A Web-Based Introduction to Programming

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

Third Edition

Mike O’Kane

CAROLINA ACADEMIC PRESS
Durham, North Carolina
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
   Working with a Local Web Server 25
   What Languages Will I Use? 26
   What Software Will I Need? 27
   Installing a Text Editor 27
   Installing One or More Web Browsers 28
   Installing Your Web Server 28
## Contents

Using Your Web Server.......................... 29
Using URL's with Your Web Server............. 30
Always Use URL's to Run Your Web Applications! 32
Where to Save Your Work Files..................... 34
The Importance of Frequent Backups............... 34
Creating an HTML Document......................... 34
Creating a PHP program................................ 36
Creating an Interactive HTML and PHP Program...... 38
Summary.............................................. 41
Chapter 2 Review Questions......................... 43
Chapter 2 Code Exercises............................ 46

### Chapter 3 • Program Design — From Requirements to Algorithms

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>49</td>
</tr>
<tr>
<td>What Are Instructions?</td>
<td>50</td>
</tr>
<tr>
<td>Common Characteristics of Instructions</td>
<td>50</td>
</tr>
<tr>
<td>Sequence, Selection and Repetition Structures</td>
<td>54</td>
</tr>
<tr>
<td>A Programming Example</td>
<td>55</td>
</tr>
<tr>
<td>Creating an Input, Processing, Output (IPO) chart</td>
<td>56</td>
</tr>
<tr>
<td>Designing the User Interface</td>
<td>56</td>
</tr>
<tr>
<td>Developing an Algorithm</td>
<td>57</td>
</tr>
<tr>
<td>A Smoking Calculator</td>
<td>60</td>
</tr>
<tr>
<td>Coding the Application</td>
<td>62</td>
</tr>
<tr>
<td>Summary</td>
<td>62</td>
</tr>
<tr>
<td>Chapter 3 Review Questions</td>
<td>64</td>
</tr>
<tr>
<td>Chapter 3 Code Exercises</td>
<td>69</td>
</tr>
</tbody>
</table>

### Chapter 4 • Basics of Markup — Creating a User Interface with HTML

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>73</td>
</tr>
<tr>
<td>A Short History of HTML</td>
<td>75</td>
</tr>
<tr>
<td>Introducing HTML Tags</td>
<td>76</td>
</tr>
<tr>
<td>Ignoring White Space</td>
<td>79</td>
</tr>
<tr>
<td>More HTML Tags</td>
<td>79</td>
</tr>
<tr>
<td>Introducing HTML Tables</td>
<td>82</td>
</tr>
<tr>
<td>Using HTML Tables to Layout Web Pages</td>
<td>86</td>
</tr>
<tr>
<td>Other HTML Tags</td>
<td>87</td>
</tr>
<tr>
<td>Deprecated HTML Tags</td>
<td>87</td>
</tr>
<tr>
<td>Introducing Style Sheets</td>
<td>88</td>
</tr>
<tr>
<td>Multiple Styles for a Single Tag</td>
<td>89</td>
</tr>
<tr>
<td>Selecting Colors for Fonts and Backgrounds</td>
<td>90</td>
</tr>
<tr>
<td>Referencing a Style Sheet in Your HTML Document</td>
<td>91</td>
</tr>
<tr>
<td>Applying a Style Sheet to Multiple Pages</td>
<td>92</td>
</tr>
<tr>
<td>Interactive User Interfaces</td>
<td>92</td>
</tr>
<tr>
<td>Creating HTML Forms</td>
<td>93</td>
</tr>
<tr>
<td>Using HTML Forms to Obtain User Input</td>
<td>95</td>
</tr>
<tr>
<td>Using HTML Tables to Line Up Prompt and Input Boxes</td>
<td>98</td>
</tr>
</tbody>
</table>
Contents

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 117
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 136
Generating HTML Output 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 139
Including Calls to PHP Functions inside PHP print Statements 140
String Concatenation and the Concatenation Operator 141
The PHP echo Statement 142
Finding Syntax Errors 143
Finding Logical Errors 143
Summary 143
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
Chapter 9 • Programs that Count — Harnessing the Power of Repetition

