CS232 Advanced Data Structures

INSTRUCTOR:	B.J. Streller
OFFICE:	K 154; office hours: MW 11:00-12:00pm,4:30-5:30pm; T 7:00-8:00am; phone: 851.1331 email: streller@ecc.edu email must have CS232 in the subject heading course web page http://wnt.ecc.edu/streller
TEXT:	Data Structures and Algorithms in C++ (4rd ed) by Adam Drozdek ISBN-13: 978-1-133-60842-4 Data Structures and Algorithm Analysis in C++ (4th ed) by Mark Allen Weiss ISBN 0-13-284737-X
RECOMMENDED:	ADTs, Data Structures and Problem Solving with C++ (2nd ed) by Larry Nyhoff ISBN 0-13-140909-3
SOFTWARE:	MS Visual Studio 2010 is available on the computers in K238. Disk Storage Recommended: USB flash drive.

CS 232 is a study of advanced abstract data types (structures) used in Computer Science. The course will include a review of the basic data structures such as stacks, queues (including priority queues), and linked lists.Major topics will include trees, tables, hash tables, graphs, primary and secondary memory, searching and sorting. Measures of efficiency and complexity analysis will be applied throughout the course. A higher level language such as C++, including STL will be used. Several programming projects will be required. Prerequisite: CS132. (From College Catalog)

Upon completion of this course, the student will be able to:
Describe and explain: the concept of an abstract data type (ADT) and different algorithmic strategies. Design, write, execute, and debug programs in C++. Apply measures of efficiency for algorithms and ADTs and interpret the results. Explain and describe the structure representation, and access procedures for ADTs including arrays, stacks, queues, linked lists, trees, graphs, and files. Explain and describe the applications of ADTs including arrays, stacks, queues, linked lists, trees, graphs, and files. Explain and describe different search and internal sorting algorithms. Explain and describe the structure, representation, and access procedures for static and dynamic tree tables and hash tables. Explain and describe the applications of static and dynamic tree tables and hash tables. Identify problems where advanced ADTs are appropriate and select or design the most suitable ADT for the given task. Design,implement,and use advanced ADTs. Identify and perform all phases of the software system development process. Technology Objectives include that students demonstrate a "hand on" proficient using current hardware, software and languages to accomplish all of the above objectives

Program Competencies
Identify all the steps of the software system life cycle and perform problem analysis, the top-down step-wise refinement design process, coding and testing. Write, execute, and debug programs in high-level languages, an assembly language and hybrid programs. Explain the concept of an abstract data type (structure) and design such data types for use in programs. Apply efficiency measures to algorithms and abstract data types and to interpret the results. Write technical documents with an emphasis on good composition and communication skills. This includes documentation that is internal to computer programs and external documentation such as user manuals and programmer manuals. Demonstrate adequate preparation for a career or continuing education.
ECC Learning Outcomes:
Quantitative Reasoning(LV3). Technological Competence.
Learning Resources:
Room K238,Computer Skills Center, one hour per week for class and open hours for work on projects and labs. Library resources are also available.

GRADING
This course will have four (4) one hour exams which will constitute 55 % of your grade. Weekly labs will comprise 15% of your final grade and four (4) projects will make up the remaining 30 %. *In order to pass, the student must have a passing average on all three grading components*

Projects and Labs:
All the assigned projects/labs must be completed in order to receive a grade in the course. Each project must be working correctly, error-free, and well documented when turned in for credit. Students will demonstrate their projects directly to the instructor by appointment (usually during office hours). Each project must be turned in by the designated due date for full credit. Late assignments will lose 10 % per day. Each program will be judged on the basis of working correctly user and programmer manuals program structure, understandability, and form
Grades will be determined as follows:

A (above 90 %) B (80 % -89 %) C (70 % - 79 %) D (60 % - 69 %) F (below 60 %)

Honesty:
Be careful with your projects and assignments; do not leave your printouts, disks, etc., lying around. Each person is expected to do their own work. Each person is to design, write, code, test and debug their own programs, individually. The usual college statement applies to honest work.

There are NO MAKE-UP EXAMS, QUIZZES OR ASSIGNMENTS
'Regular attendance is expected of the student in every course for which he or she is registered' (page2, ECC catalog) Students are responsible for any work covered while they are absent. Excessive absence will affect your grade. Attendance is expected. Class participation and proper student conduct are considered in the final grade.

... if you were unaware

Classroom "etiquette" expectations:

Attend classes and pay attention. Do not ask the instructor in class to go over material you missed by skipping a class
If you must enter a class late, do so quietly and do not disrupt the class by walking between the class and the instructor.
Do not talk with other classmates while the instructor or another student is speaking. If you have a question or a comment, please raise your hand, rather than starting a conversation about it with your neighbor.
Show respect and concern for others by not monopolizing class discussion. Allow others time to give their input and ask questions. Do not stray from the topic of class discussion.
Before entering class turn off all electronics: mobile devices, pagers, and beeper watches. These devices are disruptive to the class and their use during a lecture is rude to the class and instructor.
Focus on class material during class time. Sleeping, talking to others, doing work for another class, reading the newspaper, checking email, and exploring the internet are unacceptable and can be disruptive.
No mobile devices should be in your hands during class. This means no fiddling with your phones, reading/sending text messages, etc.
Penalty for cell phone abuse during class: 5 points/incident subtracted from your final grade A student who really abuses this policy will be given a final grade of F
Do not ask if your lowest grade is dropped - it's not
Do not ask " What are my grades ? " You should save all tests and graded assignments and know how you are doing in the course.
Do not pack book bags or backpacks to leave until the instructor has dismissed class.
It is not required by law that you be in this class. You are free not to attend

... now you are

Topical Outline

I. Introduction and Review

A. Algorithms
B. Data Structures and Abstract Data Types
C. Algorithm Asymptotic Performance Analysis
(Big-O, best, worst, & expected cases)
D. Classes
E. Derived classes and inheritance

II. Review of List ADTs

A. Lists
B. Queues (including priority queues)
C. Stacks

III. Binary Trees

A. Review of Terminology and Definitions
B. Review of Implementations
C. Review of Binary Tree Traversals
D. Review of Ordered Binary Trees
E. Applications
F. Threaded Binary Trees
G. Height-Balanced Trees

IV. General Tree Structures

A. The General Tree ADT
B. Implementations of General Trees
C. 2-3 Trees
D. The Union-Find Problem

V. Graphs

A. Basic Concepts of Graphs and Networks
B. Definition of Graph and Network ADTs
C. Implementations of Graph & Network ADTs
D. Traversals
E. Path Algorithms
F. Spanning Trees
G. Topological Ordering
H. Activity-On-Edge & Activity-On-Vertex Networks (Optional)

VI. Additional Search Strategies

A. Tree-Based Search Techniques
B. Tables
C. Density-Dependent Search Techniques
D. Hashing Applications
E. Indexed Search Techniques

VII. Algorithmic Strategies

A. Definitions
B. Greedy Algorithms
C. Divide and Conquer
D. Backtracking Algorithms

VIII. Sorting

A. Internal
B. External

IX. Memory Management Techniques

A. Memory Allocation and Fragmentation
B. Buddy Systems
C. Heap Management for Programming Languages

X. Evaluation