

## CSC 310.010 - Procedure Oriented Programming Languages, Spring, 2010

CSC 310.010 meets MWF 9-9:50 in Old Main 140. <http://faculty.kutztown.edu/parson/>

Dr. Dale E. Parson, Old Main 260, [parson@kutztown.edu](mailto:parson@kutztown.edu), 484-646-4296

Office Hours: M&W 10-11:30, Tu 2:30-3:30, Th 1:30-2:30, and by appointment

Topics covered include basic concepts and special facilities in higher level programming languages, including scope of declarations, storage allocations, data structure variety, binding time of constituents, and control organization with emphasis on programming in several representative languages. **Prerequisite:** CSC 237.

*Programming Language Pragmatics*, Third Edition, Michael L. Scott, 2006. (PR)

*Python Essential Reference*, Third Edition, David M. Beazley, 2006. (PY) optional

You can reach Steve F. Lott's Building Skills in Python (free) from my course web page. (SL)

**Grading** (on an A-90, B-80, C-70, D-60 scale)

Midterm 25% of grade

Non-cumulative final 25% of grade

Programs 50% divided equally among the programming assignments

**Programming assignment grading criteria**

5% Concise, accurate documentation of the operations of an abstract data type interface signature, preconditions, postconditions, invariants, appropriate use of Javadoc

10% Clean separation of specification of interface from concrete module implementations

5% Documentation of client modules (including tests) that use ADT modules

30% Correct implementation of central data structure or algorithm

20% Adherence to written specification of assignment, including coding standards

5% Willingness to ask questions and clarify any uncertainties about assignments

5% Concise, structured, and accurate documentation of implementation

10% Inclusion of test code, data & results showing coverage of common and corner cases

10% Successful execution against my private evaluation test cases

**The academic integrity policy is at** <http://cs.kutztown.edu/pdfs/AcademicIntegrityPolicy.pdf>

Your first reading assignment is to read the above policy statement.

You may openly discuss ideas, algorithms, pitfalls, and the use of programming tools.

You may not share code, test drivers or test data except within groups for group projects.

Group projects, when assigned, have documented partitioning of student responsibilities.

I will be using the Moss detection system at <http://theory.stanford.edu/~aiken/moss/>.

Class attendance is not mandatory, but I will be teaching fundamentally from experience, using the textbook and some slides as additional sources of information. You are responsible for all material covered in class, including technical information, coding standards and conventions, verbal specification of assignments, and your questions about topics that are not clear to you. Please, there should be no classroom conversations, cell phones, text messaging, eating, sleeping, listening to music or other disruptions of the class. **Consistent lateness to class or disruptive or obscene language will result in a reduction to a student's grade.** Thank you.

## CSC 310.010, Spring, 2010 Syllabus, Topics and Programs

Week	Text	Lecture Topics
1	Lott Basics, PR 1,	Python, compiled and interpreted languages, gmake makefiles
2-P <sub>1</sub>	PR 2.1	Syntax, scanning, Python functions, basic data types
3	Lott Data Structures, Modules	Lists, dictionaries, sets, input/output.
4	PR 3 - <i>Names</i> ,	Binding time, extent, scope, environments, implementation
5-P <sub>2</sub>	<i>Scopes &amp; Bindings</i>	Python implementation of dynamic & statically scoped tables
6	PR 6	Control flow — language constructs and compilation issues
7-P <sub>3</sub>	<i>Control Flow</i>	Python implementation of decision logic using <i>eval</i> .
8	PR 7	<b>midterm (25% of grade)</b> , Data Types — type systems, checking, records, arrays, strings, sets, pointers, lists, dictionaries.
9-P <sub>4</sub>	<i>Data Types</i>	Python built-in types. Persistence and serialization.
10	PR 8 <i>Subroutines &amp;</i>	Calling sequences, parameter passing, generics, exceptions,
11	<i>Control Abstraction</i>	coroutines, events. Python generators, exceptions, events.
12-P <sub>5</sub>	PR 9 <i>Data</i>	Encapsulation, inheritance, method binding, multiple
13	<i>Abstraction, O-O</i>	inheritance, reflection, meta-classes. Python classes.
14	PR 12.1, 12.2	Overview of multithreaded, multiprocessor programming
15		<b>Non-cumulative final exam, 25% of grade</b>

**There are five programming assignments in the plan.**

P<sub>1</sub> requires you to write a multi-function Python program that gets your feet wet.

P<sub>2</sub> requires you to use Python data types to build lexically and dynamically scoped symbol tables.

P<sub>3</sub> requires you to construct and execute a decision tree and to use the Python *eval* function.

P<sub>4</sub> requires you to build persistent, serializable data objects and to pass them between processes.

P<sub>5</sub> requires you to use a threading library wrapper that I supply to emulate coroutines.

Python's home is at <http://www.python.org/>. Any of the 2.x versions is OK. **Do not use version 3!** /opt/csw/bin/python on bill is Python version 2.6.2. **Assignments must run on bill.**