

CSC 443 - Data Structures

Winter 2004

Instructor : Katy Snyder

Office : Engineering 321

Phone : (313) 993-1136

E-mail : snyderke@udmercy.edu

Course Web Site : <http://knowledge.udmercy.edu>

Office Hrs: After class & by appointment
Mon.-Thurs.: 11:00-1:30

Text : Mark Weiss Data Structures and Algorithm Analysis using C++, 2nd Edition

Software: Microsoft [Visual C++ Version C++ 6.0](#) will be the official compiler for the course. Instruction will be oriented toward those compilers. This software can be found at each of the university computer labs.

Grading : 20% Midterm
35% Final Exam
35% Programming Assignments
10% Written Assignments

Grading Scale :

93-100% A	87-89% B+	77-79% C+	69-67% D+
90-92% A-	83-86% B	73-76% C	60-66% D
	80-82% B-	70-72% C-	0-59% F

Course Description :

Linear lists, stacks, queues, sequential and linked allocation of storage, circular lists, applications. Binary and ordered trees, traversing and

threading trees, garbage collection, multi-linked structures, dynamic storage allocation, data packing, hash coding. Computer projects.

Prerequisites :

Students should have an extensive background in programming, including an extensive knowledge of and object oriented programming in C++ as taught in CSC 441 Object Oriented Programming or CSC 172 Computer Science II

Objectives :

Be able to implement the elementary data structures.

Be able to use these data structures in specific applications, and be able to choose data structures which are appropriate for those applications.

Be able to analyze algorithms and to select the best algorithm for a specific problem.

Be able to use recursion to implement specific algorithms.

Be able to use existing classes and existing class libraries to implement required data structures.

Programming Assignments:

1. Source Language: will be C++. Microsoft [Visual C++ Version C++ 5.0/6.0](#) is the recommended compiler for the course. Students working with other compilers must do so at their own risk.
2. Turn in: For all projects, students should hand in a copy of their source code and output from a sample run of the program, using any test data that have been provided. In addition, the Output should always be date-stamped by the computer, indicating the date on which the output was produced. You should also hand in a disk containing all your source files and an executable program.

3. Grading: Programs will be graded for scope, correctness, style, documentation, attractiveness of output, and timeliness. A program that "works" will not receive full credit unless it is well-written, properly documented, and uses the appropriate style. Please refer to the comments below regarding style.

4. Style: Style is an important component of programming. Programs should employ meaningful identifiers and informative comments. Capitalization may be used provided it is in a consistent, coherent manner. (You may also choose to follow the text and use exclusively lower case identifiers with underscore characters to enhance readability.)

Students should make judicious use of informative comments in their projects. Note that comments that merely repeat the meaning of an instruction are not informative. Comments should explain the purpose and functioning of code, not merely repeat it.

Each program should include a comment at the top giving the name of the project, the author, date, and his/her section number. Some explanation of each function is usually appropriate.

5. Academic Integrity: Students are expected to conform to a high standard of honesty and integrity in this course. Copying the work of someone else and other forms of cheating are strictly prohibited. Permitting or tolerating such behavior is also prohibited. The minimum penalty for any offense is a 0 on that assignment. The culprits may be subject to additional sanctions, up to and including expulsion from school for serious offenses, as prescribed by the University Catalog and the Engineering Science Student Handbook.

Exams :

There will be a midterm exam (20% of final grade) and a final exam (40% of final grade).

Blackboard :

Please visit UDM's blackboard page <http://knowledge.udmercy.edu>.

If you do not already have an account, please click the *Create Account* button to do so.

Click the *Login* button.

To enroll in the course **CSC443-Data Structures** :

1. Select the *Courses* tab
2. Under course catalog, choose: *College of Engineering & Science*.
3. Look through the course listings (or search) for: *MTH140-02*.
4. Enroll in the course. The course **access code** is : huffman
(the access code is case-sensitive!)

Make-up Policy :

Make-ups on exams will only be allowed in extreme circumstances beyond your control. An e-mail prior to the absence (or as soon as possible) is required.

Topics To Be Covered :

1. Implementation and Applications of Linked Lists
2. Introduction to Queues and Stacks, and the Applications to Parsing.
3. Complexity Theory and the Analysis of Simple Algorithms.
4. Recursive Algorithms.
5. Algorithms for Sorting and Searching.
6. Trees, Basic Algorithms and Implementation.

Academic Integrity :

Each student will be expected to meet the standards of academic ethics. Sanctions will be imposed on those who fail to meet these standards according to the *Student Handbook* of the College of Engineering and

Science and the *Student Rights and Responsibilities* publication of the University of Detroit Mercy.

Plagiarism will not be tolerated. Referenced code must be cited. If a determination is made that plagiarism has occurred, all parties involved will receive zero grades and the matter will be referred to the appropriate dean(s).

Dates of Note (test dates are subject to change, check blackboard for up to date schedule):

Wednesday, February 18	Mid-term Exam
Wednesday, March 3	Spring Break -- NO class
Friday, March 26	Last day to withdraw
Wednesday, April 21	Final Exam