

A Web-Based Introduction to Programming

A Web-Based Introduction to Programming

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

Mike O’Kane

CAROLINA ACADEMIC PRESS

Durham, North Carolina

Copyright © 2008
Mike O’Kane
All Rights Reserved.

Library of Congress Cataloging-in-Publication Data

O’Kane, Michael J.

A web-based introduction to programming : essential algorithms, syntax and control structures using PHP and XHTML / by Michael J. O’Kane.

p. cm.

Includes index.

ISBN 978-1-59460-523-9 (alk. paper)

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

QA76.76.D47O43 2008

005.1--dc22

2008020470

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
Acknowledgments	xxi
About the Author	xxiii
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	11
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 XAMPPLite in Windows	30
Installing Crimson Editor	31
Installing Mozilla Firefox	31

Running Your Web Server on Windows	31
Using URL's With Your Web Server	34
Always Use URL's to Run Your Web Applications!	35
Where to Save Your Files	37
The Importance of Frequent Backups	37
Creating an HTML Document	37
Creating a PHP program	39
Creating an Interactive HTML and PHP Program	41
Summary	45
Chapter 2 Review Questions	46
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	63
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	84
Using HTML Tables to Layout Web Pages	87
Other HTML Tags	89
Introducing Style Sheets	89
Multiple Styles for a Single Tag	90
Selecting Colors for Fonts and Backgrounds	91
Referencing a Style Sheet in Your HTML Document	92
Applying a Style Sheet to Multiple Pages	92
Interactive User Interfaces	93
Creating HTML Forms	93
Using HTML Forms to Obtain User Input	96
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	104
Chapter 4 Code Exercises	108
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	125
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
Using Multiple PHP Sections	138
Using the number_format() Function to Display Numbers to a Specific Number of Places	138
Finding Syntax Errors	140
Finding Logical Errors	140
Summary	141
Chapter 5 Review Questions	141
Chapter 5 Code Exercises	146
Chapter 6 · Persistent Data — Working with Files and Databases	149
Introduction	149
The Difference Between Persistent and Transient Data	150
Files and Databases	152
Working with a Text File	153
Closing a Text File	155
Reading Data from a Text File	155
PHP Functions to Read Data from a Text File	156
Writing Data to a Text File	159
PHP Functions to Write Data to a Text File	160
Be Careful to Avoid Security Holes!	164
Using Escape Characters	164
Using PHP to Append Data to Files	165
PHP Functions to Append Data to a Text File	167
Processing Files that Contain Complete Records on Each Line	169
PHP Functions to Parse a Delimited Character String	170
Processing a File with Multiple Records	172

Appending Records to a File	175
Working with Multiple Files	179
Summary	179
Chapter 6 Review Questions	180
Chapter 6 Code Exercises	186
Chapter 7 · Programs that Choose — Introducing Selection Structures	189
Introduction	189
Introducing IF and IF..ELSE Structures	192
Boolean Expressions and Relational Operators	193
Selection Using the IF Structure	194
Testing Threshold Values	196
Selection using the if..else Structure	197
When to Use Braces in IF..ELSE Statements	200
Comparing Strings — Testing for a Correct Password	201
Ignoring the Case of a Character String	204
Providing a Selective Response	205
Using Selection to Construct a Line of Output	208
Summary	211
Chapter 7 Review Questions	212
Chapter 7 Code Exercises	219
Chapter 8 · Multiple Selection, Nesting, ANDs and ORs	223
Introduction	223
Creating a Program with Multiple but Independent Selection Structures	224
Validating User Input	226
Introducing the Logical Operators AND and OR	228
Using the OR Operator to Validate Input	229
Nested Selection Structures	232
Chaining Related Selection Structures	234
Introducing the NOT Operator	236
Additional Input Validation	237
More about Input Validation: Using the trim() Function	240
Using the AND Operator to Assign a Bonus	242
When to Use AND or OR? Be Careful with Your Logic!	245
The Challenge of Software Testing	245
A Special Case: The Switch Statement	246
More Examples in the Samples Folder	247
Some Words of Encouragement	248
Summary	249
Chapter 8 Review Questions	249
Chapter 8 Code Exercises	255
Chapter 9 · Programs that Count — Harnessing the Power of Repetition	259
Introduction	259
Controlling a Loop by Counting	261

