

CS 244 Data Structures – Spring 2014

Section 1 meets at: 10:10 AM – 12:10 PM, Tuesday and Thursday
Classes are held in Jarvis Hall Science Wing, Room 345

Course Information: <http://www.uwstout.edu/lit/learn/index.cfm> (D2L login)

Required Text Book: Data Structures & Algorithms in C++, 2nd Edition, by Goodrich, Tamassia, Mount
Software Needed: VMWare Player with Ubuntu Linux and GNU C++ (aka g++)

Instructor: Brent M. Dingle, Ph.D. **Email:** dingleb@uwstout.edu

Office Location: Jarvis Hall Science Wing 219 **Phone:** 715.232.2494

Office Hours: Tuesday/Thursday, 1:30 to 3:00 PM and Wednesday, 12:30 to 2:30 PM*

Also by appointment

** start and end times are ±10 minutes due to various travel times from room to room, and under unusual circumstances may suffer larger deviation, when possible major deviations will be announced ahead of time in class.*

General Information

Provide students the opportunity to increase their proficiency in understanding, constructing, and manipulating data in problem solving. With appropriate effort and dedication to the class, they will gain experience in implementing data structures using C++ and learn how to evaluate algorithm and storage efficiency.

Generalized Learning Objectives**

1. Introduce the students to the C++ programming language and its standard template library (STL) with emphasis on maintaining good software engineering principles.
2. Provide the students with knowledge of basic abstract data types and associated algorithms: stacks, queues, lists, trees, and sorting and searching.
3. Provide the students the opportunity to practice applying this knowledge by specifying and implementing these data structures and algorithms in C++.
4. Provide students with skills needed to understand and analyze performance trade-offs of different algorithms and implementations through asymptotic analysis of their running time and memory usage.

Summary of Expected Learning Outcomes**

By the end of this course, students should be able to:

1. Design and implement different data structures that allow easy access and manipulation of data using the C++ programming language.
2. Apply the Big-Oh asymptotic notation to analyze and select efficient algorithms for solving a given problem with respect to time and memory usage.

Classroom Etiquette

Students are expected to attend every class. Students should bring their laptops to every class. The student, present or not, is responsible for obtaining material and information distributed and presented on all class days. All electronic devices should be set to mute or off before coming to class. This includes, but is not limited to, cell phones, iPods, pagers, PDAs, and laptop volume control. You (the student) may use your laptop (or other electronic devices) to follow along in class. However, the instructor may at any time, for any reason, require, on an individual basis, usage of any device be discontinued. *No photos, video or audio recording is permitted without prior written permission from the instructor.* Smoking of any kind in class is prohibited. Disruptive behavior may result in the instructor requiring its direct and indirect source(s) to leave the classroom for the day. The consequences of which remain the burden of the source(s). Arriving late, leaving early, or sleeping in class is at the student's own risk as are the consequences thereof.

***see university documents for the formally recorded objectives and outcomes, those stated here are emphasis points*

CS 244 – Data Structures – Tentative Schedule

Day	Topic	Noteworthy Items (homework, tests, etc)
Jan 28	Intro. to Data Structures and Ubuntu Linux	Bring laptop to class to install VMware Player & Ubuntu Linux
Jan 30	Types, Variables, I/O, Constants, Arith Opers	
Feb 4	Enumerated Types, Branching, Loops	Homework 1 Due
Feb 6	Loops, Arrays, STL: Objects, Strings, Vectors	Homework 2 Due
Feb 11	Makefiles, Geany, UML, Classes	<i>Optional Prep Homework A1 – Due Date – not graded</i>
Feb 13	References and Pointers, More Classes	Homework 3 Due
Feb 18	Advanced Classes, Dynamic Memory	
Feb 20	Inheritance, Polymorphism	Homework 4 Due
Feb 25	Templates, Big-Oh	
Feb 27	Big-Oh	Homework 5 Due
Mar 4	Buffer – TBD	
Mar 6	Finish Up Summary, Test Review	Homework 6 Due
Mar 11	Unit 1 Test	Online Quiz 1 Due Date, Test 1
Mar 13	Go over Test, Dynamic Arrays	
Mar 18	Spring Break	
Mar 20	Spring Break	
Mar 25	Intro to Sorting and Searching, Linked Lists	
Mar 27	Linked Lists, Stacks	Homework 7 Due
Apr 1	Queues, Deques	
Apr 3	Searches and Sorting	Homework 8 Due
Apr 8	Sorting: Bubble, Merge, others	
Apr 10	Review for Test 2, Buffer – TBD	Homework 9 Due
Apr 15	Unit 2 Test	Online Quiz 2 Due Date, Test 2
Apr 17	Go over Test, Hash Tables, Advanced Sorting	
Apr 22	Advanced Sorting, Trees	Homework 10 Due
Apr 24	Trees, Binary Trees	
Apr 29	Priority Queues, Heaps	Homework 10 Due
May 1	Heaps, Heap Sort	
May 6	Buffer – TBD	Homework 11 Due
May 8	Test Review	Online Quiz 3 Due Date, Last Class – Final Exam Remains
May 13	Final Exam: Tuesday May 13, 10:00 – 11:50	Verify this with the university system, Study

**This schedule is a tentative outline and may be adjusted throughout the semester.
Check the online course content in D2L for schedule changes and course details.**

Deadlines

Tests: Dates are indicated above.

Homework: Due dates and number of assignments may vary with the rate material is covered and the needs of the students. The above indicates what is planned – not necessarily what will be.

Quizzes: Occur “as needed,” but it would be reasonable to expect one sometime before each test.

In-Class Assignments: All dates, if any, indicated in the above are tentative, subject to change, and are NOT all inclusive. Attending all classes is the only way to guarantee none are missed.

