CS 280: Data Structures
Fall 2015 Course Syllabus

Contact Information

Instructor:	Matthew Mead
Office Hours:	By arrangement (I'm usually here M-F all day and can usually meet anytime I'm not in class.)
Office Phone Number:	895-4441
Email address:	mmead@digipen.edu
Home page (DigiPen):	http://azrael.digipen.edu/~mmead/www/Courses/2015/fall/cs280/index.html

Course Description

The objective of this course is to introduce the classical Abstract Data Types (ADT) in Computer Science. The ADTs provide the hierarchical views of data organization used in programming. Fundamental data structures and algorithms as well as complexity notation are introduced. Note that this course has a strong programming component and builds off of the programming language courses taken in previous semesters. Simply reading about data structures and algorithms and listening to a lecture is insufficient to master and implement these fundamental concepts. Every non-trivial program you write at DigiPen and in the Real World will make heavy use of data structures and algorithms and this course enables you to reason about them and apply them. CS225 is a prerequisite for this course.

Objectives and Outcomes

Upon successful completion of this course, students will understand, implement, and be able to use several data structures and their associated algorithms. Primary data structures that students will work with are arrays, lists, stacks, queues, trees, hash tables, and graphs. In addition, students will be able to apply algorithm complexity and notation in their discussions and evaluations of data structures. Students will have the knowledge to select (and implement) the appropriate data structures that will be used in larger projects, especially the game projects in the third and fourth year.

Day and Time	Room
W/F, 9:00 am - 10:20 am	Michelangelo

Textbooks and References

Recommended

- *Algorithms in C++, Third Edition*, by Robert Sedgewick. Copyright © 1998 by Addison-Wesley Publishing Company, Inc. (ISBN: 0-201-35088-2).

Additional references (Optional)

- *Introduction to Algorithms, Second* Edition, Cormen et al. Copyright © 2001 by MIT. (ISBN: 0-262-03293-7).

Grading

Grades will be derived from homework assignments and exams. The detailed weightings and letter grades are as such:

Homework	40%
Midterm exam	20%
Final exam	40%

You must receive an average score of 60% on both exams combined to pass this course, regardless of your homework scores. Homework scores will only be counted if the above condition is met.

x%	Grade
$x \geq 93$	A
$90 \leq x < 93$	A-
$87 \leq x < 90$	B+
$83 \leq x < 87$	B
$80 \leq x < 83$	B-
$77 \leq x < 80$	C+
$73 \leq x < 77$	C
$70 \leq x < 73$	C-
$60 \leq x < 70$	D
$x < 60$	F

Exams are given only on the day specified during the course. There are no make-up exams.

Attendance is mandatory. For each lecture that is missed, you will lose one point from your final grade (e.g. a 90 becomes an 89). The only exceptions are if you notify me prior to your absence with a valid reason. (Sleeping, studying for another class, working on your game, etc., are not valid reasons for an absence.) Class participation will boost your grade if you are on the border. (e. g. It is possible to get an A- with an overall average of 88.5%)

Tentative Schedule (topic ordering, dates, and/or assignments are subject to change)

Wk	*Topic*	*Possible Tentative Assignments*	*Reference material*
1	Intro to CS280, memory management	Program #1 Memory management	Online
2	Intro to analysis of algorithms (arrays/lists)		Ch. 2
3	Introductory sorting	Program #2 List algorithms	Online
4	Recursion and recursive funtions		Ch. 5
5	Abstract Data Types (ADT), stacks, queues	Program #3 Recursion algorithms	Ch. 3, 4
6	Intro to Trees, BST, AVL Trees, expression trees		Ch. 12
7	**Midterm,** More trees, 2-3-4, splay	Program #4 Tree algorithms	Ch. 13.1 – 13.3
8	Red-black trees		Ch. 13.4
9	Hashing	Program #5 Hashing algorithms	Ch. 14
10	**TBD**		
11	Graphs	Program #6 Graph algorithms	Online
12	Skip lists, B-Trees		Ch. 13.5, 16.3
13	Heaps		Ch. 9.2
14	**TBD**		
16	**Finals week**		

Submitting Homework
Programming assignments will use the C++ language. Although C techniques will also be demonstrated in class as part of the discussions, everyone must be using the same language for assignments, and that language will be Standard C++. (No extensions, Microsoft, GNU, or other). Also, unless otherwise specified, we will not allow the use of the STL for any implementations. A major focus in a data structures class is to learn and understand data structures and their associated algorithms by "constructing the data structures manually." Assignments will be graded using the GNU C++ compiler version 4.8.2, but the assignments will also need to be compiled and tested with other compilers (e.g. Clang) and will be specified in the assignment. You are encouraged to build and run your programs with many compilers, since this is the only way to help ensure that your code is legal and robust.

