

CS 2308: Foundations of Computer Science II

Fall 2011

Section 0003

Instructor: Jill Seaman
Nueces 221
245-4706
js236@txstate.edu

Course Webpage: <http://www.cs.txstate.edu/~js236/cs2308>

Office Hours: M: 11:00AM – 12:00PM
T: 1:00PM – 1:30PM
Th: 9:30AM – 10:30AM
Subject to change (if so you will be notified).

Meeting Time/Place: TH 11:00AM-12:20PM DERR 235

Open Labs: DERR 231: Linux Lab
MCS 590: Windows Lab

Text: Tony Gaddis, *Starting out with C++: From Control Structures through Objects*, 6th Edition, ISBN: 0321545885

List of recommended/required readings:
Chapters 1-7 (review of CS 1428) (recommended)
Chapters 8,9,10,11,13,17,18 (required)

Prerequisites: CS 1428

Course Description: Fundamentals of object-oriented programming. Introduction to abstract data types (ADTs) including lists, stacks, and queues. Searching and sorting. Pointers and dynamic memory allocation. A continuation of CS 1428.

Course Objectives:

1. Develop and use appropriate algorithms.
2. Know that there are typically many algorithms for the same task (for example, searching and sorting).
3. Implement a divide-and-conquer algorithm to solve an appropriate problem (binary search).
4. Have an introductory knowledge of the time/space efficiency of various algorithms.
5. Understand structured programming in terms of modules and functions.
6. Understand how to separate source code into multiple files, including header (.h) files.
7. Use pointer variables and memory operations.

8. Resize an array with dynamic memory allocation.
9. Delete allocated memory to avoid memory leaks.
10. Create and use simple linked-lists.
11. Insert into, delete from, and traverse a linked structure.
12. Understand the principle of the Abstract Data Type (ADT) and, in particular, the separation of interface and implementation.
13. Implement user-defined data structures in a high-level language.
14. Compare and contrast the costs and benefits of dynamic and static data structure implementations.
15. Have an introductory understanding of object-oriented programming.
16. Write a program using an array of objects.
17. Design, implement, test, and debug simple programs in an object-oriented programming language.
18. Describe how the class mechanism supports encapsulation and information hiding.
19. Write programs that use each of the following data structures: arrays, structures, strings, and linked lists.
20. Describe and understand concepts of Stacks and Queues.
21. Be able to create, compile, and run a program in a Unix style, command-line environment.

Grading:	Attendance:	5%	
	Programming Assignments:	25%	6-8 total
	Exam I:	20%	Oct 6
	Exam II:	20%	Nov 10
	Final Exam (comprehensive):	30%	Dec 8, 11:00am-1:30pm

Attendance: Each day without an exam counts 1 point. The first three absences do not count against you.

Makeup Policy: Programming assignments cannot be late or made up. Exams may be made up or rescheduled in exceptional circumstances, with documentation and/or prior approval from the instructor.

Withdrawals/drops: You must follow the withdrawal and drop policy set up by the University and the College of Science. You are responsible for making sure that the drop process is complete.
<http://www.registrar.txstate.edu/registration/drop-a-class.html>

Notifications from the instructor: Notifications related to this class will be sent to your Texas State e-mail account. Be sure to check it regularly.

Classroom Behavior: The main rule is to not disrupt other students during class. Please do not arrive late or leave early (without prior permission from the instructor). Cell phones, ipods, etc. should be kept out of sight and turned off or on vibrate during lecture. Laptops are ok for note-taking ONLY.

Academic Honesty: You are expected to adhere to the University's Academic Honor Code as described in <http://www.txstate.edu/effective/upps/upps-07-10-01.html>. Also see the Texas State Student Handbook. Unless otherwise stated, all assignments are to be done individually. You may discuss general strategies for attacking assignment problems with other students in the class but you must write your own code. Your submitted programs may be run through an internet service designed for detecting plagiarism in software code.

Accommodations for students with disability:

Any student with a special needs requiring special accommodations should inform me during the first two weeks of classes. The student should also contact the office of disability services at the LBJ student center.