Note: This syllabus may represent a past offering of this course and future course offerings may differ.

HCDE 310: Communication Design & Technology*

Instructor: Sean Munson
Quarter/Year: Autumn 2012
Course Schedule: MW, 1:30-3:20pm; or
TTh, 10:30am-12:20pm

**DISCLAIMER: This syllabus represents the Autumn 2012 offering, before a prerequisite of CSE142 was added. Future offerings will spend less time on programming fundamentals and more time on (1) applying these fundamentals to design projects and (2) identifying opportunities to use programming to solve problems or add value, and designing systems that act on these opportunities.*

Course Description

This course will cover the basics of programming in the Python language. This course has been designed for students with no or limited prior programming experience. Weekly assignments provide a venue for applying programming concepts. Assignments will focus on data manipulation and the creation of small applications that leverage data and computational resources from larger, publicly available sources. The course culminates in a final project.

Course Objectives

At the end of this course, students should be able to:

- Understand the following programming concepts:
    - Data types
    - Variables
    - Functions
    - Conditional statements
    - Iteration
    - List and dictionary data structures
    - Mapping a function onto a list
    - APIs (application programming interfaces)
- Write programs in Python that demonstrate understanding of all of the above concepts and that use the following features:
    - File operations
    - String processing operations
    - External modules and APIs, including unfamiliar APIs and modules
- Manipulate data to
    - Extract and summarize desired elements
    - Output the processed data in .csv and HTML formats
- Create a web site that generates pages by running Python programs and processing data.
- Apply basic computational thinking, including:

Note: This syllabus may represent a past offering of this course and future course offerings may differ.

- Analyzing and logically organizing data
- Formulating problems such that computers may assist
- Identifying, testing, and implementing possible solutions
- Automating solutions via algorithmic thinking
- Generalizing and applying this process to other problems

Grading and Assignments

Assignments

There will be assignments throughout the course (pretty much every week and sometimes mini exercises during a lecture). Regular assignments allow you to learn the material in small "chunks" and to keep a close eye on how well you understand the material. In some cases, we will do part or all of the assignments during a lecture, though you will submit it later.

Generally, part of the sessions will be lab time. The teaching staff will be there and will circulate. You can confer with other students and the instructors. This is a great time to work on assignments, and you may be able to finish the weekly assignment in the lab — but don't count on it. The previous assignments are typically discussed in the class and we like some time to review your submissions. For this reason, late assignments will receive a zero. You can submit late assignments to get feedback, but no points will be given for an assignment that is turned in after the deadline, unless you have made prior arrangements.

There will be one project in the last 3-4 weeks of the quarter, where you bring all the skills you have learned on one significant problem solving task. You will learn debugging skills, modularity skills, and testing skills. We think the project will be interesting to do and help you pick up life-long problem solving skills that will be useful even beyond programming.

Resource Writeup

As a supplement to the project, you will write up a description of an interesting data source or API that other students might want to use in their final project, or an interesting python module or package. More about this later in the quarter. You are free to use any online module or package to talk about, but one list of many of the Python modules can be found at: Python Global Module Index.

Exam

There will be one exam, most of the way through the course. The first part is a traditional written exam intended to measure mastery of the course material, including programming knowledge; the second is a practical exam, intended to measure programming skills. The exam will be administered during class to allow approximately 2 hours. The exam date is announced well in advance (see the dates at the end of this

Note: This syllabus may represent a past offering of this course and future course offerings may differ.

document). If you have a conflict, please let me know at least 2 weeks in advance so that I can arrange a different time for you.
There's a chance that I will decide to add in-class quizzes to supplement the exam; if so, the quizzes will be announced in advance.

Class Participation (Bonus Points)
Class participation, helping others, interacting on the email list and answering questions, asking good questions that lead to interesting discussions, and pointing out corrections to lectures or code will contribute to bonus points, which you can use to help boost your grade.
In addition, there are optional challenge problems at pythonchallenge.com. I encourage you to try to solve the problems there when you have time and discuss approaches or even code on the email list — that all contributes to class participation points. If you are not able to solve them initially, don't worry. Treat them as optional and fun part of the course. The good thing about the Python challenges is that once you submit a solution to a challenge, you can see several solutions to the previous challenge. It is a learning experience to see how other people approached the same problem.

Grading
The graded work in the course will be weighted roughly as follows to determine a final percentage grade. (Note that bonus points could allow you to get above 100%):

Weekly Assignments	40%
Project:	20%
Python module/package or data source writeup ("resource writeup")	5%
Exams: (I reserve the option to add quizes, which will reduce exam's weight)	35%
Class Participation:	±5%

Also see the HCDE grading policy.

Course Schedule

Week	Topic
1	Introduction to the Course, Computers, and Python. Operations on strings and lists. Version control.
2	File operations; Iteration on sequences Supplement: Expressions and operator precedence; More on strings Conditionals
3	Dictionary data structure; string output formatting
4	Functions. Parameter passing and returning values
5	Indefinite iteration; nested data structure, and nested iteration

Note: This syllabus may represent a past offering of this course and future course offerings may differ.

6	Objects and Classes Mid-quarter course feedback
7	Project intro (30 min) URL lib; Try/except; REST APIs; JSON processing
8	Tuples; Sorting; list comprehensions; enumerate; zip
9	Review Exam (in class)
10	HTML CSS
11	flex day