

The Art Institute of Atlanta

IMD 410 Dynamic Web Applications

Section A Winter, 2006

Course Outline

Table of Contents

<i>COURSE DESCRIPTION</i>	1
Credits.....	1
Prerequisite	1
<i>OBJECTIVES</i>	1
<i>COURSE INFORMATION</i>	1
Textbooks	1
Recommended Reading.....	1
Supplemental Materials	1
Technology	1
Course Online	2
Room/time	2
Instructor.....	2
<i>COURSE GRADING</i>	2
Assignments	2
Quizzes & Exams	2
Extra Credit	2
Grading Scale.....	2
Submitting Assignments	2
Web Server.....	3
Late Assignments	3
Exams and Quizzes	3
Grading Policy.....	3
<i>AiA ATTENDANCE POLICY</i>	4
Course Attendance Policy.....	4
Attendance Appeals Process.....	4
<i>ACADEMIC HONESTY</i>	4
<i>DISCLAIMER</i>	4
<i>SCHEDULE</i>	5
<i>LAB EXERCISES</i>	7
Lab 1. Introduction to PHP	7
Lab 2. Template System.....	7

Lab 3. Working With Databases	7
Lab 4. Add & Edit Records	7
Lab 5. Upload Files	7
Lab 6. Delete Records, Administrative Home Page.....	7
Lab 7. User Authentication	7
<i>STUDY QUESTIONS</i>	<i>8</i>
Week 1.....	8
Week 2.....	8
Week 3.....	8
Week 4.....	8
Week 5.....	8
Week 6.....	9
Week 7.....	9
Week 8.....	9
<i>ASSIGNMENTS</i>	<i>10</i>
Assignment 1. Jump Page	10
Assignment 2. Research Article	11
Assignment 3. Site Map, Process Flow, Pseudo-Code, and Database Structure for CMS	12
Assignment 4. Content Management System	13
Extra Credit	14

Please note: The contents of this course outline may be revised by the instructor during the quarter. The changes may be made to improve and facilitate the students' achievement of the competencies for the course.

The Art Institute of Atlanta
IMD 410 Dynamic Web Applications

Section A
Winter 2006

COURSE DESCRIPTION

Students apply user-centered design principles, database structures, and server-side scripting to design and develop content for server-based dynamic delivery. The course emphasizes design issues relating to the display of dynamic content on the screen and how that content is updated as well as delivered from databases.

Credits

4 Credits, 6 Hours

Prerequisite

IMD 310 Intermediate Scripting Languages

OBJECTIVES

Upon completion of this course, you should be able to:

- List and describe server components, communication protocol, configuration, and software
- Identify and describe critical components of the server/client relationship
- Identify and use appropriate coding practices for debugging, reuse, and sharing
- Conceptualize a server-side application structure
- Use various advanced variable techniques
- Use advanced programming logic to develop a server-side application
- Identify the benefits of and basic structure of Object Oriented Programs
- Use Object Oriented and related methodologies to create reusable code
- Identify appropriate techniques for dealing with user input and potential security risks
- Create and manage a server-side database
- Integrate a database into a server-side application
- Apply user-centered design principles to the design of dynamic web sites
- Use practical development techniques to make development and updating of a dynamic web application efficient

COURSE INFORMATION

Textbooks

Fuecks, Harry. (2003). *The PHP Anthology I*. SitePoint Pty. Ltd. ISBN: 0-9579218-5-3
Fuecks, Harry. (2003). *The PHP Anthology II*. SitePoint Pty. Ltd. ISBN: 0-9579218-4-5

Recommended Reading

Robinson, Dan. (2000). *Fundamentals of Structured Program Design*. Prentice Hall. ISBN: 0-13-927930-X

Supplemental Materials

You will need more than one means of backing up your files created for this class. Advised methods include: server, jump drive, portable hard drive, laptop, or CD. You will need a three-ring binder with tabbed dividers for your final two projects.

Technology

In order to test all scripts we build in this class you will need a server running a contemporary version of PHP and MySQL. It is advisable to have PHP MyAdmin installed on your server to administer your databases (often installed by default).

