

CPSC-24700 Web and Distributed Programming Course Syllabus

MWF 2pm-2:50pm, Room AS-104-A

Instructor – Eric Pogue

Office location: Virtual [\[link\]](#)
Office hours: Monday & Wednesday noon-1pm and Thursday 11-noon (and by appointment)
Phone: 563-209-7280 (personal mobile phone)
Email: epogue@epogue.com or epogue@lewisu.edu

Special Thanks to:

Dr. Piotr Szczurek, Assistant Professor, Computer and Mathematical Sciences

Course Catalog Description [\[link\]](#)

Languages and technologies for programming and leveraging web-based computer services securely. Languages include PHP, Perl, JavaScript, Java, Ruby, CSS, and HTML5. Technologies include relational databases, web services, Hadoop, and cloud computing platforms. This course teaches students how to develop useful applications using a variety of distributed data and programming models. (3 Credits)

Prerequisites

CPSC 20000 (Introduction to Computer Science) or programming experience

Course Objectives

On the successful completion of this course students will be able to:

- Understand the ideas of distributed computing and the World Wide Web
- Create web pages with HTML5 and CSS
- Understand the best Web design practices
- Create dynamic and interactive web sites using JavaScript
- Understand XML and Web services
- Understand basic database concepts and make simple SQL queries
- Write PHP scripts to process forms and interact with databases
- Utilize a cloud based computing platform (Microsoft Azure) to host a basic website
- Understand the Hadoop framework and MapReduce programs
- Be able to provide an overview of Perl, Java, and Ruby
- Understand the basics of the various software development lifecycle processes

Student Learning Outcomes

By the end of the semester students will be able to:

- Understand the concepts of software development and distributed computing
- Create static and dynamic Web pages using the latest technologies
- Configure and cloud based website
- Utilize server-side scripts and access databases

Textbook

Programming the World Wide Web 8th Edition, Robert W. Sebesta, Addison-Wesley, 2015
(ISBN 978-0-13-377598-3) [\[link\]](#)

Development Environment/Tools

To successfully complete this course, you will need a:

- Windows, Macintosh, or Linux / Unix computer and a place to store programming files
- Web browser
- Text editor
- FTP client
- Microsoft Azure (or alternative website hosting) account

Note that class examples we be done primarily on Windows 10 using the Chrome browser, Microsoft Code editor, the FileZilla FTP client, and the Azure cloud service.

Additional Materials

Lecture videos, assignments and other course materials will be posted or linked via our Blackboard site [\[link\]](#). Please verify your BlackBoard email address, check our course site regularly for updates, and it would be greatly appreciated if you included your picture in your BlackBoard profile.

Course Requirements

Projects

You will have approximately 5 programming projects due during the semester. I consider these projects to be the most important part of the class. The purpose of these assignments is for you to be able to apply the knowledge learned from the lectures and readings to develop concrete real-life programming solutions. You are welcome to suggest alternative projects or changes to the project assignments.

Presentations/Demos

At least once during the semester you will be asked to demo your project and provide an overview how it was developed. This is a time to share and appreciate each other's work. It is intended to be an interactive, positive, and fun experience. It is each of our responsibility to make sure it works out that way for the person presenting and for the group as a whole.

Readings

Before each lecture, you will need to complete any identified readings and be ready to ask questions and actively participate in the discussion.

Lectures/Labs

The course will consist of interactive lectures and labs times where we will work on exercises. Please come ready to participate and to do your part in creating an enjoyable and interactive environment.

Although attendance and participation in lectures/labs is required, I sincerely hope that you will be there and be fully present because you find the experience to be valuable and enjoyable. We each have an obligation to make our time together interesting and It is all of our responsibility to make this happen.

The exercises will typically require you to write a program that executes some task. You are required to save these programs somewhere where you can access it for next time. You should also make sure to complete any missed exercises and have your code ready before the next class.

Quizzes

There will be 4 to 6 quizzes posted on BlackBoard throughout the semester. The main purpose of these quizzes is to test your understanding of material covered by the lectures and the reading materials. You will take these quizzes on your own time through BlackBoard by the required due date. Each quiz consists of 5-10 true/false, multiple choice, or fill-in-the-blank type questions. You will have exactly 20 minutes to complete each quiz. You may retake the quiz twice, but you may have different questions. Your grade will be based on the quiz with the highest score.

Exams

There will be two in-class midterm exams and a final exam. All exam scores will be normalized such that the highest score becomes 100% and the average score is not less than 80%. In case of the second midterm and the final, the exam will focus mainly on material covered since the last exam, but may include questions from material covered by previous exams.

