University of Delaware
Computer & Information Sciences

CISC 220 – 010 Syllabus Fall 2000 Data Structures

Course Instructor and TA Info:

	Instructor	Teaching Assistant Name:
	Professor Albright	Joel Schnall
Office:	Smith 101G	Wolf 404
Email:	albright@cis.udel.edu	demenxia@udel.edu
Telephone:	831-2715	831-2770 (Wolf 404 number)
Hours:	MW 1-3pm	Sunday 3-5pm

Class Meeting Times: MWF 9:05 – 9:55 in Gore 102

Catalog Description: Review of data type abstraction, recursion, arrays, stacks, queues, multiple stacks, and linked lists. Emphasis on trees, graphs, tables, sorting and searching.

Course URL: http://www.cis.udel.edu/~albright/220

Course Objectives:

1. Proficiency in writing C++ programs which implement and use stacks, queues, lists, hash tables, trees, (and graphs if time permits);

2. Refine programming skills, including dynamic memory handling, errors handling, use of inheritance, use of recursion, and good software engineering techniques;

3. Thorough understanding of the basic abstract data types, and their representation;

4. Thorough understanding of standard algorithms (methods) for manipulating those representations;

5. Understanding of the functionality and efficiency considerations that dictate the selection of representation in a specific application;

7. Familiarity with a variety of searching and sorting techniques;

Course Prerequisites and co-requisites: CISC181 or permission of the instructor; co-requisite is Math 210 or Math 241.

What you should already know:

1. Programming in C++, including the use of pointers to implement dynamic structures, and the basics of classes (first 9 chapters of Deitel);
2. Familiarity with the UNIX System and C++ compilers:
3. Familiarity with a good text editor (e.g. vi, emacs, hopefully NOT pico - ☺) and compiling tools (CC, g++).

Textbooks:

Robert L. Kruse & Alexander J. Ryba, Data Structures and Programming in C++, Prentice Hall, 1999. (required)

Deitel & Deitel, C++ How to Program, 2nd edition, Prentice Hall 1998 (optional)

Herbert Schildt, C/C++ Programmer's Reference, Osborn McGraw Hill, 1997 (optional)

Graded Work:

Weekly (or almost weekly) homework	20 %
Two program projects	30 %
Several quizzes	10 %
Midterm exam	10 %
Final exam	30 %

Homework will consist of bookwork and/or small programming exercises. These will hone your skills and provide practice is using the rather abstract ideas we will be discussing in class. Homework assignments will be handed out on Fridays and due the following Friday.

Program projects will be larger exercises, sometimes using pieces of homeworks done earlier, and giving you a chance to see something closer to actual applications in practice. Projects will generally take about 2-3 weeks to complete.

Quizzes will be given occasionally – usually following a homework assignment. These will be short, 10 minutes quizzes on either the homework just completed, or some other current aspect of the course.

The midterm will be around the end of October, and will be a full class period. And the final exam will of course be comprehensive.

Late and Missing Work

Late assignments will be penalized 2^n percentage points, where n is the number of days turned in late. If you suspect an error in grading either see the instructor or TA within one week of receiving your graded assignment.

Exams and quizzes will be closed book, closed notes. They are to test what you have learned, not how well you can look things up. If you miss an exam or quiz you will receive a ZERO --- no makeup unless previously excused due to illness, etc.

Programming projects will require a hard copy of the code, sample output and any sample data to be turned in. In some cases the TA or instructor may ask the student to email part or all of the project (more details on this later).

Academic Dishonesty

While working on a program, you may consult with the instructor, TA, classmates, friends, etc., but the actual programming that you hand in MUST be your OWN work. You may discuss in GENERAL terms the GENERAL approach to solving a programming problem. Once the discussion gets down to specific programming issues such as names and types of variables to use, algorithms, control structures such as loops, if-then-else statements, you must end any collaboration!

Specifically, you may **NOT**:

Copy, or allow another student to copy, a computer file that contains another student's assignment, and submit it, in part or in its entirety, as your own.

Work together on an assignment, sharing the computer files and programs involved, and then submit copies of the assignment as one's own individual work.

Edit a script file and then submit it as an original transcript of your computer session.

Any evidence of performing any form of academic misconduct will be appropriately handled as stated in the Official Student Handbook of the University of Delaware. If you are in doubt whether or not a behavior is permitted, then ask me beforehand.

Approximate Schedule of Topics

Topics	Number Of Lectures	Reading
PART I –Linear Structures		
Review of C++	3-4	pp 1-44 in Kruse Pp 192-196 in D&D Chap 6-9,12,13 in D&D
Asymptotics	1-2	Class notes
Lists	4	pp 212-232 in Kruse
Queues	2	pp 78-11 in Kruse And 137-150 in Kruse
Stacks	3	pp 49-77 & 112-136 In Kruse
Applications	2-3	Handouts
Arrays	2-3	Handouts
Midterm Exam	1	
PART II – Non-linear Structures		
Intro to Trees and Recursion	3	
Search Techniques and Intro To Trees	3-4	pp 268-316 in Kruse
Sorting Algorithms	3	pp 317-377 in Kruse
Information Retrieval	5	pp 379-490 in Kruse
Intro to Graphs	3	pp 569-597 in Kruse
Review for Final	1	