

A Web-Based Introduction to Programming

A Web-Based Introduction to Programming

**Essential Algorithms, Syntax,
and Control Structures Using PHP and XHTML**

Second Edition

Mike O’Kane

CAROLINA ACADEMIC PRESS

Durham, North Carolina

Copyright © 2011
Mike O'Kane
All Rights Reserved.

Library of Congress Cataloging-in-Publication Data

O'Kane, Mike, 1953-

A web-based introduction to programming : essential algorithms, syntax, and control structures using PHP and XHTML / Mike O'Kane. -- 2nd ed.

p. cm.

Includes bibliographical references and index.

ISBN 978-1-59460-844-5 (alk. paper)

1. Computer software--Development. 2. Internet programming. I. Title.

QA76.76.D47O43 2011

005.1--dc22

2010046893

Carolina Academic Press
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.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of the author.

Please note: The information in this book is provided for instructional value and distributed on an "as is" basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Carolina Academic Press shall have any liability to any person or entity with respect to any loss or damage caused by or alleged to be caused, directly or indirectly, by the instructions contained in this book or by the programs or applications that are listed in, or provided as supplements to, this book.

Macintosh® and Mac OS® are registered trademarks of Apple, Inc. in the United States and other countries. Windows® is a registered trademark of Microsoft Corporation in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the United States and other countries. Mozilla® and Firefox® are registered trademarks of the Mozilla Foundation. Apache® is a trademark of the Apache Software Foundation. All product names identified in this book are trademarks or registered trademarks, and are the properties of their respective companies. We have used these names in an editorial fashion only, and to the benefit of the owner, with no intention of infringing the trademark.

To my dear mother, with all my love and appreciation

Contents

Preface	xv
Changes to the Second Edition	xvii
Acknowledgments	xxiii
About the Author	xxv
Chapter 1 • Introducing Computer Programming	3
Introduction	3
What Is a Computer Program?	4
What Do Programmers Do?	5
The Software Development Life Cycle	11
The Importance of Writing and Communicating	12
What Are Programming Languages?	12
Compilers and Interpreters	13
So Many Languages!	14
Standalone and Network Applications	14
Markup Languages	15
Combining Markup and Programming Languages	16
Summary	16
Chapter 1 Review Questions	17
Chapter 2 • Client/Server Applications—Getting Started	21
Introduction	21
Client/Server Design in Web Applications	22
Working with Files and Folders	23
Locating Files and Folders on Computers Running a Windows Operating System	24
Locating Files and Folders on the Internet	26
Working with a Local Web Server	27
What Languages Will I Use?	28
What Software Will I Need?	29
Installing a Text Editor	29
Installing One or More Web Browsers	30
Installing Your Web Server	30
Using Your Web Server	31

Using URL's with Your Web Server	32
Always Use URL's to Run Your Web Applications!	34
Where to Save Your Work Files	36
The Importance of Frequent Backups	36
Creating an HTML Document	36
Creating a PHP program	38
Creating an Interactive HTML and PHP Program	40
Summary	43
Chapter 2 Review Questions	45
Chapter 3 • Program Design—From Requirements to Algorithms	51
Introduction	51
What Are Instructions?	52
Common Characteristics of Instructions	52
Sequence, Selection and Repetition Structures	56
A Programming Example	57
Creating an Input, Processing, Output (IPO) chart	58
Designing the User Interface	58
Developing an Algorithm	59
A Smoking Calculator	62
Coding the Application	64
Summary	64
Chapter 3 Review Questions	66
Chapter 3 Code Exercises	71
Chapter 4 • Basics of Markup—Creating a User Interface with HTML	75
Introduction	75
A Short History of HTML	77
Introducing HTML Tags	78
Ignoring White Space	80
More HTML Tags	81
Introducing HTML Tables	83
Using HTML Tables to Layout Web Pages	87
Other HTML Tags	89
Deprecated HTML Tags	89
A Note About XHTML Standards and the DOCTYPE Declaration	89
Introducing Style Sheets	90
Multiple Styles for a Single Tag	92
Selecting Colors for Fonts and Backgrounds	93
Referencing a Style Sheet in Your HTML Document	93
Applying a Style Sheet to Multiple Pages	94
Interactive User Interfaces	94
Creating HTML Forms	95
Using HTML Forms to Obtain User Input	97
Using HTML Tables to Line Up Prompt and Input Boxes	100
Problems with Form Submission	101

