CSCE 2100.001 Computing Foundations I

Instructor Information
Name: Ian Parberry
Office: NTDP F209
Office hours: MW4:00pm - 5:00pm
URL: http://larc.unt.edu/ian

Teaching Assistants
Venkata Yanambaka
Yang Zhou

Peer Mentors
Chris Wolfe
Brian Reeder

Course Description
This course introduces students to both data structures and formalisms used in computer science, such as asymptotic behavior of algorithms. The text does an excellent job of weaving these topics together so that students learn about data structures and the formalisms used to both describe and evaluate those data structures simultaneously. By the end of the two-semester sequence of which this course is the first part, each student will have a solid foundation in conceptual and formal models, efficiency, and levels of abstraction as used in the field of computer science.

Prerequisites
CSCE 1040

Textbook
We will be using an online text this term.
Authors: Alfred Aho and Jeffrey Ullman
Title: Foundations of Computer Science
URL: http://infolab.stanford.edu/~ullman/focs.html

Recitations and Quizzes
There are five recitation sessions. Recitations will begin on Monday, February 1.
201 M 8:30-9:20am, NTDP B158
205 M 12:30-1:20pm, NTDP B157
202 W 09:30-10:20am, NTDP D215
203 W 10:30-11:20am, NTDP D215
204 F 11:30am-12:20pm, NTDP B157
Please attend only the recitation section for which you are signed up. Attendance is mandatory. Quizzes will be given during recitations. There will also be time for staff to answer questions about course material, quizzes, programming assignments, and examinations.
Examinations
The Midterm Examination will be held in class on Wednesday, March 9. Monday, March 7 will be a review session. The Final Examination will be held in the classroom at 1:30-3:30pm on Monday, May 9. The Final Examination will not be comprehensive, although knowledge of all class material may be necessary. Wednesday, May 4 will be a review session.

Learning Outcomes
Students will demonstrate:
1. A solid foundation in conceptual and formal models.
2. The ability to use abstraction in the design and description of algorithms.
3. Use of C++ classes to implement trees, and lists.
4. Application of big-Oh notation to evaluating and comparing algorithms.
5. Use of tree, and list data structures in design of software.
6. An ability to apply combinatorics in solving real-world problems.

Topics
- Iteration, induction, and recursion
- The running time of programs
- Trees
- Lists
- Combinatorics

Grading
- Programs (25%)
- Quizzes (given in Recitations) (25%)
- Midterm Exam (25%)
- Final Exam (25%)

Collaboration and Cheating
You are to do your own work. That means that you are forbidden to copy code from any source, including but not limited to other students in this class and the internet. Any student who violates this requirement will be given an F for the class. Just as there are many sources of code on the internet, there are many automated systems that detect any feeble attempt to hide your cheating by, for example, changing the variable names or restructuring the code.

ADA
UNT complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services please contact the Office of Disability Accommodation