Feedback

Your sincere feedback is requested throughout the semester. We will be meeting for several months which should allow us to make continuous improvement adjustments. Your participation in the final course evaluation survey is also requested; however, the higher priority is for us to make this semester the best it can be.

Programming Comments/Suggestions

For many of you, this course is your first-time doing significant programming. It can be a creative and rewarding endeavor. Feel free to modify or enhance your projects when the spirit moves you. However, it's important to also let me know where you have enhanced the project so that I can fully appreciate your work.

On the other hand, since this is likely your first time doing significant programming, at times you will very likely feel overwhelmed. This is absolutely normal. Take a deep breath, take a break and walk away for a while (be sure to save your work first), and come back when you are able to look at the project with a fresh set of eyes.

How to Be a Successful Programmer (Projects)

1. Start project assignments early
2. Establish a physical work environment that allows you to focus for extended periods of time
3. Become comfortable with your development environment/tools
4. Read or re-read the project assignment and related materials
5. Suggest changes to the assignment if you feel you have found a better way
6. Save and deploy working versions of your project regularly... this allows you to experiment without risking what you have already accomplished
7. Ask for help if you are stuck... often simply articulating the problem/question will lead you to finding your own answer
8. Look for similar examples... but write your own code that you understand
9. Come to class and participate in class exercises... ask questions during or after class
10. Review the textbook and review the lecture slides

Getting Help

The best way to get in touch with me is through e-mail. I almost always reply in less than 24 hours during the week. If you don't receive a reply within 24 hours, don't hesitate to follow-up.

If you need an immediate response, I can be reached via my personal mobile phone provided above. Don't hesitate to call or text, but please also be respectful and use it sparingly.

Please make an effort to "stop in" and say hi during my office hours. I will likely be eating lunch and would greatly enjoy the company. If the scheduled times do not work, we can set up a separate time to talk. I'm also always up for a coffee either before or after class.

Grading Policies

Your final grade in the course will be based on the following:

Projects	40%
Presentation/Demo	10%
Quizzes	10%
Midterm Exam 1	10%
Midterm Exam 2	10%
Final Exam	20%

Final course letter grade will be determined using the following scale:

A	94-100	C	74-76
A-	90-93	C-	70-73
B+	87-89	D+	67-69
B	84-86	D	64-66
B-	80-83	D-	60-63
C+	77-79	F	59 and below

Key Drop Dates:

Friday, Sep 1: Last Day to Drop Fall Class with 100% Refund

Friday, Sep 8: Last Day to Drop Class at 50% Refund

Friday, Nov 3: Last Day to Withdraw from 16-Week Course with a "W" Grade

Course Policies

Class attendance is mandatory. It is your responsibility to know what goes on in class. Students must turn in all assignments and take all scheduled tests. Extensions for assignments and make-up tests will not be given. **Late assignments will not be accepted.**

You will need to regularly check BlackBoard for any new announcements or changes to the course. I may post an announcement about the exam day and time or the programming assignment, etc.

Any form of plagiarism will result in a severe consequence for all parties involved. Please see the "Lewis University Copyright and Intellectual Property Guidelines" document [\[link\]](#).

Specifically for this course, the policy is that you are not allowed to look at anyone else's code to allow someone to look at your code. I reserve the right to use code plagiarism detection software to check for plagiarism occurrences. In any situation in which I have reason to believe that you have copied program code or allowed anyone else to use your code, you will receive, at the minimum, a grade of zero for that assignment and possibly an F grade for the entire course. So, while you may and in fact are encouraged to discuss the assignments with other students, you need to be careful to make sure you write and understand your own code. The same expectations exist for various other course content include quiz and test questions/answers.

Assistance

"If you have a disability that may require consideration by your instructor and you have not previously submitted documentation to the staff in the Leckrone Academic Resource Center (LARC), please make an appointment with Denise Rich, Director of Academic Support Services in LARC (x5593). It is recommended that you address this prior to the start of class or within the first week of class. If you need accommodations for successful participation in class activities prior to your appointment in LARC, you should provide information in writing to your instructor that includes suggestions for assistance in participating in and completing class assignments. It is not necessary to disclose the nature of your disability to your instructor. For more information about academic support services, visit the LARC website [\[link\]](#).

Relationship to Mission

“Lewis University is a Catholic University in the Lasallian Tradition. Our Mission is integrated into all aspects of University life, including this course. This course embraces the Mission of the University by fostering an environment in which each student is respected as an individual within a community of learners. In the spirit of the vision of Lewis University, the goals and objectives of this course seek to prepare students to be successful, life-long learners who are intellectually engaged, ethically grounded, socially responsible, and globally aware.”