Drop Down Lists	101
Combining Textboxes and Drop Down Lists	103
Other Types of Input	105
Stylesheets and Forms	105
Summary	106
Chapter 4 Review Questions	107
Chapter 4 Code Exercises	111
Chapter 5 • Creating a Working Program—Basics of PHP	117
Introduction	117
Why PHP?	118
Working with HTML and PHP	118
Important Features of Client/Server Programs	123
Receiving Input from a Form—wage2.php	124
Processing the Smoking Survey—smoking.php	128
PHP—General Guidelines and Syntax	131
Arithmetic Expressions	135
Using Arithmetic Functions	136
White Space in PHP Files	138
Generating HTML Output from PHP	139
Including Double Quotes in Strings	140
Using Multiple PHP Sections	140
Using the number_format() Function to Display Numbers to a Specific Number of Places	141
Including Calls to PHP Functions inside PHP print Statements	141
String Concatenation and the Concatenation Operator	142
The PHP echo Statement	144
Finding Syntax Errors	144
Finding Logical Errors	144
Summary	145
Chapter 5 Review Questions	146
Chapter 5 Code Exercises	151
Chapter 6 • Persistence—Saving and Retrieving Data	155
Introduction	155
The Difference Between Persistent and Transient Data	156
Files and Databases	158
Working with a Text File	159
Closing a Text File	160
Reading Data from a Text File	161
PHP Functions to Read Data from a Text File	162
Writing Data to a Text File	165
PHP Functions to Write Data to a Text File	166
Be Careful to Avoid Security Holes!	170
Using Escape Characters	170
Escape Characters and HTML Tags	171

Using PHP to Append Data to Files	172
PHP Functions to Append Data to a Text File	173
Processing Files that Contain Complete Records on Each Line	175
PHP Functions to Parse a Delimited Character String	176
Processing a File with Multiple Records	179
Appending Records to a File	181
Working with Multiple Files	185
Summary	186
Chapter 6 Review Questions	187
Chapter 6 Code Exercises	192
Chapter 7 • Programs that Choose—Introducing Selection Structures	197
Introduction	197
Introducing IF and IF..ELSE Structures	200
Introducing Flow Charts	201
Boolean Expressions and Relational Operators	204
Selection Using the IF Structure	205
Testing Threshold Values	208
Selection Using the IF..ELSE Structure	209
When to Use Braces in IF..ELSE Statements	213
Comparing Strings—Testing for a Correct Password	214
Ignoring the Case of a Character String	217
Providing a Selective Response	218
Using Selection to Construct a Line of Output	220
Summary	224
Chapter 7 Review Questions	224
Chapter 7 Code Exercises	232
Chapter 8 • Multiple Selection, Nesting, ANDs and ORs	237
Introduction	237
Creating a Program with Multiple but Independent Selection Structures	238
Validating User Input	240
Introducing the Logical Operators AND and OR	242
Using the OR Operator to Validate Input	243
Nested Selection Structures	246
Use of Braces in Nested Selection Structures	247
Chaining Related Selection Structures	249
Introducing the NOT Operator	251
Additional Input Validation	253
More about Input Validation: Using the trim() Function	255
Using the AND Operator to Assign a Bonus	257
When to Use AND or OR? Be Careful with Your Logic!	260
The Challenge of Software Testing	260
A Special Case: The Switch Statement	261
More Examples in the Samples Folder	262
Some Words of Encouragement	263

