

# **A Web-Based Introduction to Programming**



# **A Web-Based Introduction to Programming**

---

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

**Fifth Edition**

**Mike O’Kane**



**CAROLINA ACADEMIC PRESS**

**Durham, North Carolina**

Copyright © 2022  
Mike O’Kane  
All Rights Reserved.

### Library of Congress Cataloging-in-Publication Data

Names: O’Kane, Mike, 1953- author.

Title: A web-based introduction to programming : essential algorithms, syntax, and control structures using PHP, HTML, and MariaDB/MySQL / by Mike O’Kane.

Description: Fifth edition. | Durham, North Carolina : Carolina Academic Press, LLC, [2021] | Includes index.

Identifiers: LCCN 2021035347 (print) | LCCN 2021035348 (ebook) | ISBN 9781531022105 (paperback) | ISBN 9781531022112 (ebook)

Subjects: LCSH: Computer software—Development. | Internet programming. | Computer programming—Web-based instruction. | PHP (Computer program language) | XHTML (Document markup language)

Classification: LCC QA76.76.D47 O43 2021 (print) | LCC QA76.76.D47 (ebook) | DDC 005.3—dc23

LC record available at <https://lcn.loc.gov/2021035347>

LC ebook record available at <https://lcn.loc.gov/2021035348>

Carolina Academic Press  
700 Kent Street  
Durham, North Carolina 27701  
(919) 489-7486

[www.cap-press.com](http://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®, macOS®, OSX®, Safari, and iOS® are registered trademarks of Apple, Inc. in the United States and other countries. Windows® and Windows Mobile® are registered trademarks 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. MySQL® is a registered trademark of Oracle Corporation and/or its affiliates. MariaDB® is a registered trademark of MariaDB Corporation Ab. Mozilla® and Firefox® are registered trademarks of the Mozilla Foundation. Joomla!® is trademarked by Open Source Matters. Apache® is a trademark of the Apache Software Foundation. Django® is a registered trademark of the Django Software Foundation. GitHub® is a registered trademark of GitHub, Inc. XAMPP and Apache Friends are registered trademarks of BitRock. The WordPress® trademark is owned by the WordPress Foundation. Android® and Google Chrome® are trademarks of Google Inc. The Drupal® trademark is owned and controlled by Dries Buytaert. BlackBerry® is a trademark of Blackberry. 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 and father,  
thank you for the love and light that you bestowed on us.*



# Contents

Preface	xix
Acknowledgments	xxvii
About the Author	xxix
<b>Chapter 1 · Introducing Computer Programming</b>	<b>3</b>
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	10
Version Control Systems	10
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	14
Additional Learning Material	14
Summary	14
Chapter 1 Review Questions	16
<b>Chapter 2 · Client/Server Applications—Getting Started</b>	<b>19</b>
Introduction	19
Client/Server Design in Web Applications	20
Working with Files and Folders	21
File and Folder Addresses	23
File and Folder Addresses under Windows	23
File and Folder Addresses under macOS	24
File and Folder Addresses under Linux	24
More About Managing Files and Folders	25
Files and Folders on the Internet	25
Naming Conventions for Files and Folders	26

---

Working with a Local Web Server	27
What Languages Will I Use?	28
What Software Will I Need?	29
Installing a Text Editor	30
Installing One or More Web Browsers	30
Installing Your Web Server	31
Using Your Web Server	31
Using URLs with Your Web Server	33
Always Use URLs to Run Your Web Applications!	34
Where to Save Your Work Files	36
The Importance of Frequent Backups	37
Creating an HTML Document	37
Creating a PHP Program	39
Creating an Interactive HTML and PHP Program	41
Additional Learning Material	43
Summary	44
Chapter 2 Review Questions	46
Chapter 2 Code Exercises	50
<b>Chapter 3 · Program Design — From Requirements to Algorithms</b>	<b>53</b>
Introduction	53
What Are Instructions?	54
Common Characteristics of Instructions	54
Sequence, Selection and Repetition Structures	58
A Programming Example	61
Creating an Input, Processing, Output (IPO) chart	61
Designing the User Interface	62
Developing an Algorithm	63
Variables, Assignments and Expressions	65
Named Constants	66
A Smoking Calculator	68
Coding the Application	70
Additional Learning Material	70
Summary	70
Chapter 3 Review Questions	72
Chapter 3 Code Exercises	77
<b>Chapter 4 · Basics of Markup — Creating a User Interface with HTML</b>	<b>81</b>
Introduction	81
A Short History of HTML	82
Introducing HTML Tags	83
Ignoring White Space	87
More HTML Tags	87
Introducing HTML Tables	90
Other HTML Tags	94