General Syntax of a FOR loop	263
Including the Counting Variable in Your Loop Statements	264
Using a Variable to Control the Loop Condition	265
Converting from Celsius to Fahrenheit	267
Changing the Increment Value	269
Using Loops with HTML tables	271
Allowing the User to Control the Loop	272
Improving Processing Efficiency	275
Using Loops to “Crunch Numbers”	276
Using a Loop to Accumulate a Total	276
Finding the Total and Average from a File of Numbers	278
Finding the Highest and Lowest Values in a Series	280
Performing Multiple Operations on a File of Numbers	282
Nesting IF..ELSE Structures to Customize Output from a Loop	284
Loops within Loops — Creating a Bar Chart	287
Selecting from a List of Data Files	291
Summary	293
Chapter 9 Review Questions	293
Chapter 9 Code Exercises	300
Chapter 10 · "While NOT End-Of-File" — Introducing Event-Controlled Loops	303
Introduction	303
Characteristics of WHILE Loops	304
The Structure of WHILE Loops	306
An Algorithm to Process Files of Unknown Length	306
Using a WHILE Loop to Process a File of Scores	308
Including Selection Structures Inside a WHILE Loop	313
Using a WHILE Loop to Count, Sum and Average Data	315
Using a WHILE Loop to Process a File of Records	318
Processing Weekly Wages from a File of Timesheet Records	321
Processing Selected Records from a File of Timesheet Records	323
Processing Selected Fields from a File of Records	326
Processing a File of Survey Data	328
Using DO..WHILE or REPEAT..UNTIL Loops	333
Summary	333
Chapter 10 Review Questions	334
Chapter 10 Code Exercises	342
Chapter 11 · Structured Data — Working with Arrays	345
Introduction	345
What Is an Array?	346
Working with Array Elements	347
Extending an Array	348
Displaying Array Values	348

Receiving Scores into an Array from an HTML Form	349
Arrays of Strings	352
How Large Is the Array?	353
Why Do Array Indices Begin with 0 and Not 1?	353
Using FOR Loops with Arrays	354
Using the sizeof() Function to Control a FOR Loop	355
Summing and Averaging the Values in an Array	355
Counting Selected Values in an Array	356
Multiple Operations on an Array	356
Reading Data from a File into an Array	358
Reading Data into an Array from a File of Unknown Length	361
A Special Loop for Processing Arrays - FOREACH	362
Associative Arrays and Hashes	363
Associative Arrays and the FOREACH Loop	365
More about the \$_POST Array	365
Web Sessions and the \$_SESSION array	367
Associating Multiple Pages with a Web Session	368
Creating, Initializing and Modifying Session Variables	368
Destroying the Session	369
Managing a Web Session	369
Putting it All Together	370
Multi-Dimensional Arrays	372
Summary	372
Chapter 11 Review Questions	374
Chapter 11 Code Exercises	380
Chapter 12 · Program Modularity — Working with Functions and Objects	383
Introduction	383
Using Functions	385
Understanding Function Arguments	386
Receiving Values from a Function	387
Researching Available Functions	388
Creating Your Own Functions	389
Creating a Library of Functions	392
Including Functions from External Files	393
Using the Same Functions in Different Programs	395
Learning to Think Beyond Specific Applications	398
More about Include Files	401
Introducing Object Oriented Programming (OOP)	402
Defining an Object Class	403
Creating and Using Instances of an Object Class	407
The Importance of OOP	409
Important OOP Terms and Concepts	410
Summary	411
Chapter 12 Review Questions	413
Chapter 12 Code Exercises	418