Course Online

<http://www.classbot.com/>

Room/time

	Day/Time	Room
Class:	Mon 6 - 7:50 PM	320
Lab:	Thurs 6 - 9:50 PM	216

Instructor

Name:	Aarron Walter
Phone:	770-689-5006
Email:	aarron@classbot.com
Office hours:	Mondays, 12 - 2 PM Thursdays, 5 - 6 PM Or By Appointment Room 217

COURSE GRADING

The final grade will be based on the following scheduled activities:

Assignments

Activity / Assignment	Title	Points	%
Assignment 1	Jump Page	50	5
Assignment 2	Research Project	100	10
Assignment 3	Site Map, Flow Chart, Pseudo-Code, and Database Structure	150	15
Assignment 4	Content Management System	200	20

Quizzes & Exams

Quizzes 1-4		50	5 each
Mid-term		150	15
Final Exam		150	15

Extra Credit

Extra Credit	See details at the end of the course outline.		Value varies
--------------	---	--	--------------

Grading Scale

%	Grade	%	Grade	%	Grade	%	Grade	%	Grade
96-100	A	88-90	B+	78-80	C+	66-70	D+	=< 60	F
91-95	A-	84-87	B	74-77	C	61-65	D		
		81-83	B-	71-73	C-				

Submitting Assignments

Every student will create a personal online “jump” page for this class. The URL for this page should be emailed to the instructor by the end of the second week.

All assignments need to be uploaded to a web server by the stipulated time/date and linked from your personal “jump” page. Each student’s jump page should have the following information on it:

- Your name
- Course number, name and section
- Instructor’s name
- Assignment number
- Assignment title
- Due date (from the course outline)
- Date of submission
- Link to the assignment (typically the assignment title)

Web Server

Please note that all your assignments need to be submitted online. Please remember that servers can go down. If your primary server goes down, please upload your work to a backup server (you could use one of the free web hosting services for this).

Late Assignments

If an assignment cannot be accessed online on the due date, it will not be graded (resulting in a 0 for the assignment). If an assignment is not received on time, it cannot be resubmitted.

Exams and Quizzes

The midterm and the final exams must be taken at the scheduled time/date. Failure to appear for these exams will result in a score of 0 for the corresponding test. Similarly, if you are absent or late on the day of a quiz, you will not be able to take the quiz and will receive a 0 for the corresponding quiz.

Grading Policy

Please note that per the department policy, in order to take the portfolio class and graduate, all students in the Interactive Media Design program need to have a grade of D or higher in all core courses (courses beginning with IMD).

AIA ATTENDANCE POLICY

It is the goal of this policy to improve the academic performance of students in the classroom by stressing the importance of course attendance and reinforcing the work-ready expectations of employers for employee attendance.

Students should be prepared to start the quarter the first day of classes and to drop/add courses early in the first week of the quarter to minimize absences.

The following attendance requirements are in addition to and amend those currently published in the AIA Student Handbook and Daily Planner.

Course Attendance Policy

- Students are required to attend all class meetings, to arrive on time, and to stay for the full duration of the class.
- Students arriving 20 minutes after the start of each class will be marked absent. Students who leave class before the class is over and without the approval of the instructor will be marked absent.
- Students who accumulate more than three absences may be dropped from the course and a grade of “FS,” Fail/Suspension, will be recorded for the course. Students who are suspended receive no refund for the course.

There are no excused absences. Students may verify their attendance with the instructor of the course. In the event a student is suspended from a course and believes an error in recording attendance has been made, he or she should first contact the instructor.

Attendance Appeals Process

Students may appeal a course suspension. Such appeals must be made in writing to the Registrar. Appeals will only be considered when the absences were beyond the control of the student and **all** absences are fully documented.