Grading:	
	45% Tests (expect 3 tests each of roughly equal percentage value)
	10% Quizzes
	35% Homework
	10% In-class Assignments/Work/Participation

Final Grade:		
A weighted grade of:	96% or above will earn you at least an A	72% or above at least a C
	92% or above at least an A-	68% or above at least a C-
	88% or above at least a B+	64% or above at least a D+
	84% or above at least a B	60% or above at least a D
	80% or above at least a B-	56% or above at least a D-
	76% or above at least a C+	Below 56% is an F

Please note the usage of the words "at least"

Tests

Tentative test dates are given above. Exact dates will be announced in class. Unless otherwise instructed, NO electronic devices will be allowed to be used on test days BUT BRING YOUR LAPTOP anyway. By default, an absence on a test day will result in a zero for the test. However, if properly brought to the attention of the instructor, preferably before the test day, mitigating circumstances may allow a make-up exam to be taken.

Quizzes

Quizzes will be given on an as needed basis and may occur at any time. They may vary between sections. Make-ups will not be allowed after-the-fact, except under extreme circumstances. However, if a student informs the instructor he/she will be missing a quiz/class before actually missing it, leniency may be granted.

Homework

Grading details and turn-in procedures for each homework will be provided with the homework assignment. Unless otherwise stated, homework is due at 11:59 PM on the given due date. Homework will be turned in using D2L's drop box feature for the course. Late assignments will incur a percentage penalty as follows: Let m be the number of minutes late. The percentage penalty is $(m^2)/82944$. So, an assignment that is 24 hours (1440 minutes) late will have a 25% penalty. In such a case a score of 90% would be reduced to 65% and any score of 25% or less would receive a 0. In similar fashion an assignment that is 48 hours (2880 minutes) late will receive a 100% penalty. Turn-in times will be truncated to the minute (example: 1:23:55 PM truncates to 1:23 PM).

Code Documentation

With regard to any course work involving programs or other source code, just because it compiles and produces the 'correct' output does not guarantee a grade of 100% will be awarded. Source code may be examined to verify the way the solution was achieved, or to award partial credit. It is the student's responsibility to make sure the source code is presented in a clear, readable, manner. Appropriate and meaningful comments within the source code as well as good programming style may improve a student's grade. Lack of such may reduce a student's grade.

In-Class Assignments

These are intended to be done and turned in during class. The days they occur and quantity will be as needed and mostly random. It is unlikely any will be accepted late. Due dates will be set on the associated drop boxes in D2L. How long the drop box stays open may vary from assignment to assignment. The only way to guarantee not missing any of these is to attend every class. Grading will be less critical than that applied to homework as it is expected the time spent will be relatively small. Feedback will be minimal/none as most solutions will be provided in class or online.

General Grading of Programming Assignments

In the real world, programs that fail to compile and programs that fail to run to completion receive zero points, and may result in unemployment or worse. In general, unless otherwise stated, such programs in this class will receive between zero and half the maximum number of total points for the assignment. For these cases, unless otherwise stated, an attempt will be made to score the assignment based on examination of the turned-in source code. Penalties for errors will be applied. Then any late penalties will be assessed. It will not always be possible to explicitly quantify this score. This may seem arbitrary, but usually offers the opportunity for more than zero points, as well as a place for the student to start discussion with the instructor about how to fix the errors and learn from them.

Missing Class

This should be avoided. The consequences of any absence shall be determined solely by the instructor on a case by case basis. In general missing a class will result in zero points being awarded for any activities that took place during the missed class period. However, discussion with the student may greatly influence the consequences. This does not imply the instructor must find consequences that the student likes or agrees with. However, more lenient consideration may be given if the instructor is informed BEFORE the absence occurs. Further, the nature of any absence can only be taken into account if and when discussion about the absence occurs.

The responsibility of initiating discussion over any absence falls to the student. In general the student will have seven days after the absence before any consequence of the absence becomes permanent. However, near the end of the semester, or as a necessity of providing timely entry of grades, or for other reasons mostly beyond the instructor's control, this grace period may be shortened without notice.

Academic Dishonesty

Students are expected to do their own work unless specifically directed otherwise by the instructor. Plagiarism and cheating are serious offenses and may be punished by failure on assignment, failure on exam, failure in course, and/or expulsion from the University. For more information, refer to the [university policy](#).

Incompletes and Withdrawals

By [university policy](#), incompletes will only be given in circumstances that do not allow a student to finish the class, and only if it is beyond the student's control. Poor performance is not a condition for an incomplete. Withdrawals (dropping the class) will be allowed in accordance with [university policy](#).

Special Needs

UW-Stout strives for an inclusive learning environment. If you (the student) anticipate or experience any barriers related to the format or requirements of this course you should meet with the instructor to discuss ways to ensure full access. If you determine that additional disability-related accommodations are necessary please contact the Disability Services office (206 Bowman Hall, 232-2995, www.uwstout.edu/disability).

Helpful Links

C++ Reference - <http://www.cplusplus.com/reference/>
STL Reference - http://www.sgi.com/tech/stl/table_of_contents.html
Additional C++ Info - <http://www.stroustrup.com/Programming/>
Sorting Algorithms - <http://www.sorting-algorithms.com/>

Escape Clause

Extreme circumstances such as, but not limited to, natural disasters, campus tragedies, and acts of terrorism or warfare, may require the modification of part or all of the above terms with respect to none, one, some, or all parties. Such modification would be subject to university policy and the general concepts of fair, balanced, and just with respect to all parties involved. Final decisions, on what that means and the modifications necessary, are held by the instructor to the extent university policy allows. In the event the instructor is not physically or mentally able to make such decisions then they will pass to the department head to which the instructor belongs, again subject to university policy.