---

Introducing Style Sheets	94
Specifying Fonts	96
Defining the Same Properties for Multiple Elements	96
Defining Different Styles for the Same HTML Element	97
Selecting Colors for Fonts and Backgrounds	97
Referencing a Style Sheet in Your HTML Document	98
Applying the Same Style Sheet to Multiple Pages	99
Don't Feel Overwhelmed	99
Interactive User Interfaces	100
Creating HTML Forms	100
Using HTML Forms to Obtain User Input	103
Problems with Form Submission	106
Drop Down Lists	106
Combining Textboxes and Drop Down Lists	108
Other Types of Input	110
CSS Layout and HTML Forms	110
Internal and Inline Style Sheets	112
Additional Learning Materials	114
Summary	114
Chapter 4 Review Questions	116
Chapter 4 Code Exercises	121
<b>Chapter 5 · Creating a Working Program — Basics of PHP</b>	<b>129</b>
Introduction	129
Why PHP?	130
Working with HTML and PHP	130
How a .php File is Processed	132
Important Features of Client/Server Programs	135
Receiving Input from a Form — wage2.php	136
Processing the Smoking Survey — smoking.php	140
Using Fewer print() Statements	144
PHP — General Guidelines and Syntax	144
Arithmetic Expressions	149
Using Arithmetic Functions	150
White Space in PHP Files	152
Generating Character Strings from PHP	153
Including Double Quotes in Character Strings	154
Using the number_format() Function to Display Numbers to a Specific Number of Places	155
Including Calls to PHP Functions inside PHP Print Statements	156
String Concatenation and the Concatenation Operator	157
The PHP Echo Statement	159
Multiple PHP Sections	159
Finding Syntax Errors	159
Finding Logical Errors	160

---

Additional Learning Materials	160
Summary	161
Chapter 5 Review Questions	162
Chapter 5 Code Exercises	167
<b>Chapter 6 · Persistence — Saving and Retrieving Data</b>	<b>171</b>
Introduction	171
The Difference Between Persistent and Transient Data	172
Files and Databases	174
Working with a Text File	175
Closing a Text File	176
Reading Data from a Text File	177
PHP Functions to Read Data from a Text File	178
Writing Data to a Text File	181
PHP Functions to Write Data to a Text File	183
Why fgets() and fputs()?	186
Using Escape Characters	186
Escape Characters and HTML Tags	187
Using PHP to Append Data to Files	187
PHP Functions to Append Data to a Text File	189
Processing Files that Contain Complete Records on Each Line	191
PHP Functions to Parse a Delimited Character String	192
Processing a File with Multiple Records	195
Appending Records to a File	197
Working with Multiple Files	201
Additional Learning Materials	202
Summary	202
Chapter 6 Review Questions	204
Chapter 6 Code Exercises	210
<b>Chapter 7 · Programs that Choose — Introducing Selection Structures</b>	<b>215</b>
Introduction	215
Introducing IF and IF..ELSE Structures	218
Introducing Flow Charts	219
Boolean Expressions and Relational Operators	222
Selection Using the IF Structure	223
Testing Threshold Values	227
Selection Using the IF..ELSE Structure	227
Indenting Code Inside Selection Structures	231
When to Use Braces in IF..ELSE Statements	231
Creating a Program with Multiple but Independent Selection Structures	233
Comparing Strings — Testing for a Correct Password	235
Ignoring the Case of a Character String	238
Providing a Selective Response	239
Using Selection to Construct a Line of Output	242