ACADEMIC HONESTY

As a member of the academic community, students are expected to recognize and uphold standards of intellectual and academic integrity. Under all circumstances, students are expected to be honest in their dealings with faculty, administrative staff, and fellow students. In speaking with any member of the college community, students must give an accurate representation of the facts at hand. Students are required to refrain from any and all forms of dishonorable or unethical conduct related to academic work. In class assignments, students must submit work that fairly and accurately reflects their level of accomplishment. Any work that is not the product of the student’s own efforts is considered dishonest. Engaging in academic dishonesty can have serious consequences for the students. Academic dishonesty includes, but is not limited to, the following:

- Cheating
- Plagiarism
- Submission of the same work in two or more classes without prior approval of the faculty members of the classes involved.
- Submission of any work (full or partial) not actually produced by the student.
- Submission of any work without clear acknowledgement (reference/credit) of the original author or creator of work.

Students proven to have been dishonest in submitting or presenting their work in this class will receive the F (fail) grade for the class. Record of this incident will also be kept in the student’s file. If such an incidence occurs and you would like to file a written appeal, you may do so with the academic director of the department.

DISCLAIMER

All work designed and developed in this class should carry the following statement (typically displayed at the bottom of the page):

This interactive media project was created by students for educational purposes at The Art Institute of Atlanta and is in no way intended for commercial gain or as a source of public information.

SCHEDULE

Date		Readings to be Completed Before Class	Assignments Due	Topic/Activity
Mon Jan 9 (wk 1)	Class			<p>Introduction to course policies, syllabus and each other.</p> <p>Understanding PHP and MySQL PHP capabilities and related technologies Useful web sites Getting a server</p> <p>Understanding Servers Linux, Unix, Windows Apache, IIS A clearer understanding of the Server/Client relationship The four branches of a web application</p>
Thurs Jan 12 (wk 1)	Lab	PHP Anthology I: Chapter 1. PHP Basics http://us2.php.net/tut.php		Authoring environments for PHP Basic PHP structure and placement echo print() phpinfo() require once() Variables Intro to template systems Writing the date and time to the page
Mon Jan 16	Class			No Class: MLK Holiday
Thurs Jan 19 (wk 2)	Lab	PHP Anthology I: Chapter 2. Object Oriented PHP	Assignment 1. Jump Page	Quiz 1 Conditionals: if else and switch Operators: arithmetic, assignment, and comparative Page template system
Mon Jan 23 (wk 2)	Class	PHP Anthology I: Chapter 3. PHP and MySQL		Quiz 2 Introduction to Content Management Systems (CMS) Discussion of databases and data modeling
Thurs Jan 26 (wk 3)	Lab			Introduction to Object Oriented Programming Connecting to a database (procedural approach vs. OOP approach). Looping through records in a database and displaying them in HTML tables.
Mon Jan 30 (wk 3)	Class			Quiz 3 Flow charting visual vocabularies Group exercise: flow charting a process Review for midterm exam
Thurs Feb 2 (wk 4)	Lab			Creating a class to clean all user input through forms
Mon	Class			Midterm exam

Feb 6 (wk 4)				
Thurs Feb 9 (wk 5)	Lab	http://en.wikipedia.org/wiki/Database_normalization http://www.sitepoint.com/article/database-design-management		Construction of add and edit record pages Handling dates Detailed discussion of SQL and data modeling
Mon Feb 13 (wk 5)	Class	http://www.devshed.com/c/a/MySQL/An-Introduction-to-Database-Normalization/	Assignment 2. Research article	Presentation of research articles
Thurs Feb 16 (wk 6)	Lab			Modify add and edit records pages to include file upload feature
Mon Feb 20 (wk 6)	Class	http://en.wikipedia.org/wiki/Pseudocode http://www.phpnoise.com/tutorials/7/1		Quiz 4 Discussion of Assignment 4 Pseudo code Group exercise: draw a flow chart and write pseudo code for a process
Thurs Feb 23 (wk 7)	Lab	PHP Anthology I: Chapter 8. Email		Construction of delete record page Construction of an administrative home page
Mon Feb 27 (wk 7)	Class	http://zend.com/zend/tut/authentication.php PHP Anthology II: Chapter 1. Access Control	Assignment 3. Site map, flow chart, pseudo-code, and database structure	Present site map, flow chart, pseudo code, and database structure for CMS project
Thurs Mar 2 (wk 8)	Lab			User authentication (procedural and OOP approach)
Mon Mar 6 (wk 8)	Class	PHP Anthology II: Chapter 2. XML		Review for final exam
Thurs Mar 9 (wk 9)	Lab			Work in class on CMS project
Mon Mar 13 (wk 9)	Class			Final exam
Thurs Mar 16 (wk 10)	Lab			Work in class on CMS project
Mon Mar 20 (wk 10)	Class		Assignment 4. Content Management System	Presentation of CMS projects.

