Syllabus

20-CS-2040L Data Structures Programming
TR 8:00-9:30am, Summer 2016
Baldwin 645

Instructor: Prof. Chia Y. Han, 820B Old Chem, 556-1813, chia.han@uc.edu

Required/Elective: Required for CompE majors

Catalog Data: 20 CS 2040L. Data Structures Programming Lab. Credits: 2. Introduction to intermediate and advanced programming methods. Objects, pointers, proper use of dynamic data structures, files.

Prereqs: EECE 1080C, ENED 1091, MATH 1061
Prereqs by Topic: Basic C++ and one semester of calculus

Textbook: Data Abstraction and Problem Solving with C++ -- Walls and Mirrors, by Frank M. Carrano and Timothy M. Henry, 7th edition, Pearson Pub., 2017.

References: Gaddis, *Starting Out With C++ - 7th or 8th Edition,* (the CS1 text)
From Control Structures to Objects or *Early Objects*
Pat Morin, *Open Data Structures - (any version),* opendatastructures.org (pdf)
Plus online resources

Goals: This course is designed to enhance the student's ability to develop software programs to solve typical engineering problems of intermediate complexity. Upon successful completion of this course the student should be able to efficiently and correctly specify, design, implement, and test reasonably complex computer programs in the general, engineering, and embedded systems domains.

Topics Students will implement and test well-designed prorams in the following areas:

1. Development environment; review of pointers, array and linked list data structures
2. Stack data structure (e.g., using a stack to evaluate expression).
3. Queue data structure; circular implementation of a queue
4. Binary tree build, sort, and traverse
5. Heap/priority queue (e.g., simulate a simple digital circuit at the logic level)
6. Graphs: array and linked list implementation
7. C++ features including: overloading, inheritance, virtual functions (e.g., develop a very simple drawing program); scripting

Grading Policy:

Class/Lab Participation (15%), Quizzes (20%), Homework (35%),
Midterm Exam (15%), Final Exam (15%)

Class/Lab: Attendance is REQUIRED. Attendance will be a portion of your quiz grade. Repeated, unexcused absences will negatively impact your grade.

Short in-class hands-on assignments will be given. Bring your computer and be ready to work. Submit you work within 2 hours after the class is done.

Homework Projects: To augment the lab assignments, weekly homework assignments will be given. Homework is due the following Tuesday (either in class or online, as directed).

Videos and Quizzes: Reading materials and online videos will be posted on the Canopy site and announced. They must be watched or read before the next class, when a quiz covering the main concepts and specific details given in the video and the book content.

Late Policy: Late submissions will incur a late penalty of 10% per day late up to 50%.

Plagiarism: Students may discuss programming assignments with each other, but *should not* write *programs together or copy*. Any significant help, from student, book, web, etc. must be acknowledged in a comment in the program. Any duplication of code (without adequate acknowledgement in the documentation) will result in a grade divided by the number of students copying. Group submissions, unless explicitly noted, will be considered copying. The Internet is considered a student in this case. Write your own code and you'll be fine. Document your code with header information and highlighting blocks of code with your personal insights about the key lesson learned in the code.

Cheating on a quiz or exam will result in an F for the course.

Class participation: Use of electronic devices during lecture to take notes or work along with the examples is acceptable. Other uses of electronic devices during class (texting, Facebook, etc) are unacceptable and doing so may result in you being asked to leave the classroom and/or receive negative points in your class participation grade. Active participation by contributing positively to the group learning experience will receive positive bonus points in the overall grade.

Grading: All assignments will contain grading rubrics in advance, so that you know how your work will be evaluated.

ABET Outcomes: a, c, e

Learning Objectives: Students will

1. Program data structures using array-based and linked list implementation (a)
2. Design develop, and test programs using C++ and a standard development environment (a, c, e)
3. Comprehend and utilize object-oriented concept (a)

Computer Usage: extensive use of computer for programming

Professional Component:

Engineering Science: 1 credit (50%)
Engineering Design: 1 credit (50%)

[RE: Syllabi prepared by Prof. C. Purdy, August 12, 2010 and Prof. P. Talaga, Fall 2015]