CSC 237.010 - Data Structures, Fall, 2008 Syllabus

Old Main 136, MWF 8:00-8:50 AM
Dr. Dale E. Parson, Old Main 252, parson@kutztown.edu, 484-646-4296
Office Hours: M 5-6, TuTh 2-3, W 9-10, W 11-12, and by appointment

Topics include basic concepts of data; linear lists, strings, arrays and orthogonal lists; representation of trees and graphs; storage systems, structures, allocation and collections, multilinked structures; symbol tables and searching techniques; formal specification of data structures, data structures in programming languages and generalized data management systems.

Prerequisites: CSC 125 and C or better in CSC 136.


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

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage of Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>25% of grade</td>
</tr>
<tr>
<td>Non-cumulative final</td>
<td>25% of grade</td>
</tr>
<tr>
<td>Programs</td>
<td>50% divided equally among the programming assignments</td>
</tr>
</tbody>
</table>

Programming assignment grading criteria

5% Concise, accurate documentation of the operations of an abstract data type interface signature, preconditions, postconditions, invariants
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. Thank you.
There are seven programming assignments in the plan.

P1 applies data abstraction and recursion to the manipulation of stacks of data.

P2 uses C++ abstract classes for interfaces, and C++ pointers, to implement a sequence ADT.

P3 involves building a C++ interface and two or three implementations for queues.

P4 involves coding, testing and measuring several sorting algorithms.

P5 involves building a C++ interface and two or three implementations for search trees.

P6 entails constructing, testing and measuring hash tables and hashing algorithms.

P7 entails constructing, searching and manipulating a directed graph.