

Syllabus for **CMSC 431**
Data Structures/Algorithm
Module II, 2003-2004
Division of Science & Math, Rust College

- Prerequisites:** CMSC 236 or CMSC 333 (C Language or C++ Programming)
- Textbook:** Ford and Topp, *Data Structures with C++ Using STL*, Second Edition. Prentice Hall, 2002, ISBN 0-13-085850-1.
- Class Meeting:** MW 2:30 p.m. to 5:00 p.m., Science Building, room 5.
- Instructor:** Valerie Chu, Ph.D. *e-mail address:* vchu@midsouth.rr.com
URL: <http://biobio.loc.edu/chu/web/>

Course Description:

An introduction on principles of object oriented programming and problem solving with recursion and data containers as abstract data type, including string, arrays, vectors, stacks, queues, linked lists, and binary search trees. Other topics include basic sort and search algorithms.

Objectives:

1. To learn the fundamentals of generic vector, list, stack, queue, and priority queue containers supplied by Standard Template Library. The student will learn when a particular container is appropriate for use in an application.
2. To develop a disciplined approach to problem solving methods and algorithm development.
3. To give the concepts of data abstraction and abstract data types.
4. To teach the basic data structures used in computer science.
 - a. To learn the way to set up pointers and use them.
 - b. To learn the advantages of using linked lists over arrays, vice versa.
 - c. To apply stacks on recursion programs.
 - d. To know the differences between queues and stacks, arrays and vectors.
5. To teach the concepts of object oriented programming.
 - a. To understand the three fundamental features of object oriented programming.
 - i. How *private data type* related to *encapsulation* feature.
 - ii. How *derive class* related to *inheritance* feature.
 - iii. How *template class* related to *polymorphism* feature.
 - b. To know the way to declare and implement a class and declare objects in a class.
 - i. To apply the operator overload concepts on constructors or functions.
 - ii. To use friend functions to access private data.
6. To provide a foundation for further studies in Computer Science.

Course Outline:

<u>Weeks</u>	<u>Chapters</u>	<u>Topics</u>
1	1	Introduction To Data Structures
2	2	Object Design Techniques
	3	Introduction to Algorithms
3	4	The Vector Container
	5	Pointers And Dynamic Memory
4		Review and Test I
5	6	The List Container and Iterators
	7	Stacks
6	8	Queues and Priority Queues
	9	Linked Lists
7	10	Binary Trees
8		Review and Test II
		Review for Final Comprehensive Exam
9	Dec. 15	Final Comprehensive Exam

Course Requirements and Evaluation Procedures:

Several quizzes, two mid-term tests and a final comprehensive examination will be given.

There are no make-up tests except for a valid document from a doctor; however, a note from home is not acceptable.

Programming or written assignments will be assigned frequently. It has to be sent through the e-mail by the deadline (noon of the due day). **Late assignments will receive the following penalties:** 1 day late, minus 10 points; 2 days late, minus 20 points; 3 days late, minus 30 points; ...and so on until zero credit. Students are responsible to check a return e-mail for assignment credits. If a student has turned in assignments and there is no response from the instructor, the student has to contact the instructor directly; otherwise, the student would get a zero credit for the assignment. **Duplication of programming or written assignments will not be permitted. Duplicated programming assignments as well as the original will be assigned a grade of "F".**

The **assignments** and **announcements** will be posted at instructor's *web site*. Students are required to **check announcements every day**.

The score of the final examination can be used to replace the lowest score of the mid-term examinations.

The course grade will be calculated on the following distribution:

Programming & Written Assignments	20%
Quizzes	20%
Mid-Term Tests	40%
Final Comprehensive Exam	20%

Grades will be recorded in numerical form until the final averages are determined at the end of the semester.

Grading Scale:

90 to 100	A
80 to 89	B
70 to 79	C
60 to 69	D
below 60	F