LAB EXERCISES

Lab 1. Introduction to PHP

Use basic PHP syntax and structure to output information to an HTML page. Use functions to create a simple page template system. Use a PHP function to include content from another page into your PHP page. Use the `php_info()` function to investigate the configuration of your server and its modules.

Lab 2. Template System

Create a PHP template system complete with a dynamic status indication and page heading system. Output the current date and time in your time zone on your template pages. Use an external style sheet to format your pages. Using your new template system, create a small 4 page web site.

Lab 3. Working With MySQL Databases

Use both procedural and object oriented approaches to connect to a database, query a table and retrieve content, looping through it to display it on an HTML page. Use PHP MyAdmin to create a database table with a primary key and unique id.

Lab 4. Creating Classes

Create a PHP class to filter user input through HTML forms.

Lab 5. Add & Edit Records

Create a page with an HTML form that will allow you to add a record to a database, and another page for editing database records.

Lab 6. Upload Files

Modify your add and edit record pages from last week to include a file upload feature. In the case of editing a record with a file associated, delete the existing file before replacing it with a new one.

Lab 7. Delete Records, Administrative Home Page

Create a page that will allow you to delete database records. Your page should first ask for confirmation before deleting the record. Create an administrative home page that will connect to a database and loop through all records, displaying them in a logical order, and dynamically building buttons or links to edit or delete the record (these buttons/links connect to week 5 lab exercise pages). Include an “Add Record” link at the top of your administrative home page.

Lab 8. User Authentication

Using classes, create a user authentication system to protect your CMS from unauthorized users. Compare your class based user authentication system to a procedural approach and list the benefits or pitfalls of each approach.

STUDY QUESTIONS

Week 1

1. What are the most common operating systems used for servers?
2. Explain the process of displaying a PHP page from request to page display in a browser.
3. What do Apache and IIS do?
4. What does open source mean? Identify 5 open source technologies.
5. For what is the function phpinfo() used?
6. Write the PHP code used to write the date into an HTML page.
7. What is an HTTP header?
8. What is the problem with caching associated dynamic pages?
9. What is concatenation?
10. What is a string?
11. Identify what isset, unset, and empty are used for.
12. What sort of data do environment variables store?
13. Identify some web sites where you could get help with PHP problems.

Week 2

14. What are the key features of a CMS?
15. Identify the various types of CMS.
16. What is a cookie? How do you set and delete a cookie?
17. What is a session? How do you set and delete session variables?
18. Identify the various types of operators and give examples of each.
19. What is the difference between =, ==, and ===?
20. Write the basic structure of an if else statement.
21. What is a switch statement? Write an example.
22. What do the keywords exit and break do?
23. What happens if an exit or break statement are used before the HTML page is closed?
24. What is a class? What are the key steps to take to use a class?
25. What are the benefits of using Object Oriented methodologies?

Week 3

26. What is a regular expression and what are they used for?
27. How would you use a conditional to check the success or failure of the mail function?
28. What output is generated when \$_SERVER["PHP_SELF"] is echoed to a page?
29. How can you use \$_GET and \$_POST to catch content passed from forms?
30. What is URL encoding?
31. How are Get and Post different?
32. What does the register_globals flag do? Why is it so important for security and for using forms and variables?
33. How can hidden form fields be helpful when sending form content to a PHP script?
34. What is a constructor function?
35. What are the roles of "getter" and "setter" functions within classes?
36. What is the difference between a public and private method in a class?

