CS 4390 Special Topic in Computer Science: Bio-related Data Structures and Algorithms

Fall 2013

1. General Information

Instructor:
Olac Fuentes
Email: ofuentes@utep.edu
Web: www.cs.utep.edu/ofuentes
Office hours: Mondays and Wednesdays 2:00-3:00, or by appointment, in CCSB 3.0412 (feel free to drop by at other times if my door is open). Chat: olacfuentes@gmail.com

Lectures
TTh 1:30-2:50, UGLC 216

Class web site:
www.cs.utep.edu/ofuentes/cs4390.html

Textbook

2. Objectives and Outcomes

This is the third and final course in the fundamental computer science sequence. Students will learn about fundamental data structures and analysis and design of algorithms.

Level 3: Synthesis and Evaluation:
Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to

1. Design and implement solutions to computational problems using the following data structures:
   a. multi-dimensional arrays
   b. strings
   c. lists implemented as arrays or linked lists

Level 2: Application and Analysis:
Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:
1. Describe, implement, and use the following data structures:
   a. Stacks
   b. Queues
   c. Binary trees and binary search trees
   d. Balanced trees
   e. Graphs
2. Describe, implement, and use the following algorithms:
   a. Sequential and binary search
   b. Quadratic and $O(n \log n)$ sorting
   c. String manipulation and parsing
3. Describe, implement, and apply the following graph algorithms:
   a. Connected components
   b. Breadth-first search
   c. Depth-first search
   d. Topological sorting
   e. Minimum spanning trees (Kruskal's and Prim's)
   f. Single-source shortest paths
4. Use recursion and iteration as problem solving techniques

**Level 1: Knowledge and Comprehension**
Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able to:

   1. Understand use the following concept
      a. Classes, subclasses, and inheritance
      b. Encapsulation and information hiding
   2. Understand basic notions of algorithm complexity and the need to assess them when designing algorithms:
      a. Big-O notation to express the best-, average- and worst-case behaviors of an algorithm
      b. Best, average and worst-case behaviors of a simple algorithm
      c. Time and space trade-offs in algorithms
   3. Identify and explain the following algorithm design techniques:
      a. Greedy algorithms
      b. Divide and conquer
      c. Dynamic programming
      d. Backtracking
      e. Randomization

**3. Policies and Other Information**

**Prerequisite:** Minimum "C" grade in CS1401 or equivalent.

**Textbook:** Reading and laboratory assignments will be drawn from Data Structures and Algorithm Analysis in Java by Mark Allen Weiss. You are required to obtain this book for use in this course. Note that photocopied textbooks are a violation of copyright law. Any student caught with a photocopied book will be referred to the Dean of Students for discipline.

**Grading:** Final grades will be based on a combination of lab projects, homework assignments, in-class attendance and performance, three partial exams, and a final exam. The approximate weights are as follows:
- 25% - Lab projects
- 10% - Homework assignments, in-class exercises, and quizzes
- 39% - Partial Exams (3 exams, 13% each)
- 26% - Final Comprehensive Exam

The nominal percentage-score-to-letter-grade conversion is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- below 60% is an F

Additionally, any one of the following will result on a final grade of F, even if the overall average is greater than 60%.

- Obtaining an average of less than 60% on the lab projects
- Obtaining a grade of less than 50% on the final exam
- Obtaining an average of less than 50% on the partial exams
- Not submitting ALL lab projects by the end of the semester, even if they are too late to receive credit

We reserve the right to adjust these criteria downward, e.g., so that 88% or higher results in an “A”, based on overall class performance. The criteria will not be adjusted upward, however. You must earn a “C” or better to be able to register for upper division computer science courses.

**Late homework submission**: Homework up to a day late will receive up to 80% of full credit, and it will not be accepted after that.

**Collaboration**: Collaboration among students is strongly encouraged.

It is OK to:
- Talk with other students about approaches and ideas.
- Get ideas and extra information from the internet, books, etc.

However, it is not OK to:
- Share code with another student (if a piece of code is submitted by two or more students, both students are guilty of cheating, regardless of who wrote the original code).
- Use code acquired from an outside source (the internet, a friend, etc.)
- Look at another student’s code
- Debug another student’s code

We will use software to detect plagiarized programs and take appropriate disciplinary actions if necessary.

**Cellular telephones are prohibited** during lecture and lab sessions. Students are required to turn off their cellular telephones before entering the classroom or laboratory session.

**Laptops and tablets can only be used occasionally with the instructor’s permission.**

**Disabilities**: If you feel that you may have a disability that requires accommodation, contact the Disabled Student Services Office at 747-5184, go to Room 106E Union, or email dss@utep.edu

4. **Lab Submission Guidelines**
Lab assignments will be posted on-line. Each lab grade will be computed from the following three elements:

- Report (40% of grade)
- Source code (60% of grade)
- Demo session (pass/fail)

**Report:**
You must submit a printed report of every lab that includes the following items:

- **Introduction** – Description of the problem you are trying to solve
- **Proposed solution design and implementation** – How did you solve (or attempt to solve) the problem? Provide an informal, high-level description. Description of your code (not the actual code). Explain the design choices you made, including how you broke the program into modules, your user interface, input and output, etc.
- **Experimental results** – Describe the experiments you performed to test your program and show the output your program produced. The experiments must be described in a way that allows anybody to replicate them using your code. Include sample runs that illustrate the outputs and running times of your program under different types of inputs. **If results are not included in the report, we will assume that your program does not work.**
- **Conclusions** – Explain what you learned from the project.
- **Appendix** – Source code
- **A signed academic honesty certification** stating the following: “I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.”

Reports will be graded as follows:

- **Completeness** (12%)
  Does your report cover all required aspects in enough detail?
- **Clarity** (10%)
  Are those aspects clearly explained?
- **Language** (10%)
  Is the report written with proper grammar and spelling?
- **Presentation** (8%)
  Is the formatting appropriate?

**Source Code:**
Working programs must be submitted by email to both your instructor, using the e-mail address listed below. Labs not submitted this way will not be eligible for credit.

- Olac Fuentes: olacfuentes@gmail.com

Include at the beginning of the subject line the following characters: “[CS4390-Fall 2013]”

Source code will be graded using the following guidelines:

- **Correctness** (36%)
  Does the program compile?
  Does the program run correctly?
- **Design** (6%)
  Are operations broken down into methods in a reasonable way?
- **Style** (6%)
Is the program indented correctly and consistently?
Do methods and variables have meaningful names?
- Robustness (6%)
  Does the program handle erroneous or unexpected input gracefully?
- Documentation (6%)
  Do all program files begin with a comment that identifies the course, author, assignment, instructor, T.A., date of last modification, and purpose of program?
  Are all methods clearly documented?
  Are all non-obvious code segments clearly explained?

Policy on late projects:
Lab project grades will be reduced by a factor of 10% for each working day or fraction they are late.

Missing lab assignments:
All labs must be submitted by the end of the semester in order to pass the class. Additionally, a student who has submitted less than 75% of the labs due by the time a midterm exam is given won’t be allowed to take that exam.

5. Standards of Conduct and Academic Dishonesty

You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct: http://admin.utep.edu/Default.aspx?tabid=30292

Academic dishonesty includes but is not limited to cheating, plagiarism and collusion. Cheating may involve copying from or providing information to another student, possessing unauthorized materials during a test, or falsifying data (for example program outputs) in laboratory reports. Plagiarism occurs when someone represents the work or ideas of another person as his/her own. Collusion involves collaborating with another person to commit an academically dishonest act.

Professors are required to - and will - report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.