Introduction 273
Controlling a Loop by Counting 275
Coding a FOR Loop in PHP 275
General Syntax of a FOR Loop 278
Including the Counting Variable in Your Loop Statements 279
Using a Variable to Control the Loop Condition 280
Converting from Celsius to Fahrenheit 282
Changing the Increment Value 284
Using Loops with HTML Tables 286
Allowing the User to Control the Loop 287
Improving Processing Efficiency 290
Using Loops to “Crunch Numbers” 291
Using a Loop to Accumulate a Total 291
Finding the Total and Average from a File of Numbers 293
Finding the Highest and Lowest Values in a Series 295
Performing Multiple Operations on a File of Numbers 296
Nesting IF..ELSE Structures to Customize Output from a Loop 298
Loops within Loops — Creating a Bar Chart 302
Selecting from a List of Data Files 306
Summary 307
Chapter 9 Review Questions 308
Chapter 9 Code Exercises 314

Chapter 10 • “While NOT End-Of-File” — Introducing Event-Controlled Loops

Introduction 319
Characteristics of WHILE Loops 320
The Structure of WHILE Loops 323
An Algorithm to Process Files of Unknown Length 323
Using a WHILE Loop to Process a File of Scores 326
Including Selection Structures Inside a WHILE Loop 331
Using a WHILE Loop to Count, Sum and Average Data 333
Using a WHILE Loop to Process a File of Records 336
Processing Weekly Wages from a File of Timesheet Records 338
Processing Selected Records from a File of Timesheet Records 340
Processing Selected Fields from a File of Records 343
Processing a File of Survey Data 346
Using DO..WHILE or REPEAT..UNTIL Loops 350
Summary 350
Chapter 10 Review Questions 351
### Chapter 10 Code Exercises 359

### Chapter 11 • Structured Data — Working with Arrays 363

- **Introduction** 363
- **What Is an Array?** 364
- **Working with Array Elements** 365
- **Extending an Array** 366
- **Displaying Array Values** 366
- **Receiving Scores into an Array from an HTML Form** 367
- **Arrays of Strings** 370
- **How Large Is the Array?** 371
- **Why Do Array Indices Begin with 0 and Not 1?** 371
- **Using FOR Loops with Arrays** 372
- **Using the sizeof() Function to Control a FOR Loop** 373
- **Summing and Averaging the Values in an Array** 373
- **Counting Selected Values in an Array** 374
- **Multiple Operations on an Array** 374
- **Reading Data from a File into an Array** 376
- **Reading Data into an Array from a File of Unknown Length** 379
- **Using [] with no Index Value** 380
- **Reading Selected Data from a File into an Array** 381
- **Reading Data from a File into Multiple Arrays** 382
- **Reading Selected Data from a File of Records into an Array** 383
- **More About the explode() and list() Functions** 384
- **A Special Loop for Processing Arrays — FOREACH** 386
- **Multi-Dimensional Arrays** 387
- **Summary** 387
- **Chapter 11 Review Questions** 389
- **Chapter 11 Code Exercises** 394

### Chapter 12 • Associative Arrays 399

- **Introduction** 399
- **Using a Variable to Reference the Key of an Associative Array** 400
- **Using Associative Arrays as Lookups** 401
- **Using the array() Function to Create Associative Arrays** 402
- **Associative Arrays and the FOREACH Loop** 403
- **More about the $_POST Array** 404
- **Using the isset() Function to Combine a Web Form with the Form Processing Code in a Single Page** 406
- **Web Sessions and the $_SESSION Array** 408
- **Adding Code to Manage a Web Session** 410
- **Creating, Initializing and Modifying Session Variables** 411
- **Validating $_SESSION and $_POST Arrays** 414
- **Revisiting the Same Page in a Web Session** 416
- **Summary** 420
Chapter 13 • Program Modularity — Working with Functions and Objects 427

