Course Syllabus

# Home Department

WDDBS Web Design and Development – Bachelor of Science

# Course Name

Design Patterns for Web Programming

# Contact Information

|  |  |
| --- | --- |
| Catalog Course Code: | WDD 312-L |
| Three-Letter Course Abbreviation: | DPW |
| Instructor: | Rebecca Carroll |
| Telephone: | 407.551.2024 ext. 8595 |
| Email: | [rcarroll@fullsail.com](mailto:rcarroll@fullsail.com) |
| iChat: | fsrcarroll@aim.com |
| Hours: Indicate office hours, chat hours, and preferred method of contact. | [http://ical.me.com/rebecca.carroll/DPW](http://ical.me.com/rebecca.carroll/FFM) |

# Course Description

Design Patterns for Web Programming explores development techniques that go beyond procedural programming, such as object-oriented programming (OOP). Students will be introduced to concepts including composition aggregation, encapsulation, abstraction, and refactoring. Students will also learn objet model implementation of industry standard techniques such as reusability and efficiency.

# Course Materials

* Notebook or other device for note-taking
* MacBook Pro with Python (comes with OSX)
* Google AppEngine
* Editor or IDE for editing code.

# Course Objectives

* Distinguish between procedural and non-procedural programming
  + Understand the benefits of object-oriented programming
  + Understand objects, instances and classes and how to use them to create reusable organized code.
* Understand and utilize encapsulation.
  + Create getters and setters to encapsulate data and functionality
  + Better design and piece together encapsulated components of code
  + Understand the use, limitations and benefits of access modifiers.
* Understand utilize Polymorphism and Abstraction
  + Create abstract classes to template code for reusability and functionality
  + Understand when abstract classes are appropriate to use
* Create web applications using the Python server side language.
  + Understand tiered Application Architecture
  + Understand and use GET and POST methods with forms to request and receive data.
  + Be able to request, receive and synthesize data from outside sources and/or APIs

# Course Outcomes

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

* Create Object-Oriented code in python
* Encapsulate data and functionality
* Utilize inheritance, polymorphism and abstraction with reusable code.
* Use a server side language to create applications for the web

# General Education Component

Design Patterns for Web Programming is greatly supported by College Mathematics course. Programming in any language relies on a base understanding of manipulating variables as provided in the algebra of College Mathematics. Scientific formulas and equations students explore will be required in some of the course work.

# Degree Connection

Just under halfway through the Web Design and Development Degree program (month 11) Design Patterns for Web Programming serves as the student’s introduction to object oriented programming. The class is supported by the programming classes that precede it; namely Web Programming Fundamentals, Programming Web Applications I and II as well as Front-end Frameworks and Web Interaction and Animation courses. The concepts learned are essential as the base for the server side classes near the end of the degree program such as Server-Side Languages and Advanced Server-Side Languages.

# Industry Connection

This class is designed to introduce the basics of programming to the student. While using Python as a learning tool, the elements of language learned in the class can be applied to all of the object-oriented programming languages in use today. In addition to concepts, industry standards and conventions will be covered so that students can produce strict, clean, efficient and well-documented code. The class will be essential to developers seeking to learn this and all other programming languages used in the industry and important to designers who will have to interface their artwork with the dynamically driven sites used today.

# Research Component

With the landscape of web design and development constantly changing it is essential that students are continually researching and referencing documentation for development. While the core concepts and elements will not be different in other languages, some of its vocabulary and syntax will be different. As a result, it is important that students know where to look for these changes in industry documentation. There will also be an assignment that requires students research the use of Object Oriented concepts and solutions in other web languages and technologies.

# Additional Resources

Here is a list of some of the online guides, code documentation and references we will be using throughout the course:

# Topics Covered

* OOP terminology
* Requesting and synthesizing API code.
* Python frameworks
* Encapsulation for data protection and modularization
* Using inheritance, abstraction and polymorphism to make code reusable
* Procedural vs Object Oriented Programming techniques

# Learning Activities

Quizzes

* Short practical quizzes will be administered to examine students’ ability to use concepts and techniques covered in class to solve new problems.
* Written quizzes to assess on core concepts and terminology covered in class.

Lab Assignments

* Labs are small-scale projects where students apply a combination of concepts and techniques learned during lecture.

Practical Exam

* Assesses the ability to execute concepts and techniques learned in class. Students are given possible exams preceding the exam day to prepare them. Notes and previous assignments are permitted for reference as long as they are not collaborative in any way. No internet access will be permitted as airports will be switched off during the exam.

# Grade Weights

Lab/Projects 30%

Quizzes 20%

GPS 10%

Final Exam 40%

Total 100%

Many assignments will have a craftsmanship element to assignments grades, not exceeding 25% in weight. Requirements for good craftsmanship include (but are not limited to) following industry conventions discussed in class, good commenting and formatting of code for readability and reusability.

# Strategies for Successful Learning

* Keep a well-organized set of files with all activities and demonstrations so you can refer back to them as you would with your notes.
* As with any class, taking notes, asking questions, and participating during lecture are invaluable learning tools for every student.
* Students are encouraged to ask questions during the class. There is no such thing as a “stupid question”
* If you find you are having trouble with any part of the class material do not hesitate to get help and do this sooner rather than later.