Summary	264
Chapter 8 Review Questions	264
Chapter 8 Code Exercises	270
Chapter 9 • Programs that Count—Harnessing the Power of Repetition	275
Introduction	275
Controlling a Loop by Counting	277
Coding a FOR Loop in PHP	277
General Syntax of a FOR Loop	280
Including the Counting Variable in Your Loop Statements	281
Using a Variable to Control the Loop Condition	282
Converting from Celsius to Fahrenheit	284
Changing the Increment Value	286
Using Loops with HTML Tables	288
Allowing the User to Control the Loop	289
Improving Processing Efficiency	292
Using Loops to “Crunch Numbers”	293
Using a Loop to Accumulate a Total	293
Finding the Total and Average from a File of Numbers	295
Finding the Highest and Lowest Values in a Series	297
Performing Multiple Operations on a File of Numbers	298
Nesting IF..ELSE Structures to Customize Output from a Loop	300
Loops within Loops—Creating a Bar Chart	304
Selecting from a List of Data Files	308
Summary	309
Chapter 9 Review Questions	310
Chapter 9 Code Exercises	316
Chapter 10 • “While NOT End-Of-File” —Introducing Event-Controlled Loops	321
Introduction	321
Characteristics of WHILE Loops	322
The Structure of WHILE Loops	325
An Algorithm to Process Files of Unknown Length	325
Using a WHILE Loop to Process a File of Scores	328
Including Selection Structures Inside a WHILE Loop	333
Using a WHILE Loop to Count, Sum and Average Data	335
Using a WHILE Loop to Process a File of Records	338
Processing Weekly Wages from a File of Timesheet Records	340
Processing Selected Records from a File of Timesheet Records	342
Processing Selected Fields from a File of Records	345
Processing a File of Survey Data	348
Using DO..WHILE or REPEAT..UNTIL Loops	352
Summary	352
Chapter 10 Review Questions	353
Chapter 10 Code Exercises	361

Chapter 11 • Structured Data — Working with Arrays	365
Introduction	365
What Is an Array?	366
Working with Array Elements	367
Extending an Array	368
Displaying Array Values	368
Receiving Scores into an Array from an HTML Form	369
Arrays of Strings	372
How Large Is the Array?	373
Why Do Array Indices Begin with 0 and Not 1?	373
Using FOR Loops with Arrays	374
Using the sizeof() Function to Control a FOR Loop	375
Summing and Averaging the Values in an Array	375
Counting Selected Values in an Array	376
Multiple Operations on an Array	376
Reading Data from a File into an Array	378
Reading Data into an Array from a File of Unknown Length	381
A Special Loop for Processing Arrays—FOREACH	382
Associative Arrays	383
Using a Variable to Reference the Key of an Associative Array	384
Using Associative Arrays as Lookups	385
Using the array() Function to Create Associative Arrays	386
Associative Arrays and the FOREACH Loop	386
Multi-Dimensional Arrays	388
More about the \$_POST Array	388
Web Sessions and the \$_SESSION Array	390
Adding Code to Manage a Web Session	392
Creating, Initializing and Modifying Session Variables	393
Validating \$_SESSION and \$_POST Arrays	396
Revisiting the Same Page in a Web Session	397
Summary	401
Chapter 11 Review Questions	403
Chapter 11 Code Exercises	408
Chapter 12 • Program Modularity — Working with Functions and Objects	413
Introduction	413
Using Functions	415
Understanding Function Arguments	416
Receiving Values from a Function	417
Researching Available Functions	418
Creating Your Own Functions	419
Where Do I Put My Functions?	422
Creating a Library of Functions	424
Including Functions from External Files	425
Using the Same Functions in Different Programs	426

Learning to Think Beyond Specific Applications	429
More about Include Files	433
OPTIONAL: Introducing Object Oriented Programming (OOP)	434
Defining an Object Class	435
Coding the Object Class	436
Creating and Using Instances of an Object Class	439
The Importance of OOP	442
Important OOP Terms and Concepts	442
Summary	444
Chapter 12 Review Questions	445
Chapter 12 Code Exercises	451
Chapter 13 • Where to Go from Here...	455
Introduction	455
Moving Forward with PHP and XHTML	455
More about PHP	459
PHP and Other Languages	459
XHTML and XML	461
Client Side Processing with Javascript and Ajax	462
The Importance of OOP	463
IDE's, Modeling Languages and Frameworks	463
Client/Server and Server/Server Programming	464
GUI Programming, Content Management, and Interface Design	464
Database Programming and SQL	465
In Summary: Follow Your Heart!	465
Appendix A • Data Representation and Formats	467
Introduction	467
Storing Data in Bits and Bytes	467
How Multimedia Data Is Represented in Binary	468
How Numeric Values Are Represented in Binary	469
How Plain Text Is Represented in Binary	470
How Source Code and Markup Code Is Represented in Binary	471
How Program Instructions Are Represented in Binary	472
How Memory Addresses Are Represented in Binary	472
What Else Can Be Represented in Binary?	472
Appendix B • Files, Folders, Addressing Schemes, and Command	
Line Arguments	473
File Types and File Extensions	473
Disk and Disk Drives	474
Files and File Folders (Directories)	475
Naming Files and Folders	476
File Addresses in Windows and on the Web	476
Relative Addresses in Windows	477