Introduction 427
Using Functions 429
Understanding Function Arguments 430
Receiving Values from a Function 431
Researching Available Functions 432
Reasons to Use Pre-Defined Functions 433
Using die() or exit() to Terminate an Application 433
Creating Your Own Functions 434
Where Do I Put My Functions? 438
Creating a Library of Functions 439
Including Functions from External Files 440
Using the Same Functions in Different Programs 441
Learning to Think Beyond Specific Applications 444
More about Include Files 448
OPTIONAL: Introducing Object-Oriented Programming (OOP) 449
Defining an Object Class 450
Coding the Object Class 451
Creating and Using Instances of an Object Class 454
The Importance of OOP 457
Important OOP Terms and Concepts 458
Summary 459
Chapter 13 Review Questions 461
Chapter 13 Code Exercises 466

Chapter 14 • Connecting to a Database — Working with MySQL 471

Introduction 472
What Is a Relational Database? 472
The Relational Database Management System (RDBMS) 472
Structured Query Language — MySQL 473
Starting Your MySQL Server 474
Configuring MySQL for Use with This Textbook 474
Three Ways to Work with MySQL 475
Working with PHP and MySQL 475
Using PHP to Open and Close a Connection to a MySQL Server 475
Using the MySQL SELECT Query 477
Selecting Specific Records 478
Relational Operators in MySQL 479
The Logical Operators AND and OR 480
Ordering Your Query Results 481
Viewing Your Query Results 482
Using an HTML Table to Display the Query Results 484
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files and File Folders (Directories)</td>
<td>523</td>
</tr>
<tr>
<td>Naming Files and Folders</td>
<td>524</td>
</tr>
<tr>
<td>File Addresses in Windows and on the Web</td>
<td>524</td>
</tr>
<tr>
<td>Relative Addresses in Windows</td>
<td>525</td>
</tr>
<tr>
<td>Relative Addresses on the Internet</td>
<td>526</td>
</tr>
<tr>
<td>Using Relative Web Addresses in HTML Code</td>
<td>526</td>
</tr>
<tr>
<td>Managing Files at the Command Line</td>
<td>527</td>
</tr>
<tr>
<td>Introduction to MS DOS Commands</td>
<td>527</td>
</tr>
<tr>
<td>Recalling Previous Commands</td>
<td>532</td>
</tr>
<tr>
<td>Use Double Quotes when Paths Include Spaces</td>
<td>532</td>
</tr>
<tr>
<td>Printing the Contents of the Console Window</td>
<td>532</td>
</tr>
<tr>
<td>Creating Batch Files</td>
<td>532</td>
</tr>
<tr>
<td>Unix Commands</td>
<td>533</td>
</tr>
<tr>
<td>Appendix C • Installing and Running Your Standalone Web Server</td>
<td>535</td>
</tr>
<tr>
<td>Using an Active Web Server</td>
<td>535</td>
</tr>
<tr>
<td>Problems Using Your Web Server</td>
<td>535</td>
</tr>
<tr>
<td>Advanced Users</td>
<td>536</td>
</tr>
<tr>
<td>Appendix D • Debugging Your Code</td>
<td>537</td>
</tr>
<tr>
<td>Problems Viewing Your HTML or PHP Programs</td>
<td>537</td>
</tr>
<tr>
<td>Problems with HTML Layout</td>
<td>539</td>
</tr>
<tr>
<td>Locating PHP Syntax Errors</td>
<td>540</td>
</tr>
<tr>
<td>Common PHP Syntax Errors</td>
<td>540</td>
</tr>
<tr>
<td>Common Logical Errors</td>
<td>542</td>
</tr>
<tr>
<td>Appendix E • More about HTML and CSS</td>
<td>545</td>
</tr>
<tr>
<td>Useful HTML References</td>
<td>545</td>
</tr>
<tr>
<td>Useful CSS References</td>
<td>545</td>
</tr>
<tr>
<td>Inline Styles and Internal Style Sheets</td>
<td>546</td>
</tr>
<tr>
<td>Deprecated HTML Tags</td>
<td>546</td>
</tr>
<tr>
<td>Frequently Asked Questions Regarding HTML Tags</td>
<td>547</td>
</tr>
<tr>
<td>Appendix F • More about PHP Functions and Data Types</td>
<td>551</td>
</tr>
<tr>
<td>Useful PHP References</td>
<td>551</td>
</tr>
<tr>
<td>More about PHP Functions and Data Types</td>
<td>551</td>
</tr>
<tr>
<td>Standard PHP Array Functions</td>
<td>552</td>
</tr>
<tr>
<td>Standard PHP File Functions</td>
<td>553</td>
</tr>
<tr>
<td>Standard PHP Math Functions</td>
<td>554</td>
</tr>
<tr>
<td>Standard PHP String Functions</td>
<td>555</td>
</tr>
<tr>
<td>PHP Data Types</td>
<td>556</td>
</tr>
<tr>
<td>Appendix G • Additional PHP Operators and Control Structures</td>
<td>557</td>
</tr>
<tr>
<td>Shortcut Operators</td>
<td>557</td>
</tr>
<tr>
<td>Switch Structure</td>
<td>557</td>
</tr>
</tbody>
</table>
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Another Loop Structure: DO..WHILE</td>
<td>559</td>
</tr>
<tr>
<td>Multi-Dimensional Arrays</td>
<td>560</td>
</tr>
<tr>
<td>Ragged Arrays</td>
<td>562</td>
</tr>
<tr>
<td>Multi-Dimensional Associative Arrays</td>
<td>562</td>
</tr>
<tr>
<td>Index</td>
<td>565</td>
</tr>
</tbody>
</table>
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 minimal PHP 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 minimal set of HTML tags and PHP functions and makes use of some arbitrary conventions to keep the focus on basic concepts that are common to most languages. To give a couple of examples: PHP print statements are used rather than echo statements, and these statements always include parentheses and double quotes so that the syntax is consistent with that of most other languages. And the code examples mostly use a pairing of .html and .php files of the same name (for example wages1.html and wages1.php) to produce simple interactive applications (the .html file provides a form that is processed by the .php file). The last chapter (“Where to Go From Here”) explains which practices are standard and which are particular to the textbook, and also suggests best practices and areas for further study.
The third edition of the book includes a new chapter 14 that introduces MySQL. A brief (and optional) introduction to Object Oriented Programming is included in Chapter 13, but objects are not covered in any detail.