---

Troubleshooting and Desk-Checking	246
Additional Learning Materials	247
Summary	247
Chapter 7 Review Questions	249
Chapter 7 Code Exercises	257
<b>Chapter 8 · Multiple Selection, Nesting, ANDs and ORs</b>	<b>261</b>
Introduction	261
Introducing the Logical Operators AND and OR	262
Introducing the NOT Operator	265
Validating User Input	266
Validation Rules for the Wage Application	266
Using A Nested Selection Structure to Validate Input	268
Designing Applications with Nested Selection Structures	271
Use of Braces in Nested Selection Structures	272
Indenting Nested Selection Structures	274
Chaining Related Selection Structures	274
More About Chained Selection Structures	279
Reducing Input Errors and Improving Security	282
More about Validation and Security	283
When to Use AND or OR? Be Careful with Your Logic!	283
The Challenge of Software Testing	284
A Special Case: The Switch Statement	285
More Examples in the Samples Folder	286
Some Words of Encouragement	287
Additional Learning Materials	288
Summary	288
Chapter 8 Review Questions	290
Chapter 8 Code Exercises	296
<b>Chapter 9 · Programs that Count — Harnessing the Power of Repetition</b>	<b>301</b>
Introduction	301
Controlling a Loop by Counting	303
Coding a FOR Loop in PHP	305
General Syntax of a FOR Loop	307
Indenting Statements Inside a FOR Loop	308
Including the Counting Variable in Your Loop Statements	308
Using a Variable to Control the Loop Condition	309
Converting from Celsius to Fahrenheit	311
Changing the Increment Value	313
Using Loops with HTML Tables	314
Allowing the User to Control the Loop	316
Improving Processing Efficiency	319
Using Loops to “Crunch Numbers”	320
Using a Loop to Accumulate a Total	320

Finding the Total and Average from a File of Numbers	322
Using a Loop to Find the Total of Selected Values in a Series	324
Using a Loop to Count Selected Values in a Series	326
Finding the Highest and Lowest Values in a Series	328
Performing Multiple Operations on a File of Numbers	330
Nesting IF..ELSE Structures to Customize Output from a Loop	332
Loops within Loops — Creating a Bar Chart	335
Selecting from a List of Data Files	339
Additional Learning Materials	341
Summary	341
Chapter 9 Review Questions	343
Chapter 9 Code Exercises	349
<b>Chapter 10 · “While NOT End-Of-File” —</b>	
<b>Introducing Event-Controlled Loops</b>	355
Introduction	355
Characteristics of WHILE Loops	356
The Structure of WHILE Loops	359
An Algorithm to Process Files of Unknown Length	359
Using a WHILE Loop to Process a File of Scores	362
Including Selection Structures Inside a WHILE Loop	367
Using a WHILE Loop to Count, Sum and Average Data	369
Using a WHILE Loop to Process a File of Records	372
Processing Weekly Wages from a File of Timesheet Records	375
Processing Selected Records from a File of Timesheet Records	377
Processing Selected Fields from a File of Records	380
Processing Selected Records from a File of Records	382
Processing a File of Survey Data	385
Additional Learning Materials	389
Summary	389
Chapter 10 Review Questions	391
Chapter 10 Code Exercises	399
<b>Chapter 11 · Structured Data — Working with Arrays</b>	405
Introduction	405
What Is an Array?	406
Working with Array Elements	407
Extending an Array	408
Displaying Array Values	408
Receiving Scores into an Array from an HTML Form	409
Arrays of Strings	414
Using a Variable as an Array Index	415
How Large Is the Array?	417
Why Do Array Indices Begin with 0 and Not 1?	417
Using FOR Loops with Arrays	417

