Scmp 218: Data Structures and Program Design
Fall 2019

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: 11:10-12; 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
https://www.cengage.com/c/data-structures-using-c-2e-malik/
IDE: We will be using CodeBlocks (free software). You can also use DevC++ or Netbeans. Links on course web page.

Course Description and Learning Objectives: The main goal of this course is to learn basics of fundamental data structures used in computer science and learn how to use them to solve problems. 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. Another important learning goal in this course is to practice software engineering principles and write programs with good user interface. Good programming practices are emphasized. We will cover most of the chapters in the textbook. We will be using pair programming so that you practice software development as a team in a collaborative way. This is often necessary and highly valued by employers. Many real life software development projects are collaborative.
Prerequisite: Scmp 118 or equivalent, proficiency in C++.

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 possibly for long term special circumstances). The in-class quizzes may cover content from the sections that you are expected to read for that day. Therefore, 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 the agenda and reading assignment for each day.

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- Tue Oct 22.
Final Exam- Tue, Dec 17, 1:30 pm. Final will be comprehensive and 3 hours long.

Attendance and Tardiness Policy: After one unexcused absence, each unexcused absence will lower your overall course grade by (n-1)*0.75% where n is the number of unexcused absence. A total of 6 absences (whether excused or not) will result in expulsion from the course. Tardiness and walking out of the classroom are really distracting for everyone. Unless there is a real emergency, please do not leave the classroom before the class is over. Two tardiness or leaving the room during the class will count as an unexcused absence. No make-up will be given for quizzes. For the midterm exam, make up can only be given with a justified and documented excuse. See Math Dept's Class Attendance Policy.

Participation and Oral Presentations: After each assignment is due, 2-3 people will be randomly 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, unless permission is granted by the instructor in advance*. 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 2019-20 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 their fair share of the work, please let your professor know.

Disabilities: If you have a disability which requires an accommodations in this class, please feel free to discuss your concern with me, but you should also consult Ms. Erin Salva, the coordinator of student access and support services (salvae@kenyon.edu, x5453). It is Ms. Salva who has the authority and expertise to decide on the accommodations that are proper for your disability. Though I am happy to help you in any way I can, I cannot grant any accommodations without a notification from Ms. Salva.

Statement on Title IX

Kenyon College seeks to provide an environment that is free of bias, discrimination, and harassment. If you have experienced any form of harassment/misconduct/assault, interpersonal violence, or stalking we encourage you to report it. If you report the incident to a faculty member, they must to notify Kenyon College's Civil Rights and Title IX coordinator of any information about the incident you provide. More information can be found at https://www.kenyon.edu/directories/offices-services/ocr/discrimination/