Syllabus
CSE 330: Data Structures
Winter 2012, Zemoudeh
School of Computer Science and Engineering
California State University, San Bernardino

Lecture:	MW 12:00-1:15 in JB-146
Lab:	M 1:30-3:20 in JB-358
Prerequisite:	CSE 202, MATH 272

Instructor:	Kay Zemoudeh, kay@csusb.edu
Office:	JB 347
Office Hours:	M 10:00-12:00 and W 2:00-4:00

Text:	Timothy Budd, "Data Structures in C++ Using the Standard Template Library", Addison Wesley, 1999
Website:	cse.csusb.edu/kay/cs330

Grading:		
	Final	30%
	Midterm	25%
	Assignments	25%
	Lab	20%

Letter Grade Assignment:					
		93-100%	A	90-92%	A-
86-89%	B+	83-85%	B	80-82%	B-
76-79%	C+	73-75%	C	70-72%	C-
66-69%	D+	63-65%	D	60-62%	D-
0-59%	F				

Tentative Syllabus:

Week 1, Ch 1-5:	Review, C++, Algorithm, Time complexity
Week 2, Ch 6, 7:	STL, Strings
Week 3, Ch 8:	Vectors, Generic Algorithms
Week 4, Ch 9:	(Linked) Lists
Week 5, Ch 10:	Stacks, Queues
Week 6, Ch 11, 12:	Deques, Sets, Multisets
Week 7, Ch 13, 14:	Trees, Searching
Week 8, Ch 15:	Priority Queues
Week 9, Ch 16:	Maps and Multimaps
Week 10, Ch 17:	Hash Tables

Final Exam: Wednesday, March 21, 12:00-1:50

Course Description and Objectives
A Data Structure organizes, represents, and manipulates a collection of items.

Examples of data structures are arrays, vectors, stacks, linked lists, trees, maps, hashes, and heaps to name a few. Data structures are one of two integral parts of any program; the other part is the algorithm. One can not study data structures in isolation ignoring the algorithms operating on them. And vice versa, an algorithm cannot be fully developed without considering the data structures involved. A program can be thought of as a transformation on the data structure instances used to solve a problem.

The objective of the course is to learn why and how a data structure is chosen to solve a problem. This learning process is done repeatedly through the exercise of introducing and analyzing a problem, and then converging toward the best data structures and algorithms to solve the problem. Along the way, one must consider and maintain correctness, time and space complexity, brevity, and elegance. In this process other objectives of the course are also realized, in the order of importance, they are:

- To practice Object Oriented Analysis, Design, and Programming (OOA/D/P).
- To get exposed to Standard Template Library (STL).
- To become more proficient in C++ and Unified Modeling Language (UML).

Course requirements:

- Read the text! Read ahead of the class.
- Program, program, program! Get your hands dirty. Learn from your mistakes. Few people learn by watching, most people learn by doing.
- Attend all class and lab sessions. Hand in all assignments even if they are not complete. Incomplete work gets partial credit; no work gets no credit.

Policies

Final and Midterm exams are open book and open notes. The exams will consist of 4-6 problem solving questions similar to the homework questions but at a level to permit for the limited time in the exam. The Final will be comprehensive.

Home works will consist of programming assignments, algorithm design, and other problem solving exercises. Exams and home works must be individual effort. If it is demonstrated that two or more students have collaborated on an assignment, the assignment grade will be divided among those involved. If you find the solution of a problem on the Internet or other publicly available sources, be prepared to share your grade with the other students who have copied the same answer! Any type of cheating or misconduct in an exam will result in an F in the exam and possibly an F in the course and expulsion from the University.

Students with Disabilities

If you are in need of an accommodation for a disability in order to participate in this class, please let me know and also contact Services to Students with Disabilities at UH-183, (909)537-5238.