---

Using the sizeof() Function to Control a FOR Loop	418
Summing and Averaging the Values in an Array	419
Counting Selected Values in an Array	420
Multiple Operations on an Array	420
Reading Data from a File into an Array	422
Reading Data into an Array from a File of Unknown Length	425
Using [] with no Index Value	426
Reading Selected Data from a File into an Array	427
Reading Data from a File into Multiple Arrays	428
Reading Selected Data from a File of Records into an Array	429
More About the explode() and list() Functions	430
A Special Loop for Processing Arrays — FOREACH	432
Multi-Dimensional Arrays	433
Additional Learning Materials	433
Summary	433
Chapter 11 Review Questions	435
Chapter 11 Code Exercises	440
<b>Chapter 12 · Associative Arrays and Web Session</b>	<b>445</b>
Introduction	445
Using a Variable to Reference the Key of an Associative Array	446
Using Associative Arrays as Lookups	447
Using the array() Function to Create Associative Arrays	449
Associative Arrays and the FOREACH Loop	449
More about the \$_POST Array	451
Other Superglobal Variables	452
Using the \$_SERVER Array to Combine Web Form and Form Processing Code in a Single Page	453
Web Sessions and the \$_SESSION Array	455
Adding Code to Manage a Web Session	457
Creating, Initializing and Modifying Session Variables	458
Validating \$_SESSION and \$_POST Arrays	461
Revisiting the Same Page in a Web Session	463
Additional Learning Materials	467
Summary	467
Chapter 12 Review Questions	469
Chapter 12 Code Exercises	471
<b>Chapter 13 · Program Modularity — Working with Functions</b>	<b>475</b>
Introduction	475
Using Functions	477
Understanding Function Arguments	478
Receiving Values from a Function	479
Researching Available Functions	480
Reasons to Use Pre-Defined Functions	481

Using die() or exit() to Terminate an Application	481
Creating Your Own Functions	482
Where Do I Put My Functions?	485
Creating a Library of Functions	487
Including Functions from External Files	488
Using the Same Functions in Different Programs	489
Functions Calling Functions	492
Learning to Think Beyond Specific Applications	495
More about Include Files	499
Additional Learning Materials	501
Summary	501
Chapter 13 Review Questions	503
Chapter 13 Code Exercises	509
<b>Chapter 14 · Connecting to a Database — Working with MySQL</b>	<b>513</b>
Introduction	514
What Is a Relational Database?	514
The Relational Database Management System (RDBMS)	514
Structured Query Language — MySQL	515
Starting Your MySQL Server	516
Configuring MySQL for Use with This Textbook	516
Three Ways to Work with MySQL	517
Working with PHP and MySQL	517
Using PHP to Open and Close a Connection to a MySQL Server	517
Using the MySQL SELECT Query	519
Selecting Specific Records	520
Relational Operators in MySQL	521
The Logical Operators AND and OR	522
Ordering Your Query Results	523
Viewing Your Query Results	524
Using an HTML Table to Display the Query Results	526
Putting It All Together	526
Using Input from an HTML Form to Construct a Query	528
Processing Queries with a Single Result	529
Performing Calculations with the Result Set	529
Performing Aggregate Operations on MySQL Queries	530
Performing JOIN Operations on Multiple Tables	531
Using INSERT to Add Records to a Table	533
Using UPDATE to Modify a Record	533
Removing a Record	534
Storing MySQL Connection Data in an Include File	535
Creating, Dropping, and Altering Databases and Tables	536
Additional Learning Materials	536
Summary	537