I hope that you will find the book useful for your purposes, and that, if you use the book, you will provide your own feedback and suggestions for the next edition.

### 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.

### 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 style sheets). This makes sense in today’s programming environment where markup and programming are integrated components of 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, Java or
C++, 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), and a database query language (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 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 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 pro-
duce 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 required to install and use the Web server, and also provides hints and help for students, as well as 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 Third Edition

As an instructor I know how frustrating it can be to adapt to changes in textbook editions, so I have tried not to make any more significant structural changes than necessary for this new edition. I have made minor edits throughout, based on feedback and my own experiences in the classroom. These include small changes to quizzes and exercises. The previously lengthy Chapter 11 (arrays) has been broken down into two chapters so that numerically indexed arrays can be treated separately (in Chapter 11) from associative arrays (now in Chapter 12). This has also allowed additional coverage of both topics. For example Chapter 11 now includes more coverage of reading data from files into arrays. And Chapter 12 includes a worked example showing how, by testing $_POST
array input, a single PHP file can combine the code to display an HTML form with the code to process the form. Chapter 13 (previously Chapter 12) now includes a section on the use of die() and exit() functions, and Appendix F has been expanded to include a longer list of useful PHP functions. The new Chapter 14 provides an introduction to the syntax and use of MySQL queries. And Chapter 15 (previously Chapter 13) has been updated to include some important new terms and technologies. The references to XHTML have been removed from the book in light of the growing significance of HTML 5. Reference to XHTML DOCTYPEs have been removed for the same reason. The HTML in the book is syntactically consistent with both XHTML and HTML 5.

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 and Objects. This chapter demonstrates the importance of program modularity and introduces functions, include files and objects. Students learn to write their own functions, to build li-
libraries of related functions, and to call functions from different applications as needed. Key concepts and examples of object oriented programming are also introduced in this chapter as an optional topic.

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: 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.

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 multi-dimensional arrays).
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, and all at Carolina Academic Press for their supportive style, professionalism and experience. Thanks to my fellow instructors at Asheville-Buncombe Technical Community College, especially to Charlie Wallin and Fred Smartt who field-tested the book, made invaluable suggestions and submitted any number of 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.

And 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 fifteen 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.