Relative Addresses on the Internet	478
Using Relative Web Addresses in HTML code	478
Managing Files at the Command Line	479
Introduction to MS DOS Commands	479
Recalling Previous Commands	484
Use Double Quotes when Paths Include Spaces	484
Printing the Contents of the Console Window	484
Creating Batch Files	484
Unix Commands	485
Appendix C • Installing and Running Your Standalone Web Server	487
Using Other Web Servers	487
Installing the Web Server on a Linux Computer	488
Problems Using Your Web Server	488
Security	488
Appendix D • Debugging Your Code	489
Problems Viewing Your HTML or PHP Programs	489
Problems with HTML Layout	491
Locating PHP Syntax Errors	492
Common PHP Syntax Errors	492
Common Logical Errors	494
Appendix E • More About HTML and CSS	497
Useful HTML References	497
Useful CSS References	497
Inline Styles and Internal Style Sheets	498
Deprecated HTML Tags and XHTML Standards	498
Frequently Asked Questions Regarding HTML Tags	499
Appendix F • More About PHP Functions and Data Types	505
Useful PHP References	505
More about PHP Functions and Data Types	505
Standard PHP Array Functions	506
PHP Data Types	507
Appendix G • Additional PHP Operators and Control Structures	509
Shortcut Operators	509
Switch Structure	509
Another Loop Structure: DO..WHILE	511
Multi-Dimensional Arrays	512
Ragged Arrays	513
Multi-Dimensional Associative Arrays	514
Index	517

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 XHTML code to develop small, interactive Web applications. This approach has proved very successful. Many students take the trouble to report how much they enjoy the course, how much they have learned, and how well the course 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 be frustrated by my “keep it simple” approach, and may have concerns that my coverage of the PHP and XHTML is insufficient. The book uses a minimal set of XHTML 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 in order to illustrate text output in a manner consistent with most other languages. And the code examples mostly use a pairing of .html and .php files of the same name to produce simple interactive applications (the .html file provides a form that is processed by the .php file). I have tried to differentiate between these editorial strategies and actual language requirements. 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.

Two major topics that are hardly touched in this book are Object Oriented Programming and database/SQL programming. I considered how to incorporate these but came to the conclusion that it was simply too much for a beginning course.

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 increasingly 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, 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), and a scripting language (HTML), but does not attempt to provide a complete overview of either. Instead, students learn sufficient syntax to convert requirements into working applications using basic programming structures, arithmetic and logical expressions, user interfaces, functions, and data files. 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. As Chapter 2 demonstrates, students can begin programming in HTML and PHP 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 CD includes 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).

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 all materials found on the CD, and also provides access to 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 Second Edition

As an instructor myself I know how frustrating it can be to adapt to changes in textbook editions, so I have tried not make significant structural changes in this new edition. Throughout the book I have made minor edits to clarify concepts and procedures where students have seemed to struggle. Chapter 2 provides a much more efficient procedure to install and run the Web server. Examples of flow charts have been added to the chapters that cover control structures. Chapter 11 now has significant new material to further explain the use of associative arrays and Web sessions (note that these additions have significantly extended the length of this chapter and it may require more class time). Chapter 12 provides more extensive discussion of the use of include files, and a more coherent introduction to objects. Chapter 13 has been renamed "Where to Go From Here" and provides additional references to important current technologies. The appendices have also been reviewed and updated. One important language update has been the replacement of the PHP `split()` function with the `explode()` function, since the `split()` function is now deprecated. The `explode()` function is identical in the way that it is used.

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 summarizes significant important 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 in HTML. This chapter explains the significance of data rendering, and provides a brief overview and history of Hypertext Markup Language (HTML), up to the present XHTML implementation. 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, AND’s and OR’s. 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 and associative 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. Students learn how to use associative arrays as lookups, and gain a better understanding of the way that data is received from HTML forms. Web sessions are introduced, and students learn how to use session variables to maintain session data between applications. This chapter is longer than most and additional time may be needed.

Chapter 12: 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 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 13: 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 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 short 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).

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, 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 twelve years experience teaching computer science courses, including his current position 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.