---

Chapter 14 Review Questions	539
Chapter 14 Code Exercises	543
<b>Chapter 15 · Introduction to Object-Oriented Programming</b>	<b>545</b>
Introduction	546
What is an Object?	547
Creating and Using Instances of a Class	548
Using Employee Objects in an Application	549
Defining an Object	553
Coding the Object Class	554
Creating and Using Instances of an Object Class	558
The Class Constructor Method	559
Method Overloading	560
Why do Objects Matter?	560
Object Design and Inheritance	562
Abstract Classes and Methods	569
Method Over-riding	570
Polymorphism	571
OOP and Databases	571
OOP Development	572
OOP Languages	572
Additional Learning Materials	573
Summary	573
Chapter 15 Review Questions	575
Chapter 15 Code Exercises	578
<b>Chapter 16 · More About PHP</b>	<b>581</b>
Introduction	582
Textbook Conventions	582
Code Comments	583
Location of Curly Braces	583
HTML and PHP Files	584
Multiple PHP Sections	585
Program Variables and the \$_POST Array	585
Naming Files, Variables, and Constants	585
Choosing Between print and echo	586
Single or Double Quotes in PHP	586
Beware of “Smart” Quotes!	587
The PHP Concatenation Operator	587
PHP Shortcut Operators	588
The PHP SWITCH Statement	588
The PHP DO..WHILE Loop	591
Multi-Dimensional Arrays	592
Ragged Arrays	594

---

Multi-Dimensional Associative Arrays	594
PHP Data Types	595
PHP Tools to Validate and Sanitize Data	597
PHP Date and Time Functions	598
Standard PHP Functions	600
Standard PHP Array Functions	601
Standard PHP File Functions	602
Standard PHP Math Functions	602
Standard PHP String Functions	603
<b>Chapter 17 · More about HTML and CSS</b>	<b>605</b>
More About HTML Forms	606
Hidden Fields	606
Radio Buttons	607
Single and Multiple Check Boxes	608
Text Areas	611
Processing the Form	612
HTML Form Reset Buttons	615
Validating and Sanitizing Form Input	615
HTML List Elements	615
Ordered Lists	616
Bulleted Lists	616
Definition Lists	616
Horizontal Rules	618
Background Images for Web Pages	618
Creating a Clickable Image	618
Block Quotes and Citations	619
DIV and SPAN Containers	619
HTML Entities	622
Deprecated HTML Tags	623
Useful References	624
<b>Chapter 18 · Where to Go from Here . . .</b>	<b>625</b>
Introduction	625
1. Read Chapters 16 and 17	625
2. Take Your Work Online	626
3. Review/Download the Support Tutorials	626
4. Learn to use Git and GitHub for Version Control	627
5. Investigate IDEs, Frameworks, and Patterns	628
6. Learn more about OOP	629
7. Learn Javascript	629
8. Explore other Languages	630
9. Join Online Communities, Forums and Projects	631
10. Consider your Professional Direction	632
And Step 11 ... Have Fun and Follow your Heart!	633



---

<b>Appendix A · Data Representation and Formats</b>	635
Introduction	635
Storing Data in Bits and Bytes	635
How Multimedia Data Is Represented in Binary	636
How Numeric Values Are Represented in Binary	637
How Plain Text Is Represented in Binary	638
How Source Code and Markup Code Is Represented in Binary	639
How Program Instructions Are Represented in Binary	640
How Memory Addresses Are Represented in Binary	640
Data Formats and File Types	640
What Else Can Be Represented in Binary?	642
<b>Appendix B · File Storage, File Types, and File     Addresses (Absolute and Relative)</b>	643
Files and Folders	643
File Storage Locations	643
Naming Files	644
File Addresses	645
Relative Addresses	646
Absolute Addresses	647
When to Use Absolute or Relative Addresses	648
Using File Addresses at the Command Line	649
<b>Appendix C · Working from the Command Line</b>	651
The Terminal Window	651
To View the Operating System Version	652
To Clear the Screen	652
To View the Content of the Current Folder	652
To View the Contents of a Different Folder	653
Using wildcards	654
To Change to Another Folder Location	655
To Copy Files	655
To Rename or Move a File	656
Deleting Files	656
Creating and Removing Folders	657
Running Programs from the Command Line	657
Running git Commands at the Command Line	657
Recalling Previous Commands	657
Creating Batch or Bash Files	657
Conclusion	659
<b>Appendix D · Debugging Your Code</b>	661
Problems Viewing Your HTML or PHP Programs	661
Problems with HTML Layout	663
Locating PHP Syntax Errors	664