Week 4

37. What is user authentication?
38. Draw a flow chart showing how a user might log into a Content Management System.
39. What are the critical components of user authentication?
40. How could you ensure that users cannot access admin pages without signing in first?
41. What does it mean that HTTP is a stateless protocol?
42. How could you use conditionals to validate forms?

Week 5

43. What is MySQL?
44. What is SQL and what is its role in a dynamic web application?
45. What is PHP MyAdmin?
46. How do you insert content into a database?

47. How could you use a loop to display the contents of a database table?
48. Write the basic structure for a while loop and a do while loop.
49. What is an infinite loop? Why is it bad?
50. What does it mean to increment or decrement an index number? How is this done?
51. Write the basic structure of a for loop.
52. What is an array?
53. Identify the PHP functions used to submit an SQL query to a database and to retrieve information.
54. What is the difference between a database and a table?
55. Identify the PHP code used to connect to a database.
56. Why is beneficial to keep database connection code separate in an external file and then included in all pages that need it?
57. Why is a unique id important for all database tables?

Week 6

58. Identify how SQL and PHP could be used together to modify and delete database content.
59. Identify how you could display the contents of a database table in an HTML table.
60. How could you limit the number of records retrieved from a database?
61. How could you use SQL to order the records retrieved from a database?
62. What is a Join?
63. Explain the nature of a one to one data relationship.
64. Explain the nature of a one to many data relationship.
65. Why are unique identifiers so important to proper database structures and their management?

Week 7

66. How would you build a PHP page that could retrieve a users password and email address and email it to the stored email address?

Week 8

67. Identify the HTML form element needed to upload files to a server.
68. What is the \$_FILES array used for?
69. Identify the key information that needs to be tested before the file is uploaded and the conditional statements needed to test the file information.
70. How could you store a title, the file name, and comments about the file in a database?
71. How could you dynamically display all the files that have been uploaded?
72. What security issues must you consider when incorporating a file upload system in your web site?

ASSIGNMENTS

Assignment 1. Jump Page

Obtain server space that supports PHP4 or higher and MySQL. A Linux or Unix based server is preferable. Identify the server operating system, the server software (example: Apache, IIS, etc.). Create a jump page meeting the following requirements:

- Course name and number
- Instructor name
- Your name and email address
- The current date (use PHP to write into page)
- The web hosting company name and web site address (address should be an active link to their site)
- The server operating system
- The server software
- The **exact** version of PHP and MySQL the server is running
- A link to a PHP info page with the following: `<? phpinfo(); ?>`
- Links to all class assignments
- A design theme of your choosing

Anyone planning to use the school server for this class will need to submit a request **no later than week 1** to the instructor to have a database setup and web space allocation increased. If you fail to do so in the first week and are then unable to post assignments because of this, you will receive a 0 for the assignments not posted.

Use an external CSS page to format your page and valid XHTML for the page structure. Table-less layout is preferable.

Post the URL in the Class Bot drop box.

Marking Criteria

Total Points: /50

- Professional presentation (10)
- Usability (10)
- Inclusion of requirements (30)

Assignment 2. Research Article

Choose one of the below research topics and create a 3 page web based research article on your topic. Your research article should have an introduction page presenting an overview of the topic in a compelling manor so as to entice the reader to read on. The second page will be the body of the research paper, which should be **no less than 700 words**. The final page will have links to your research sources and books consulted (**use MLA style citation**).

Include links to examples of your topic and resources available for further reading.

Take a look at the formatting of sitepoint.com articles. Emulate this layout style.

Use the template design techniques discussed in week 2 lab to create your site. Be sure to incorporate dynamic status indicators and headings for each page.

