Data Structures

Course Description

This course provides a thorough study of standard structures used in the storing and retrieving of data and the processes by which these structures are created and manipulated. Topics include: software engineering, object oriented design, linked lists, classes, trees, graphs, hashing, stacks, queues, sorting, searching, and recursion.

Learning Objectives

Upon completion of this course, students will be able to...

1. Understand and use searching and sorting algorithms.
2. Understand the concept of algorithm efficiency and be able to determine the Big-O efficiency of an algorithm.
3. Understand the concept of object-oriented programming through the use of abstract data types.
4. Understand and use dynamic memory allocation to create list, stack, and queue data structures.
5. Understand and use dynamic memory allocation to create tree data structures.
6. Be able to analyze and select appropriate data structures to implement a solution to a problem.
7. Understand and use recursion to solve a problem.
8. Understand the basic concepts of graph data structures and some of the algorithms associated with graphs.

Prerequisites / Co-requisite

The prerequisite for this course is successful completion of Introduction to Problem Solving with Computers II (COSC 1436). The co-requisite for this course is Discrete Structures (MATH 2305).

Major Course Requirements

Projects, Lab Assignments, Homework, Quizzes (40%): All projects, lab assignments, and homework assignments will be submitted electronically through BlackBoard Learn version 9. There will be approximately 3-5 programming projects. Details on project submission will be given to you together with the project assignment. There may be some assigned homework throughout the semester. There may be a pop quiz from time to time.

Exams (60%): There will be three exams (on approximately the 6th, 11th, and finals week).
Required or Recommended Readings


Websites:  
http://www.sci.tamucc.edu/~mscherger/DS  
http://bb9.tamucc.edu

Course Policies

Attendance / Tardiness  
Attendance and active participation during lecture will help you succeed in the course.

Grading Scale  
A=90-100    B=80-89    C=70-79    D=60-69    F=0-59

Late work and Make-up Exams  
All homework and programming projects are due on the assigned date and time. Furthermore, all programming projects must be submitted (and grade received) “in order” (i.e. you cannot submit project 3 without having a grade for project 2). Late projects will be accepted with a 10% penalty per calendar day.

You will have adequate time to complete each assignment (homework or programming project) or exam. However, you should begin working on each assignment early so that you will have plenty of time for debugging which may take significantly longer than the initial code writing. Waiting to start coding until the night before the project is due is a bad idea.

No late homework will be accepted. No make-up quizzes.  
Late programming projects will be accepted with a penalty of 20% per calendar day.  
No late exams. No make-up exams.

Extra Credit  
Some programming projects may have extra credit opportunities available for all students to attempt. These will be described in the homework or programming project description and made available for everyone in the class. Extra credit opportunities will not be made available at an individual student basis.

Cell Phone/Electronic Device Usage  
Please turn off cell phones and other related electronic devices. You may use a laptop during lecture so long as the applications are course related. Please refrain from using email, chatting, or updating social networking sites.

Academic Integrity / Plagiarism.  
University students are expected to conduct themselves in accordance with the highest standards of academic honesty. You are expected to avoid all forms of academic dishonesty as defined in the current Catalog. Academic
misconduct for which a student is subject to penalty includes all forms of cheating, such as illicit possession of examinations or examination materials, falsification, forgery, complicity or plagiarism. (Plagiarism is the presentation of the work of another as one’s own work.) This includes NOT sharing code for the individual lab assignments! If you feel uncertain about a particular activity, please speak to me BEFORE problems arise. You are not allowed to work with someone else to actually solve the assignment, or to write code (even pseudo-code) for a program, and you are certainly not allowed to copy anyone else's solution!

Unless otherwise stated on the assignment sheet, all graded material must be completed individually. Students may give each other general advice, but they may not share algorithms, final answers, or program source code. I strongly believe that discussion with your peers is an excellent way to learn.

Finally, you should be careful not to give others access to your code. This means that you should not keep your program in a publicly accessible directory, you should not leave your terminal unattended, and you should not forget to pick up your printouts.

In this course, academic misconduct or complicity in an act of academic misconduct on an assignment or test will result in failure of the course (grade ‘F’).

Student Computer and Networking Security Statement
Please be sure to read the student computer and networking security statement. A copy (link) of this can be found on our course web page.

Dropping a Class
You must initiate the process to drop the course by going to the Student Services Center and filling out a course drop form. Just stopping attendance and participation WILL NOT automatically result in your being dropped from the class. Please consult the current academic calendar for the last day to drop a class with an automatic grade of “W” this term.

Grade Appeals
As stated in University Rule 13.02.99.C2, Student Grade Appeals, a student who believes that he or she has not been held to appropriate academic standards as outlined in the class syllabus, equitable evaluation procedures, or appropriate grading, may appeal the final grade given in the course. The burden of proof is upon the student to demonstrate the appropriateness of the appeal. A student with a complaint about a grade is encouraged to first discuss the matter with the instructor. For complete details, including the responsibilities of the parties involved in the process and the number of days allowed for completing the steps in the process, see University Rule 13.02.99.C2, Student Grade Appeals, and University Procedure 13.02.99.C2.01, Student Grade Appeal Procedures. These documents are accessible through the University Rules Web site at http://www.tamucc.edu/provost/university_rules/index.html. For assistance and/or guidance in the grade appeal process, students may contact the Office of Student Affairs.

Disabilities Accommodations
The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please call or visit Disability Services at (361) 825-5816 in Driftwood 101.

If you are a returning veteran and are experiencing cognitive and/or physical access issues in the classroom or on campus,
please contact the Disability Services office for assistance at (361) 825-5816.

**Academic Advising**
The College of Science and Technology requires that students meet with an Academic Advisor as soon as they are ready to declare a major. The Academic Advisor will set up a degree plan, which must be signed by the student, a faculty mentor, and the department chair. The College's Academic Advising Center is located in Faculty Center 178, and can be reached at 825-6094.

**Syllabus**

(subject to change)

<table>
<thead>
<tr>
<th>Wk</th>
<th>Lecture/Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Software Engineering Principles, OOD</td>
</tr>
<tr>
<td>2</td>
<td>Recursion, Asymptotic Algorithm Analysis</td>
</tr>
<tr>
<td>3</td>
<td>Stacks</td>
</tr>
<tr>
<td>4</td>
<td>Deques and Queues</td>
</tr>
<tr>
<td>5</td>
<td>TBD, Exam 1</td>
</tr>
<tr>
<td>6</td>
<td>Linked Lists</td>
</tr>
<tr>
<td>7</td>
<td>Binary Trees</td>
</tr>
<tr>
<td>8</td>
<td>Binary Search Trees</td>
</tr>
<tr>
<td>9</td>
<td>Array Based Trees, General Trees, AVL Trees</td>
</tr>
<tr>
<td>10</td>
<td>TBD, Exam 2</td>
</tr>
<tr>
<td>11</td>
<td>Heaps and Heapsort</td>
</tr>
<tr>
<td>12</td>
<td>Advanced Recursion and Sorting</td>
</tr>
<tr>
<td>13</td>
<td>Graphs and Graph Algorithms</td>
</tr>
<tr>
<td>14</td>
<td>Graphs and Graph Algorithms, Hash Tables</td>
</tr>
<tr>
<td>15</td>
<td>Hash Tables</td>
</tr>
<tr>
<td>16</td>
<td>TBD, Exam 3</td>
</tr>
</tbody>
</table>