---

Common PHP Syntax Errors	665
Common Logical Errors	666
<b>Appendix E · Security and Validation</b>	669
Security and Cross Site Scripting (XSS)	669
Protecting MySQL Databases	672
Validating Input	673
Regular Expressions	674
Useful Code Samples	675
<b>Appendix F · Working with FTP</b>	677
Starting your FTP Server	677
FTP Server Accounts	677
Setting up your local FTP server in Windows (Filezilla)	678
Installing an FTP Client	679
Using Your FTP Client to Connect to Your FTP Server	679
Getting to Know Your FTP Client Interface	680
Closing Your FTP Connection	681
Summary	681
<b>Appendix G · Tracking Your Changes: Version Control Using Git   And GitHub</b>	683
Introduction	683
What is a Version Control System?	684
Git and GitHub	684
Using the Tutorials	685
General Procedure and Terminology	685
Summary of Common Git Commands	687
Introducing GitHub	688
In Conclusion	689
<b>Appendix H · Working with phpMyAdmin</b>	691
Before You Start	691
Connecting to phpMyAdmin	692
Issuing SQL Commands	692
Using the phpMyAdmin Interface	693
<b>Index</b>	695

# 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 strategy, using PHP, CSS, and HTML 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. Some have even indicated that the approach positively changed their opinion of programming as a career and course of study, which is most gratifying.

The book focuses on the primary challenges that face beginning students: an appreciation of language syntax (especially when working between different languages, such as PHP, HTML, CSS, and MySQL); fundamental concepts and structures; common algorithmic requirements; problem-solving and debugging; and perhaps most important, confidence to explore and experiment. There is no intention to provide a complete coverage of HTML, CSS, PHP, MySQL, or OOP, although all of these are introduced in some detail, and the PHP coverage is quite extensive. As a language, PHP offers the programmer some alternative coding choices (such as use of the `print` or `echo` construct, or a choice between double quotes or single quotes); in these cases the book follows some arbitrary conventions in order to minimize differences between PHP syntax and the syntax of other widely used languages, and to facilitate explanation of topics such as escape characters and concatenation. Similarly a simple client/server model is

followed throughout in order to develop familiarity with this type of design. A summary of these conventions is included in Chapter 16 (More About PHP). The last chapter in the book (Where to Go From Here) provides some guidance on next steps, with brief overviews of useful tools, version control systems, Javascript and other languages, OOP, IDEs and frameworks, and potential areas of study and career paths. The textbook appendices are designed to support students in need of additional assistance on various topics, and the textbook Web site provides installation guides, “Hints and Help” FAQs to help with chapter exercises, and hands-on tutorials on a range of general-purpose tools considered essential for any beginning student (including Using the Command Line, FTP, phpMyAdmin, Security and Validation, and Version Control with Git and GitHub).

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

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

## Approach

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

- Important concepts such as client/server design, server-side processing, and interface-driven code modules can be introduced in the form of working applications, and then applied in hands-on exercises.
- Students not only learn the essential control structures and syntax of a programming language, but also learn to use a markup language (and associated style sheets), and a database query language to access and query a

database. This makes sense in today's programming environment where these languages are routinely used in combination to develop a networked application.

- The material is relevant to students across a range of disciplines: Computer Science, Information Systems, Technical Communications, Network Systems, Digital Media, Web Technologies, Mobile Applications, Database Programming, and other technology-related fields.
- The focus on hands-on problem-solving and fundamental structures prepare students for next-level, language-specific courses such as PHP, Python, Java or C++, as well as Web design and database courses, without replicating a great deal of material, while the syntax covered here is generally consistent with these and other languages.

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

Although the book focuses on Web-based applications, there is NO requirement for a network-based programming environment. The book uses a fully functional but standalone Apache Web server (the open source xampp distribution provided by the Apache Friends group) that students can easily install on their Windows, macOS, or Linux computer. Students can begin programming in HTML, PHP and MariaDB or MySQL in literally minutes, and can work in any location with no Internet connection.

