Section 1: TTH noon - 1:20 PM

class web page:http://www.csc.lsu.edu/∼duncan/courses/csc3102-s20

Instructor: William E. Duncan, PhD
Office: 3270C Patrick F. Taylor Hall
Email Address: duncan@[**bit.cse** or **csc**].lsu.edu
Office Hours: http://www.csc.lsu.edu/∼duncan/sched.htm

Course Description:

Description and utilization of formal abstract data type representations, especially of sequences, trees and graphs. Time and space analysis of recursive and non-recursive algorithms. Sorting algorithms, graph algorithms and algorithm design techniques. – 3 sem. hrs.

Required Textbook:

Algorithm Design and Applications 1/E,
Michael T. Goodrich and Roberto Tamassia (ISBN:978-1118335918)

Supplementary Reading:

- Data Structures and Algorithm Analysis in C++ 4/e,
  Mark Allen Weiss (ISBN:978-0132847377)
- Data Structures and Algorithm Analysis in Java 3/e,
  Mark Allen Weiss (ISBN:978-0132576277)

Prerequisites:

CSC 1254 or CSC 1351 and credit or concurrent enrollment in CSC 2259 or EE 2740.

Learning Objectives:

To explore formal methods of analyzing algorithms and data structures so that students can make appropriate choices when designing programs. By the end of this course, students will be able to

- Use advanced programming techniques such as functional cohesion and data abstraction in implementing advanced classical data structures,
- Use asymptotic time complexity analysis to evaluate associated algorithms in order to select from a range of possible options, provide justification for that choice, and implement the algorithm in a particular context, and
- Use various algorithm design techniques such as brute force, greedy, divide-and-conquer, and dynamic programming.

Evaluation:

Grading will be based on homework assignments, programming projects and exams.

Homework Assignments	20%
Programming Projects	25%
Midterm Exam	25%
Final Exam	30%

Final grade will be determined by overall average as follows:

Table 1: Percentage Score to Letter Grade Assignment

SCORE	GRADE	SCORE	GRADE	SCORE	GRADE
90-92.99	A-	93-96.99	A	97-100	A+
80-82.99	B-	83-86.99	B	87-89.99	B+
70-72.99	C-	73-76.99	C	77-79.99	C+
60-62.99	D-	63-66.99	D	67-69.99	D+
0-59.99	F				

Class Policies:

- Attendance: **There will be no make-up for missed work**. A grade of 0 is awarded for missed work in the absence of a valid excuse. It is your responsibility to obtain notes and assignments from a willing classmate if you **MUST** miss class.

- Collaborative Work: *High standards of academic integrity are crucial for the University to fulfill its educational mission. To uphold these standards, procedures have been established to address academic misconduct.*[from LSU Code of Student Conduct]. It is assumed that all students enrolled in this course have read the Code of Student Conduct - specifically section 10.1 (Academic Misconduct). The Code of conduct is available at https://www.lsu.edu/saa/students/codeofconduct.php.

- Due Dates: Students are expected to turn in all assignments on time. Non-programming assignments must be turned in at the beginning of class on the due date. The cut-off period for turning in these assignments is the end of class with a penalty of 10% if they are turned in 15 minutes after class has begun. All programming assignments will be submitted via digital drop boxes on Moodle. The cut-off period for late submission of programming exercises is eight hours with a late penalty of 10%.

- Grading Corrections: All grades are uploaded to the course Moodle. Concerns about grades must be addressed within a week after the grade is posted to the grade book. Thereafter, all grade book entries are final.

- Missed Exam: Students are encouraged to take every exam. In the unusual circumstance you miss an exam due to medical reasons or other unforeseen emergencies, obtain an official excuse from the Dean's office as soon as possible. If you obtain a valid excuse from the Dean's office, the instructor reserves the right to schedule a cumulative final exam that will count for both your midterm and final exams or schedule a make-up exam if it is the mid-term exam that is missed.

- Special Accommodation: Students who have a disability that require accommodation(s) should make an appointment with the Office of Disability Services (Phone: (225)578-5919 or TDD: (225)578-2600) to discuss their specific needs and present a letter from the ODS informing the instructor of their needs. All such matters, by University regulations, are strictly confidential.

- Exam Dates:

  - ⊙ Midterm Exam -Thursday, February 27
  - ⊙ Final Exam - Thursday, May 7, 3:00 - 5:00 p.m.

- Important Dates:

  - ⊙ Monday, January 27 - Final day to drop without a W (4:30 p.m. deadline)
  - ⊙ Monday, April 6 - Final day for dropping courses (4:30 p.m. deadline)

- Topics we will study:

  1. Asymptotic Time Complexity Analysis
  2. Sorting and Searching Algorithms
  3. Heaps and Priority Queues
  4. Trees and Binary Search Trees
  5. Huffman Trees and Encoding
  6. String Matching
  7. Dynamic Programming
  8. Hash Tables, Maps and Sets
  9. Graphs and Graph Algorithms
  10. B-Trees

Problem Sets

Topic	Reading	Exercises
Analyzing Algorithms	§1.1-1.2	R-1.3-1.25, C-1.5-1.6[1]
Merge Sort and Quick Sort	§8.1-8.2	R-8.3, 8.6
Heaps and Priority Queues	§5.1-5.4	R-5.3-5.6, 5.10, 5.12-5.14
Trees and Binary Search Trees	§2.3, 3.1	R-2.6, 3.1-3.7
AVL Trees	§4.4[2]	EX 4.19[3]
	§4.2	R-4.1, 4.6, 4.10-4.12
Dynamic Programming	§10.1, 12.2, 12.6	C-12.1[4], R-10.1, R-12.5
Graphs	§13.1, 13.4	R-13.1-13.2, 13.5, 13.7
Shortest Paths	§13.4, 14.1-14.3	R-14.1-14.2
BFS and DFS Traversals	§13.2	R-13.6, 13.12-13.14
Topological Ordering	§13.4	R-13.4, 13.10d
Minimum Spanning Trees	§15.1-15.2	R-15.1[5]
Huffman Coding	§10.3	R-10.4, 10.6
Sets and Maps	§6.1	
Hash Tables	§6.2-6.3, 6.5	R-6.4-6.7
B-trees	§20.2, 4.7[6]	R-20.4, EX 4.9[7]

[1] Solve the exercises by Algebraic Unrolling
[2] Source: Data Structures and Algorithms Analysis···, Mark Allen Weiss
[3] Source: Data Structures ···, Mark Allen Weiss
[4] Draw box trace diagrams when computing c(5,3) with and without memoization
[5] Repeat this exercise using Prim's Algorithm
[6] Source: Data Structures and Algorithms Analysis···, Mark Allen Weiss
[7] Source: Data Structures ···, Mark Allen Weiss - using an order 3 B-tree