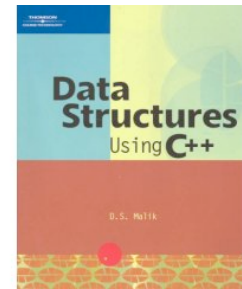


syllabus for
data structures
comp 116

Instructor: **Mark LeBlanc** SC-103 508.286.3970
mleblanc@wheatoncollege.edu Hours: M 2-3:30, R 9:30-11
<http://cs.wheatoncollege.edu/mleblanc>

Meeting Times: MWF 9:30-10:20, W Lab 3:30-5:20
Room A102, Science Center



Text: Malik, D.S (2003). Data Structures Using C++. Thompson.

I also strongly recommend that you buy a 3-ring binder. I'll pass out lots of handouts.

Content: This course is the second half of a year-long introduction to computer science for majors and minors. The content includes an introduction to the theoretical and practical aspects of data structures. Emphasis is on abstract data types and the use of the C++ class mechanism to support object-oriented implementations, including exposure to the Standard Template Library (STL). Examples include stacks, queues, linked lists, and general trees and their applications. Pointers, memory management, and recursion are discussed in length and used in some implementations.

Pedagogy: This course exposes students to the theory of and techniques for “structuring data.” Moving students to a point where they can appreciate object-oriented programming (OOP) is one underlying goal in the course, including of course the reasons for and methods toward encapsulation, information hiding, and reuse. Following practice with multiple data structures from a client perspective, students spend the second half of the course “under the hood” as they learn to implement their own classes. A second goal of the course is to emphasize good software engineering, including steps that lead to verify “program correctness.” Significant portions of time are spent on learning how to write specifications and documentation, including assertions, pre- and post-conditions, and loop invariants.



Your Grade:

Things to do	Grading Percents	Frequency
Labs	10%	weekly
Homeworks	05%	TBA
5 Programs	40%	TBA
Quizzes (approximately 8)	25%	TBA
Final Exam	15%	Thur, May 8, 2pm
Service – Norton Historical Society	05%	semester-long

Late Submissions:

Due is due. Always turn in whatever you have on time. Something turned in on time is much better than not having it accepted because it is late. Late is not an option. (Good, glad we can all agree with this). Note that the Blackboard (Bb) page associated with this course will have submission areas that are time triggered; if you are late, submissions will be blocked. Also, Bb only allows one submission per assignment. If you need to resubmit material and the due date has not passed, email the instructor and I can clear your previous submission.

Honor Code Revisited:

It goes without saying that all submitted work will be the student's own, in keeping with the Wheaton Honor Code, unless the assignment has assigned groups. For labs, you may get “help” from fellow classmates, but remember that all completed work must be your own. Use discretion; don't ask your colleague for “the” answer or for lines of code. However, I do encourage you to discuss the problem in general, such as the type of statements or functions one might use. For homework, your answers and software must be your own from beginning to end. Here is an analogy. Almost no one would ever “use/steal” a line or two from another person's poem. Consider it the same with your programs. Don't “borrow/use” lines or sections of code from another classmate. Your program is (like) your poem; everyone's program should be unique. Be wise. If a colleague is asking you for too much help, be honest and remind them your program is just that, *your* program.

Homework: It is expected that you spend at least 2-3 hours on reading and practice problems for every 50 minutes of lecture. This computes to at least 6 hours of work in the Data Structures text per week. This should be done throughout the semester and not just when studying for quizzes. The material is cumulative in a big way; for example, week 5 depends heavily on weeks 1 through 4. It is expected that you spend at least 6 hours per week on your current programming assignment. **WARNING:** Programmers typically underestimate the time it takes to complete a software project; 6 hours per week on your programming assignment may be one of those “underestimations.”



Labs: The labs are a critical part of the course. In almost all cases, the current lab will be preparing you for the current programming assignment. That is, if you complete and understand the lab, you should be well on your way to a solution for the programming assignment. In order to best grasp the material in some labs, I strongly suggest that you completely redo any labs that you find difficult. (Read that last sentence again, unless of course you've already reread it once).

Quizzes: Rather than exams spaced evenly over the semester, you will be quizzed on a (near) weekly basis. Each quiz may address material that has been assigned in the text, but not yet presented by the instructor. NOTE: you'll have to read the material and practice the sample problems BEFORE coming to class in order to be successful with the quizzes. Quizzes start at the beginning of a lecture and last approximately 10-15 minutes. After some quizzes, immediate (peer) grading will be followed by a discussion which will focus on the concepts in the quiz.

Quizzes continued ... There will be no makeups, nor will the lowest quiz be dropped. If you are an athlete and/or you have a conflict with a quiz date, please see me. When studying for quizzes, have someone pick problems from the notes and text, mix them up, and present them to you as if you were taking a quiz (not telling you from what section a problem comes). While practicing, do not peek at the textbook or an earlier solution. NOTE: its easy to “read” code; its much more difficulty to “write” code from scratch. Quizzes will sometimes ask you to write code from scratch; practice ownership of the concepts!

Service

As a “service-learning” opportunity, we will work together as a “consulting company” to help the Norton Historical Society (NHS) organize and digitize their extensive slide collection. We will discuss this project as the semester progresses and hopefully visit NHS some evening early in the semester so we can meet some of the faithful laborers for Norton history as well as learn of their needs. Together, we will make a work plan and assign tasks. Previous groups of students built a website for NHS: <http://cs.wheatoncollege.edu/mleblanc/nhs>

HELP

I have listed my office hours on the syllabus. But I'm always near a keyboard so we can schedule a time to meet. Study, study, study and talk about it with me as often as you can.

*Please don't wait too long before you see me;
a quick chat in my office can often clear things up.
I'm here a lot...*

*Wondering what you'll be doing with a degree in computer science?
Check out this website for the exciting potential of a career in computing*

Computing Careers

<http://computingcareers.acm.org/> RSS Q

izzle IS Finder Bb SEARCH xLogicCircuits Intro Stonehill (D...g Languages Apple - Scie...nostics, p.1 Apple (147) Amazon

ACM Home

Join | Renew | Subscribe

Association for Computing Machinery
Advancing Computing as a Science & Profession

acm

COMPUTING
DEGREES & CAREERS