## 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, or numerically indexed and associative arrays, 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.

All of the textbook code samples and exercise templates can be downloaded from the companion Web site, and added to the Web server's `htdocs` folder. Students can complete the exercises simply by opening, editing, and saving the appropriate files, and viewing the results in any Web browser, using the `localhost` URL. Assignments can be turned in simply by zipping and submitting the appropriate chapter folder.

## Resources for Instructors

As previously mentioned, the textbook Web site provides a range of resources for students. The Web site also provides support for verified instructors, including test banks, presentation material, and quiz and code solutions. The Web site can be found at:

<https://www.mikeokane.com/textbooks/wbip/>

One or more of the tutorials on the Web site may serve as additional teaching resources for some instructors and courses (much of this material is also provided in textbook appendices but the online versions will be updated, and can be downloaded and edited):

- A short **Command Line** tutorial covers common file addressing and file management and navigation commands.
- An **Introduction to FTP** provides instructions to install an FTP client and run the FTP server that is included in the xampp installation. Students can use this to learn the basic process of exchanging files with a remote server. This tutorial can also be used in combination with the book exercises, for example if the coursework is to be completed on a remote server.

- A **phpMyAdmin** tutorial provides a quick tour of this interface, enough to get students started submitting SQL queries and managing databases, table, and user accounts. This tutorial can be used to provide more extensive coverage of database queries and management, once students have completed Chapter 14.
- A **Version Control** tutorial explains the purpose and general terminology of version control, introduces **git** and **GitHub**, and includes hands-on instructions designed to show student how to use these products, singly and in combination. Note that Version Control is introduced in Appendix G in the book, and students are referred to this tutorial. Some instructors may wish to require students to use these tools throughout the course, at least for on-line backup and recovery.
- A tutorial to introduce **Regular Expressions** is also being considered at this time and may be available by the time of publication.

**Appendix E** in the book provides important information regarding security and validation, and is a must-read for students who choose to work with a "live" online Web server.

## Changes to the Fifth Edition

This fifth edition of the textbook includes: extensive revisions, corrections, and additions to all chapters that will hopefully improve understanding in areas where students tend to struggle; a more general introduction to file and folder management in Chapter 2 that gives equal coverage for Windows, macOS and Linux users; a combination of the HTML `<label tag>` and additional CSS in Chapter 4 to align content in HTML forms; greater use of the `trim()` function to clean input, especially when reading text files; a more structured approach to input validation and security in Chapter 8; increased coverage of superglobal arrays in Chapter 12; a new Chapter 16 to explain a range of additional PHP operators, structures and functions (including data and time functions), and explain conventions used in the book; a new Chapter 17 to cover additional HTML (form elements, lists, quotations and citations, HTML entities, and the use of `<DIV>` and `<SPAN>` tags); an expanded Appendix B, providing more detail regarding file addressing schemes and the use of absolute and relative addresses; a new Appendix C introducing navigation and management at the command line; a new Appendix E to describe basic security and validation procedures, and introduce regular expressions; a new Appendix F that introduces File Transfer Protocol (FTP), and includes a short tutorial; a new Appendix G that introduces version control with Git and GitHub, and refers students to an extended tutorial on the textbook Web site; a new Appendix H that introduces phpMyAdmin and includes a short a tutorial. Stand-alone, hands-on versions of most appendices are available on the textbook Web site, giving students and instructors greater flexibility in the way they are 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 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 file 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 repetition instructions, variables and assignment operations, and arithmetic and logical expressions.

**Chapter 4: Basics of Markup—Creating a User Interface with HTML.** This chapter introduces the topic 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, using a sample style sheet that is used throughout the course. Students also learn how to create and format HTML forms to obtain user input that will be processed by server side PHP scripts.

