays can be used to store, access, and update multiple-related values. The use of the FOR loop to process arrays is explained, and various array-processing algorithms are demonstrated.

Chapter 12: Associative Arrays. This chapter introduces associative arrays. Students learn how to use associative arrays as lookups, and gain a better understanding of the $_POST array and the way that data is received from HTML forms. Web sessions are introduced, and students learn how to use the $_SESSION array to maintain session data between applications.

Chapter 13: Program Modularity — Working with Functions. This chapter demonstrates the importance of program modularity and introduces functions, include files and objects. Students learn to write their own functions, to build libraries of related functions, and to call functions from different applications as needed.

Chapter 14: Connecting to a Database — Working with MySQL. This chapter introduces databases queries as an important application tool. The relationship between relational databases and SQL is explained, along with the purpose and syntax of common queries (SELECT, INSERT, UPDATE and DELETE). Students learn to write code to open and close database connections, submit queries, handle errors, perform simple joins, and process results.

Chapter 15: Introduction to Object-Oriented Programming. This chapter introduces Object-Oriented Programming. Examples show how simple object classes are designed, how class variables are encapsulated and accessed by class methods, how ob-

jects are instantiated and used in applications, and how classes can be inherited by other classes. An overview of basic OO terminology is provided.

Chapter 16: Where to Go From Here. This last chapter provides a short overview of key concepts and technologies that the students may want to explore after completing this textbook, along with clarification of some of the conventions followed in the book.

The textbook also includes a number of useful appendices as follows:

Appendix A introduces data representation, and shows how binary values can store data for a wide range of purposes.

Appendix B provides an introduction to overview of file and folder management, file addressing schemes (including relative and absolute addresses), and the use of the command line with a list of common DOS and Unix command equivalents.

Appendix C provides help for students wishing to use different Web server installations.

Appendix D provides debugging help for students having trouble identifying and resolving PHP code errors.

Appendix E provides additional material and references for students wishing to learn more about HTML and style sheets.

Appendix F provides additional information regarding PHP data types, and provides a list of common PHP functions not covered in the book.

Appendix G provides additional coverage of common PHP operators and structures that were omitted from the chapters to avoid overwhelming the beginning student (for example, shortcut operators, the SWITCH statement, DO..WHILE loops, and multidimensional arrays).

Acknowledgments

This textbook could not have been created without the generous help and support of many others. In particular I want to thank my dear wife Constance Humphries for her invaluable technical advice, proof-reading, development of video tutorials, and daily encouragement and patience! My sincere thanks to Scott Sipe, Beth Hall, Sara Hjelt, and all at Carolina Academic Press for their supportive style, professionalism and experience. Thanks to all my fellow instructors at A-B Tech (Asheville-Buncombe Technical Community College), especially to Charlie Wallin and Fred Smartt who field-tested the first edition, and provided invaluable suggestions and corrections. And thanks to all of those students who have learned with me and sometimes in spite of me as this book evolved in the classroom. A particular thank you to A-B Tech students Uma Benson, Jean-Jacques Maury, and Kenneth Stanley, who all voluntarily provided me with carefully compiled lists of corrections that were incorporated into the fourth edition. Their engagement with the material and concern for future students is greatly appreciated. Any remaining errors or inconsistencies are of course my own.

Lastly, a huge thank you to Kai 'Oswald' Seidler, Kay Vogelgesang, and all those who have contributed to the Apache Friends Project, and who continue to deliver and support the XAMPP distribution. So many of us owe you our great appreciation for your generosity of spirit!

About the Author

Mike O'Kane holds a master's degree in Systems Science (specializing in Advanced Technology) from Binghamton University. He has over eighteen years' experience teaching computer science courses, most recently at Asheville-Buncombe Technical Community College in North Carolina. He also has extensive practical experience in the use of technology for learning, having worked at IBM as a short-course developer, NC State University as an Instructional Coordinator, and the University of North Carolina system as the first Executive Director of the UNC Teaching and Learning with Technology Collaborative. He has a passion for developing effective instructional content, and learning environments that promote rather than hinder student learning.

A Web-Based Introduction to Programming