All pages of your site should have the following at the top of each page:

- Title of your research article
- Your name
- The course name and number
- Instructor name
- Date and time file was last modified using the following PHP code:

```
<?
putenv("TZ=Eastern");
echo "This page was last modified: ".date( "F d Y.", getlastmod() );
?>
```

Topics

- Dynamic content in Flash using PHP
- Using the GD library or Image Magic to create images on the fly (thumbnails or watermakrs)
- Using PEAR (choose any extension to focus on)
- Parsing XML with PHP (Using SAX or DOM)
- Creating and Reading RSS feeds with PHP
- Editing text files with PHP
- QuickForm (PEAR Module)
- PHP 5 features
- Sending rich email (HTML email) with PHP (PHP Mailer class)
- Database Normalization
- Ruby on Rails
- A short "how to" on a topic to be discussed with instructor
- Propose a topic of your own!

Consult the instructor if you have another PHP related topic you would like to use for this project.

As with any research project, the content should be your own and not lifted from a tutorial or other source!

Marking Criteria

Total Points: /100

- Professional presentation (25)
- Quality of research (40)
- Assignment requirements (25)
- Quality assurance (10)

Assignment 3. Site Map, Process Flow, Pseudo-Code, and Database Structure for CMS

When creating advanced web applications, it is essential that the structure and scope of the project are clear. Start out by creating a traditional site map using Visio or OmniGraffle (Mac) showing the necessary pages for the front-end portion of the site (the side users see) and the Content Management System. Do not depict process here, just page display.

Map out the necessary tables in your database, their fields, and the data types for each field using Microsoft Word.

Next, draw a flow chart (again Visio or OmniGraffle are the best tools here) depicting the process flow for each portion of the front end of your site (looping through records for display) and the CMS.

Finally, using your flow charts as reference, write the pseudo code for your web site again addressing the front end that the user sees and the CMS.

Create a formal document complete with a cover page, a table of contents, and an executive summary of your web site. Present your work bound in a three ring binder with labeled tabbed separators. Include a tab at the back for code to be added upon the completion of the project.

Note: Please don't place pages in plastic sheets.

Marking Criteria

Total Points: /100

- Site map (20)
- Database structure (20)
- Process flow chart (20)
- Pseudo-code (20)
- Presentation and requirements (20)

Assignment 4. Content Management System

Use your planning documents from the previous assignment as the foundation for the creation of a web site with a Content Management System. You may define your own topic for the site, but be sure it is conducive to the inclusion of the required features. Your topic should require the use of dynamic content and the posting of files via a Content Management System. Any change made in the CMS section should be reflected on the front-end of the site (the side users see).

Your web site should meet the following criteria:

- Use of PHP and MySQL to save, edit, delete, and display dynamic content
- Use standards compliant development techniques (XHTML and CSS)
- Have a secure user authentication system with logout feature
- Allow an administrator to upload files and store a related record in a database
- Use smart forms (forms receive, validate, and store content from one script)
- Use Object Oriented Programming
- Use a well organized and logical code structure with clear commenting
- Use a template system with dynamic headings and status indicators to display pages
- Dynamic display of the date

Submit your code (do not include external style sheets) printed with line numbers, page numbers, and file names then added to the code section of your three-ringed binder from assignment 3. Be sure to include the URL for the project on the cover sheet of your binder.

NOTE: Please make the username “admin” and password “admin” to get into your CMS.

Marking Criteria

Total Points: /200

- Professional presentation (50)
- Assignment requirements (60)
- Code quality (40)
- Quality assurance (30)
- Usability (20)

Extra Credit

- Validate your web site XHTML strict with valid CSS and receive 1 extra percent toward your final grade
- Validate your site as Section 508 or WAI compliant at <http://bobby.watchfire.com> meeting priority II requirements and receive 2 extra percent toward your final grade
- Include a style switcher to the front-end of your site and receive up to 2 extra percent (visit the following URL for example: <http://www.alistapart.com/stories/phpswitch/>)
- Use Flash for your site front-end instead of HTML to display dynamic content and receive up to 2 extra percent

Because of the depth and breadth of the subject, there are many opportunities to venture further into topics or explore entirely new ones. If you have a specific topic you would like to investigate, please propose it to the instructor for approval.