Chapter 13 · In Conclusion...	421
Introduction	421
Moving Forward with PHP and XHTML	421
PHP and Other Languages	422
Web Design and Markup	424
The Importance of OOP	425
Development Tools and Modeling Languages	425
Client/Server and Server/Server Programming	425
GUI Programming and Interface Design	426
Database Programming and SQL	426
Summary	427
Appendix A · Data Representation and Formats	429
Introduction	429
Storing Data in Bits and Bytes	429
How Multimedia Data Is Represented in Binary	430
How Numeric Values Are Represented in Binary	431
How Plain Text Is Represented in Binary	432
How Source Code and Markup Code Is Represented in Binary	433
How Program Instructions Are Represented in Binary	434
How Memory Addresses Are Represented in Binary	434
What Else Can Be Represented in Binary?	434
Appendix B · Files, Folders, Addressing Schemes, and Command Line Arguments	435
File Types and File Extensions	435
Disk and Disk Drives	436
Files and File Folders (Directories)	437
Naming Files and Folders	438
File Addresses in Windows and on the Web	438
Relative Addresses in Windows	439
Relative Addresses on the Internet	440
Using Relative Web Addresses in HTML code	440
Managing Files at the Command Line	441
Introduction to MS DOS Commands	441
Recalling Previous Commands	446
Use Double Quotes when Paths Include Spaces	446
Printing the Contents of the Console Window	446
Creating Batch Files	446
Unix Commands	447
Appendix C · Installing and Running xamppLite	449
Do You Already Have Access to a Web server?	449
Information and Help Regarding Your xamppLite Installation	449
Problems with Installation	450

Adding the Coursework and Samples Folders to Your Mac OS X or Linux Installation	450
Editing Your Files	451
Text Editors for Macintosh and Linux	451
Appendix D · Debugging Your Code	453
Problems Viewing Your HTML or PHP Programs	453
Problems with HTML Layout	456
Locating PHP Syntax Errors	456
Common PHP Syntax Errors	457
Common Logical Errors	458
Appendix E · More about HTML and CSS	463
Useful HTML References	463
Useful CSS References	463
Useful Color References	464
Inline Styles and Internal Style Sheets	464
Deprecated HTML Tags and XHTML Standards	464
Frequently Asked Questions Regarding HTML Tags	465
Appendix F · More About PHP	469
Useful PHP References	469
Use PHP print or echo?	469
Multiple PHP Sections	469
Should I Use .html or .php Files?	470
Consistent File Names	470
Program Variables and the \$_POST Array	470
Form-Processing Best Practices	470
Camelback Notation	471
More about PHP Functions	471
Standard PHP Array Functions	472
PHP Data Types	472
Appendix G · Additional Topics	475
Shortcut Operators	475
Switch Structure	475
Another Loop Structure: DO..WHILE	477
Multi-Dimensional Arrays	478
Ragged Arrays	479
Multi-Dimensional Associative Arrays	480
Using the FOREACH Loop with Associative Arrays	481
Index	483

Preface

This book is written in the hope of providing a measured, engaging, and carefully structured learning environment for anyone wishing to learn how to program. The book content has been extensively field-tested in both traditional and online courses. Feedback from instructors and students alike has been overwhelmingly positive.

Intended Audience

The book is designed to serve:

- Instructors teaching introductory programming, programming logic and design, or Web programming courses, who want an effective 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 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 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 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 containing 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/>

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. Step-by-step instructions are provided to install, run, and test the required software. Students are then 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.

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: Persistent Data – Working with Files and Databases. 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.

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.

Chapter 13: In Conclusion. 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 instructions to install the Web server under Windows, Mac OS X, and Linux.

Appendix D provides extensive 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 material and references for students wishing to learn more about PHP.

Appendix G covers topics that were intentionally left out of 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 Bob Conrow, Scott Sipe, Beth Hall, Reuben Ayres, and all at Carolina Academic Press for their supportive style, professionalism and experience. Thanks to Pam Silvers and 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.