Data Structures (810:052) Spring 2011

Time and Place: 8 - 9:15 AM Tuesday and Thursday in ITTC 322 **and** 8 – 9:50 AM Wednesday in Wright 112

Web-sites: **http://www.cs.uni.edu/~fienup/cs052s11/**

Class Email List: Send messages to 810-052-02-spring@uni.edu from your UNI account

Instructor: Mark Fienup (fienup@cs.uni.edu)
Office: ITTC 313
Phone: 273-5918 (Home 266-5379)
Office Hours: M 9-11:45, 1:10-2; T 2-3:15; W 10-11:45 (← in 112 Wright Hall lab); 1:10-3 (← in ITT 313);
Th 2-3:15; F 9-11:45

Prerequisite: Introduction to Computing (810:051), and co-requisite Discrete Structures (810:080)

Goals: After this course, you should be able to (1) write "medium" sized programs using algorithmic problem solving and functional decomposition in analysis, design, and implementation, (2) implement and understand the algorithms for manipulating the abstract data types (ADTs) stacks, queues, lists, strings, trees, and graphs, and (3) be able to select appropriate data structures when writing medium size programs.

Text: *Fundamentals of Python: From First Programs through Data Structures*, 1st Edition, 2010,
Kenneth A. Lambert, ISBN-10: 1-4239-0218-1, ISBN-13: 978-1-4239-0218-8 (We'll cover chapters 10 to 20.)

Assignments: Assignments will consist of weekly laboratory exercises along with concurrent weekly or bi-weekly programming assignments.

Pedagogic Approach: In class, I'll tend to break up the lecture with active (and group) learning exercises to aid learning. While this is not formally graded, part (5%) of your grade will be based on your participation in these in-class activities. Students benefit by (1) increased depth of understanding, (2) increased comfort and confidence, (3) increased motivation, and (4) being better prepared to work in groups on the job. This might sound great, but it will require you (and me) to work differently to prepare for class. Before the class, you must read the assigned reading, thought about what I asked you to think about, etc.; otherwise you won't be able to effectively participate during class.

Grading policy: There will be three tests (including the final). I'll announce tests at least one week in advance to allow you time to prepare. Tentative weighting of course components is:

In-class Work:	5 %	
Labs:	15 %	
Programming Assignments:	20 %	
In-class Test 1:	20 %	(about Feb. 17)
In-class Test 2:	20 %	(about April 7)
Final:	20 %	(Tuesday, May 3 from 8-9:50 AM in ITT 322)

Grades will be assigned based on straight percentages off the top student score. If the top student's score is 92%, then the grading scale will be, i.e., 100-82 A, 81.9-72 B, 71.9-62 C, 61.9-52 D, and below 52 F. Plus and minus grades will be assigned for students near cutoff points.

Scholastic Conduct: You are responsible for being familiar with the University' Academic Ethics Policies (http://www.uni.edu/pres/policies/301.shtml). Copying from other students is expressly forbidden. Doing so on exams or assignments will be penalized every time it is discovered. The penalty can vary from zero credit for the copied items (first offense) up to a failing grade for the course. If an assignment makes you realize you don't understand the material, ask questions designed to improve your understanding, *not* ones designed to discover how another student solved the assignment. The solutions to assignments should be **individual, original** work

unless otherwise specified. Remember: discussing assignments is good. Copying code or test-question answers is cheating.

Any substantive contribution to your assignment solution by another person or taken from a publication (**or the web**) should be properly acknowledged in writing. Failure to do so is plagiarism and will necessitate disciplinary action. In addition to the activities we can all agree are cheating (plagiarism, bringing notes to a closed book exam, etc), assisting or collaborating on cheating is cheating. Cheating can result in failing the course and/or more severe disciplinary actions.

Special Notice: In compliance with the University of Northern Iowa policy and equal access laws, I am available to discuss appropriate academic accommodations that may be required for students with disabilities. Requests for academic accommodations are to be made during the first three weeks of the semester, except for unusual circumstances, so arrangements can be made. Students are encouraged to register with Student Disability Services, 103 Student Health Center, to verify their eligibility for appropriate accommodations.

Data Structures Spring 2011

Lect #	**Tuesday**		**Thursday**	
1	1/11	Ch. 11: Algorithm Complexity and linear search; Review of Python lists and functions (parameter passing, stack)	1/13	Binary Searching and simple sorts
3	1/18	Simple sorts continued	1/20	Python Summary
5	1/25	Recursion review, divide-and-conquer and dynamic programming	1/27	Ch. 12: Review classes, pydoc, pyunit testing
7	2/1	Ch. 13: Collections overview, array vs. linked tradeoffs	2/3	Ch. 14: Stack Implementations
9	2/8	Ch. 15: Queue implementations	2/10	Stack Applications and Priority Queue implementations
11	2/15	Review for Test 1	2/17	**Test 1**
13	2/22	Ch. 16: Lists types and Positional List Implementations	2/24	Iterators and Advanced sort motivation
15	3/1	Ch. 17: Advanced Sorts: Heap, Merge, and Quick sorts	3/3	Coin-change backtracking
17	3/8	Coin-change dynamic programming	3/10	Ch. 18 Binary tree terminology and operations
	3/15	**Spring Break**	3/17	**Spring Break**
19	3/22	Binary Search Tree (BST): traverals and add operation	3/24	BST remove operation and AVL trees
21	3/29	AVL Trees continues	3/31	Ch. 19 Hashing Introduction
23	4/5	Review for Test 2	4/7	**Test 2**
25	4/12	Dictionary Implementations: ListDict, HashDict, HashTable	4/14	File structures: Disk and file implementation, B+ Trees
27	4/19	Ch. 20 Graph Representation and implementation	4/21	Graph traversals: BFS and DFS
29	4/26	Graph algorithms: Prim's Min. Spanning Tree and Dijkstra's shortest path Algorithms	4/28	Review for Final
Final: Tuesday, May 3 from 8-9:50 AM in ITT 322				