**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 HTML output. In the process, students learn to code arithmetic expressions, use standard operators, variables, and functions, concatenate strings, and identify and fix 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 the use of IF and IF..ELSE structures, Boolean expressions, relational operators, truth tables, 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 Boolean operators and 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, count and sum selected 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 a group of related data values. The use of the FOR loop to process arrays is explained, and various array-processing algorithms are demonstrated.

**Chapter 12: Associative Arrays and Web Sessions.** 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. Other superglobal arrays are introduced and students learn to combine Web forms and the code to process the forms in a single page. Web sessions are introduced, and students learn how to use the \$\_SESSION array to maintain session data between applications.

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

**Chapter 14: Connecting to a Database—Working with MySQL.** This chapter introduces MySQL/MariaDB and the construction and application of databases queries. 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 query results.

**Chapter 15: Introduction to Object-Oriented Programming.** This chapter introduces Object-Oriented Programming. Examples show how simple object classes are designed, how class variables are encapsulated and accessed by class methods, how objects are instantiated and used in applications, and how classes can be inherited by other classes. An overview of basic OO terminology is provided.

**Chapter 16: More About PHP.** This chapter “fills in” some important topics that were left out of the preceding text in order to avoid too much complexity. Topics include: an extended PHP function list, PHP data types, shortcut operators, WHILE..DO loops, the SWITCH statement, multi-dimensional arrays and ragged arrays, date and time functions. The chapter also clarifies some of the conventions followed in the book (file and variable names, use of quotes, print versus echo statement, etc).

**Chapter 17: More About HTML.** Chapter 17 adds to the HTML that was described in Chapter 4, with special attention to Web forms, quotes and citations, lists, images, HTML entities, and use of <DIV> and <SPAN> containers.

**Chapter 18: Where to Go From Here.** This chapter offers ten suggestions for next steps (these include the textbook tutorials, use of Git and GitHub for version control, Javascript and other languages, OOP, IDEs and frameworks, community engagement, and consideration of career options).

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 help with file addressing schemes (including relative and absolute addresses).

**Appendix C** introduces the use of the command line, and includes a short list of common Windows and macOS/Linux command equivalents.

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

**Appendix E** provides material to address security issues (XSS attacks) and input validation, and includes a brief introduction to regular expressions.

**Appendix F** explains the purpose of the **File Transfer Protocol (FTP)** and provides a short hands-on guide to install an FTP client, run a local FTP server (included in the xampp installation), and use the client to transfer files between a local file system and remote server.

**Appendix G** provides an introduction to **Version Control Systems** using Git and GitHub, and references an extensive tutorial on the textbook Web site.

**Appendix H** provides a brief introduction to **phpMyAdmin**, and includes a short hands-on tutorial.

# Acknowledgments

This textbook could not have been created without the generous help and support of many others. In particular I want to thank Constance Humphries for her invaluable technical advice, proof-reading, and development of video tutorials! My sincere thanks to Scott Sipe, Beth Hall, Susan Trimble and all at Carolina Academic Press for their supportive style, professionalism and experience. Thanks to all those instructors who have used the book and provided invaluable comments and corrections, and especially to Charlie Wallin at Asheville-Buncome Technical Community College, who has taught with the book since its inception and consistently provided invaluable feedback and suggestions, and Fred Smartt, who helped to field-test the first edition. Thank you to my co-workers and the administration at A-B Tech, for fostering such a wonderful teaching and learning environment in an institution that truly cares. And thanks most of all to all of the students who have learned with me and sometimes in spite of me as this book evolved in the classroom. A particular thank you to A-B Tech students Uma Benson, Jean-Jacques Maury, and Kenneth Stanley, who all voluntarily provided me with carefully compiled lists of corrections that were incorporated into the fourth edition. Their engagement with the material and concern for future students is greatly appreciated. Any remaining errors or inconsistencies are of course my own.

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



# About the Author

Mike O’Kane holds a master’s degree in Systems Science (specializing in Advanced Technology) from Binghamton University. He has over twenty 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.

