Scmp 218: Data Structures and Program Design
Fall 2018

General Course Information

Professor: Noah Aydin   **Office:** RBH 319   **Phone:** 5674   **E-mail**: aydinn@kenyon.edu

Course web page: http://www2.kenyon.edu/depts/math/aydin/teach/218

Office Hours: MWF: 11:10-2; TR: 3;10-4, and by appointment. See my weekly schedule on course web page.

Class Meetings: T & R 1:10-2:30 pm in RBH 311

Textbook: Data Structures Using C, by D. S. Malik, Cengage, ISBN 9780324782011
http://www.cengage.com/search/showresults.do?N=16&Ns=P_CopyRight_Year|1&Ntk=all&Ntt=9780324782011

IDE: We will be using CodeBlocks (free software). You can also use DevC++ or Netbeans. Links on course web page.

Course Objectives and Content The goal of this course is to present a detailed coverage of some of the fundamental data structures used in computer science. These include: stacks, queues, lists, heaps, hash tables, trees, and graphs. Searching and sorting algorithms are also examined. Both array-based and linked implementations are covered where appropriate. You will also learn how to use the pre-written classes contained in the Standard Template Library (STL). An introduction to analysis of algorithms and the big-O notation will be given. Analysis of major algorithms will be discussed. Software engineering principles are maintained throughout. We will cover most of the chapters in the textbook.

Grading and Evaluation Criteria:
Final grades will be determined based on the performance in the following components.

Component	Percentage
Quizzes	20
Weekly Labs in Pairs	35
Midterm Exam	15
Participation & Oral Presentations	5
Final Exam	25

Quizzes: Although they won't be announced in advance, you should expect at least one quiz per week, some weeks   there will be 2 quizzes. The total number will be 10+n, with best 10 counting. There will be NO make ups for missed quizzes for any reason (except for long term special circumstances). The in-class quizzes will be used to evaluate and solidify what you have learned from the readings in the textbook. It is imperative that you do the readings *before* each class. We won't have time in class to go over every single detail in the book. You are still held responsible from the material. See the web page for reading assignments.

Labs: The programming projects are the most important aspect of this course. Consequently, they will have the largest weight in the final course grades. Programming assignments will be assigned weekly and some assignment may consist of multiple programs. You will be working in pairs for the programming assignments. See course web page for more information on assignments and pair programming.

Exams: Midterm Exam- The midterm exam is scheduled for Tue Oct 23.
          Final Exam-   Tue, Dec 18, 1:30 pm. Final will be comprehensive and 3 hours long.

Attendance: Regular attendance is expected. No make-up will be given for quizzes. For the midterm exam, make up can only be given with a justified and documented excuse. Each unexcused absence will hurt your participation grade.

Participation and Oral Presentations: After each assignment is due, 2-3 people will be selected to briefly present their solutions. Each student will be asked to do this at least twice during the semester. Make sure you can explain the code you submit for each assignment. Everyone is expected to actively participate in class discussions and activities. Your grade on this component will be based on the combination of your attendance, the level of your engagement in class activities and discussions, and how well you explain your code.

Program Grading: All programs will be graded according to the following components.

Correctness: Each program should conform to specifications stated in the problem statement. A program should demonstrate correct handling of ordinary input, special cases, and error conditions.

Design: Your programs should be modularized into coherent independent functions or classes with strong cohesion.

User Interface: Writing a reasonable test program with good user interface *is always a default requirement for all programming assignments in this course* since this course is also about *program design*. So, this requirement is always part of the assignments. Having solved a problem correctly is not good enough to get full credit. You need to write a good test program and design a good user interface as well. A good test program and a good user interface are not fully prescribed and they may change from program to program. It is something you need to think about for each assignment. An obvious example of a good user interface would be giving the user the chance to repeat a computation before exiting the program (let the user repeat as long as s/he likes). Make sure your program tests all aspects of the assignment. Another point to consider is that asking too much input from the user is not convenient.

Style and Documentation: Your program should be easy to read and understand. This involves program indentation, modular design, variable names, user interface and comments.

Efficiency: Algorithms should be efficient with respect to both time and space. You should spend thinking about designing good algorithms rather than using brute-force. Be prepared to justify your choice of algorithms.

NOTE: If a submitted program **fails to compile** it will be graded out of 50% of the total point value. If a submitted program has a **run-time error**, then it will be graded out of 75% of the original point value.

Late Policy: *No work will be accepted late*. Be aware that the time and date is attached to your work once it is submitted to your folder. Anytime after that, if the file is opened, the date and time change accordingly. **Do not modify your submitted files after the due date**, until graded.

Academic Honesty: At Kenyon we expect all students, at all times, to submit work that represents the highest standards of academic integrity. The rules set forth in the 2018-2019 Course Catalog apply to all aspects of this course.

http://www.kenyon.edu/directories/offices-services/registrar/course-catalog-2/administrative-matters/academic-integrity-and-questions-of-plagiarism/

In general, any work submitted for credit must result from your own understanding, thoughts, and ideas. Presenting the work of others as your own is strictly prohibited. For the weekly programming assignments, follow the guidelines for pair programming carefully. If a partner does not do his/her fair share of the work, please let your professor know.

Disabilities: : If you have a disability that requires an accommodation in this class, please feel free to discuss your concerns with me as soon as possible. Also, you are required to register for support services with the Office of Disability Services in the Olin Library, Center for Innovative Pedagogy. Please contact Erin Salva at 5453 or email salvae@kenyon.edu. Though I am happy to help you in any way I can, I cannot make any accommodations for learning (or other) disabilities without proper authorization from Ms. Salva.

Title IX
Kenyon College seeks to provide an environment that is free of bias, discrimination, and harassment. If you have been the victim of sexual harassment/misconduct/assault, we encourage you to report this. If you report this to a faculty member, she or he is obligated to notify our college's Title IX coordinator about the basic facts of the incident (you may choose whether you or anyone involved is identified by name). The Title IX coordinator will assist you in connecting with all possible resources both on and off campus. Kenyon College's Title IX and VAWA Policy is available at http://www.kenyon.edu/directories/offices-services/title-ix/policy/