

Intermediate PHP 5.3 Programming

Course Length: 3 days

Who should attend?

Intermediate PHP 5.3 Programming is designed to teach the ins and outs of developing in one of the world's most popular web development languages. This course is for those who are seasoned web developers and want to take their existing HTML skills and mix them with the PHP skills acquired in this course. This course is not for the faint of heart - there will be SQL database interaction and advanced programming loops and arrays. This is a serious web-programming course that includes lots of hands on lab time.

Prerequisites

- HTML Scripting experience
- Basic concepts of computer programming
- Basic concepts of SQL and how to write a query

Course Objectives

Upon successful completion of this course, students will be able to:

- Describe what PHP is and the advantages to creating Web sites with PHP
- Use variables, arrays, and functions in PHP scripts
- Include other files in PHP scripts and manipulate files and directories
- Process forms and validate their information
- Incorporate cookies and session variables in PHP scripts
- Add, update, and delete records in a database using PHP
- Have a fully working web based application with user access levels

Course Content

Day 1

- Introductions, understanding of the students existing web knowledge
- Introduction to PHP
- Some history
- Discussion of competing technologies (.NET, JSP, Java, Python, Ruby)
- Discussion of the open source movement & its philosophies

All the Players – How do they inter-relate?

- The Operating System
- Apache Web Server
- PHP
- MySQL
- phpMyAdmin
- All-In-One Zend Server CE
- Text editors (Textpad, Vi, Zend, NuSphere, etc)



Basic web Concepts

- The PHP Environment - part 1
- The Client
- The Server(s)
- Basic installation layout (localhost)

LAB # 1

- Display PHP environment information (phpinfo)
- Create project folders
- Write "hello world" php code
- Serve PHP code in web browser

Day 1 – Afternoon

Language Basics

- `<?php ... ?>` php tags, ending commands with ;
- Ways to comment your code
- Variables (Scalar & Array)
- Data types
- Relational Operators
- Assignment Operators
- Echo & print

Logical Language Constructs

- If.. then .. else (elseif)
- Switch.. case (break;)
- Comparison (= =)

The PHP Environment – Part 2

- More on Server environment
- `$_POST`
- `$_GET`
- `$PHP_SELF`

LAB # 2

- Design a basic HTML form and use the Post method to redirect the form to another PHP page that displays the entered information in table format. Receive the following data fields: FirstName, LastName, address, city, and telephone number.
- Design a template form that calculates the sales tax on a dollar value. Allow the user to enter the tax percentage and the dollar amount of the purchase. Using logical program flow, create a single PHP file to collect the information, perform the calculations, and display the results.



- Using programming flow control, determine a customer's status with a bank based on their credit rating or existing balance. Use the following table to determine the course of action, and then display the results in a table format. Use a list box to allow for the entry of a credit code and a single-line entry box to enter a balance. Also, produce a printable form showing the results for an entered customer name, and provide a way to return to the main entry screen once the results are displayed.

Status Code	Balance Amount	Status Level
A	> \$10,000	Preferred
B	> \$1,000	Regular
C	> \$100	Caution
D	< \$100	Avoid!

- Design a form to calculate the amount of money you have in your change dish. Allow for the entry of pennies, nickels, dimes, quarters, 50-cent pieces, and dollar coins. Show the totals in both cents and dollars in a printable table. Also, provide a way to return to the coin entry page if necessary.
- A company from the Northeast wants to base its shipping costs on the area code where an item is shipped. Using the provided table create a data entry form that accepts an area code, then displays the appropriate shipping charges based on the area code and weight of the item (per pound) being shipped.

Region Name	Area code	Price for Region
Boston and area	508, 617, 413	\$2.50 per pound
Maine, New Hampshire, Vermont	207, 603, 802	\$3.50 per pound
Selected Southeast cities	404, 706, 803, 910, 804, 813, 407	\$5.00 per pound
Selected Central State cities	614, 309, 314, 501, 515, 317, 606	\$7.00 per pound
All other cities	All area codes not listed above	\$11.00 per pound

Day 2 - Morning

- Finish any outstanding Lab work
- Questions / clarifications / quick review

Flow Control

- Do While...
- While...
- For... Next

PHP Functions

- Built in Functions – look at php.net
- Rolling your own functions

Arrays

- What are they?
- Defining
- Using, traversing, managing



- Multi-dimensional

LAB #3 – The Mailman Application

System requirements:

Develop an application called Mailman that performs basic mailing list tasks such as recording mailing addresses and printing out mailing labels. Before you start coding, plan the application. First consider the design of the Mailman system, which will include three major components: security logon, system maintenance, and mailing address management. The security logon component is the gateway to the whole system. This component will either grant or deny system access. A typical logon screen will ask the user for a username and a password. The system then verifies that information against information stored in the database. If the user passes this security check, the other parts of the application become accessible. The system maintenance piece keeps the database up to date. Typically you need to add information, edit existing information, or delete information. Although users have already gained access to the system, you can help secure the data through the use of session variables. The data segments that users can change are lists of states and provinces and the user logon data itself. The Mailman system will have two levels of system permissions that are verified each time a user logs on and tries to perform a specific task. Level 1 access is given to system administrators, which means that any users with this level of access can manage this system support data as well as the address information itself. A user with level 2 access is not allowed into the system maintenance area, but is allowed to access the address data. The mailing address management component is the main portion of the system. Any user that gains access to the system is allowed to add, edit, or delete the data here.

- Discuss Requirements & Concepts
- Plan Menus
- Plan Forms
- Plan Database

Day 2 – Afternoon

Cookies

- Why cookies
- Setting cookies

Sessions

- Why Sessions
- Setting sessions
- \$_SESSION

Dates and Times

- UNIX Epoch
- Date functions
- Time functions

Database Interaction

- SQL crash course
- MySQLi
- Connections
- Sending Queries
- Managing Results
- phpMyAdmin tool



Modular Programming

- Re-using code
- Use include() and require function(s)

LAB #4 – Mailman Application

- Start building database
- Build mailman logon screen with edit checks
- Build link menus
- Build add address form with validation
- Build update address form with validation
- Build delete address form with confirmation

Day 3 - Morning

- More lab time to work on Mailman project
- Quick review of day 2's materials

E-mail management

- Setting the e-mail environment properties
- Using the mail function

Lab #5

- More on Mailman application
- Use \$_SESSION to maintain user level permissions
- Build add user module with validations
- Build update users with validations
- Build delete users with confirmation
- Build system reports
 - Show addresses by province
 - Show 2-up (2 column) labels
 - Count addresses by province
 - Prepare e-mail on the birthday of the addressee

Day 3 – Afternoon

Lab # 5 continued...

Further PHP resources

- Zend Corporation
- PHP Conferences
- phplarchitect magazine
- php.net
- apache.org
- mysql.com

Course Wrap up

