CSC 310.010 - Procedure Oriented Programming Languages, Spring, 2009

CSC 310.010 meets MWF 9-9:50 in Old Main 140.
Dr. Dale E. Parson, Old Main 252, parson@kutztown.edu, 484-646-4296
Office Hours: M 5-6, Tu 10-11, W 10-12, Th 2-3, 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, Second Edition, Michael L. Scott, 2006. (**PR**)
Python Essential Reference, Third Edition, David M. Beazley, 2006. (**PY**)

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, 2009 Syllabus, Topics and Programs

Week	Text	Lecture Topics
1	PY 1-5, PR 1	Python, compiled and interpreted languages, gmake makefiles
2-P_1	PY 6, PR 2.1-2.2	Syntax, scanning, Python functions, eval() and apply()
3	PY 8-10, PR 2.3-2.8	Parsing, formal languages, Python packages, I/O, environment
4-P_2	parser generators	Examine several parser generator packages
5	translation lectures	Example automaton construction and execution
6	PR 3.1-3.4	Binding time, extent, scope rules, implementing scope
7-P_3	PR 3.5-3.11	Referencing environments, aliases, overloading, packages
8	PR 4	Semantic analysis, review, **midterm exam (25%)**
9	PR 5	Target machine architecture, virtual run time machines
10-P_4	PR 6	Control flow — language constructs and compilation issues
11	PR 7	Data types — type checking, polymorphism, built-in data types
12	PR 8	Subroutines and control abstraction — exceptions, coroutines
13-P_5	PR 9, PY 7	Data abstraction and object orientation
14	review	Examples and consolidation
15		**Non-cumulative final exam, 25% of grade**

There are five programming assignments in the plan.

P_1 requires you to write a multi-function Python program that utilizes run-time interpretation.

P_2 requires you to use a parser generator to create a parser for translating sentences in a grammar.

P_3 requires you to construct, maintain and use symbol table data structures for the symbols of $P_{2.}$

P_4 requires you to construct a virtual machine that can run P_2/P_3 programs.

P_5 involves extending P_2/P_3/P_4 to support additional data types and referencing environments.

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 2.5.1. **Assignments must run on bill.**