This is a 4th semester programming course at Digipen. Any compiler warnings or memory leaks/bugs however slight from any compiler/executable will result in a poor grade. Please don't spend a lot of time on an assignment only to receive a poor grade because you were lazy about testing. Assignments that fail to compile will not be accepted and will receive a 0.

Programming assignments are due at the time/day specified and **NO LATE ASSIGNEMNTS WILL BE ACCEPTED.** The source files must be archived in zip format and submitted by the due date/time. It is imperative that you begin thinking about your *professional* career as a game developer as soon as you learn to program. To motivate you towards this goal, up to 25% of the grade on a homework assignment is based on programming quality, clarity, and documentation. Programs that show little or no sign of documentation may receive a very low score. This means that even if you turn in a program that runs perfectly, you can expect a low grade if you fail to adhere to good programming standards. (Documentation samples are posted on the course web page.) Partial credit will be awarded for incomplete assignments. Any additional code documentation requirements will be discussed in class.**Note that we will be using a submission server as was done in all of the previous programming courses.**

There are 6 programming assignments scheduled for the semester, with the first one being assigned during the first week. You will usually be given between 10 and 14 days to complete each assignment. This gives you adequate time to manage your workload. The amount of time actually required to complete an assignment is much less than the time allotted and is generally between 6 and 10 hours. Depending on your grasp of the subject matter during the lectures as well as your performance in CS225 (you should know C++ very well by now), some of you will require more or less time to complete the assignments. Please plan accordingly.

Disabled Student Services
Students with physical, psychological or learning disabilities that affect their ability to perform major life activities associated with this class may be eligible for reasonable accommodations under the Americans with Disabilities Act. If you have a documented disability please contact the Disability Support Services office to arrange for accommodations for this class.

Code Documentation and Conventions
Unless otherwise specified, all programs must be documented using Doxygen tags. There is an extensive help document included with Doxygen which shows examples of each tag. In addition to the "normal" documentation (file and function header comments), you must also put comments next to each header file that you include. The comments must list the functions, types, variables, symbols, etc. that you are using from that library. The purpose of this is for you to demonstrate to the graders and myself that you know why you are including these libraries in your project. Here are a few examples:

```
#include <stdio.h>     /* NULL, printf              */
#include <stdlib.h>    /* atoi                      */
#include <algorithm>   /* sort, transform, for_each */
#include <functional>  /* bind2nd, ptr_fun          */
```

As a rule, absolutely **no implementation by the student is allowed in any header file for any reason**. Please go back and re-read that last sentence.

Workload
During the semester there will be two exams (midterm and final) and 6 homework/programming assignments. The programming assignments are not very large and you will be given adequate time to complete the assignment if you manage your workload properly and start the assignments early. When you submit a programming assignment, another one will be given out. This will mean that you will have a steady-stream of programming assignments throughout the semester. So, in addition to attending the lectures, you can expect to spend 5-6 hours per week programming for this class. Programming assignments may include optional, extra-credit work for students that wish to do it. There are no projects for this class.

Academic Honesty
All homework assignments and exams must represent your own, individual work. It is permissible to discuss assignments (not solutions) with other students in the class, but the solutions must be recognizably your own. Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY (at the discretion of the instructor, which usually results in removal from the class with a grade of F.) Please keep in mind that discussing solutions to exams, quizzes, homework, etc. with students that haven't taken the course or the exam or haven't turned in the assignment is also prohibited. Ultimately, you are only wasting your time (and money) because if you can't master the fundamentals covered in this course, you have little hope of succeeding in other courses or as a programmer in the Real World.

From The "It-shouldn't-need-to-be-said-but..." Department
During class, all electronic devices must be turned **OFF**. This includes cell phones, game consoles, digital cameras, computers, or any other devices. If you absolutely must have a cell phone on for an expected emergency situation, you must put your phone on vibrate and clear it with me **BEFORE** class begins so I know to expect that you may have to leave class early.

In addition to showing up for class on time, other student responsibilities include proper behavior during class, learning the material, completing assignments correctly, submitting assignments properly and on time, studying for the exams, and participating in class by asking or answering questions during the lectures. **All students are required** to bring a pencil (or other writing instrument) and paper to class to take notes and perform other tasks. Finally, no food is allowed in the classroom.