CSCE 310J
DATA STRUCTURES AND ALGORITHMS

Fall 2004

TuTh 2:30 – 4:30pm
Kauffman 112

Instructor: Dr. Steve Goddard
Kauffman 133
472-9453 (TTh) 472-9968 (MWF)
goddard@cse.unl.edu
http://cse.unl.edu/~goddard

Office Hours: 12:00-2:30pm TTh
(Or by appt.)

TA: Mathew Johnson
Kauffman 133
472-9453
msjohnso@unlnotes.unl.edu

Office Hours: 4:30-5:30pm M-Th
(Or by appt.)

Course Overview

An algorithm is a set of instructions that, when followed, solve a specific problem. Programs are implementations of algorithms that are executed by computers. The computer science goals of this course are to study classic data structures and algorithms that solve common problems and to learn standard approaches to solving new problems. A rigorous approach to the analysis and comparison of algorithms will be followed that includes asymptotic notation and proofs of correctness. Discrete mathematics, which forms the foundation for rigorous analysis, will be covered as needed. The course will involve substantial programming and written assignments.

A broader set of objectives for this course is to teach critical thinking, how to learn, and how to communicate technical concepts. These objectives will be met through lectures, challenging team and individual assignments, regular quizzes, and a final examination.

The following topics will be covered:

- The basics of algorithm analysis including
    - Asymptotic notation
    - Complexity classes
    - Discrete mathematics
    - Recursion and induction (proofs).
    - Creating and solving recurrence relations
- Algorithmic techniques, including
    - Brute Force
    - Divide-and-Conquer
    - Decrease-and-Conquer
    - Transform-and-Conquer
    - Space and Time Tradeoffs
    - Dynamic Programming
    - Greedy Techniques

  - Distributed Algorithms
- Theory of computing, including
  - Finite state machines
  - The halting problem
  - Tractable and intractable problems
  - Complexity classes like P, NP, and NP-Complete

Course Materials

The textbook for this course is *Introduction to the Design and Analysis of Algorithms*, Anany Levitin, Addison Wesley, 2003. We will probably cover Chapters 1-7, except Sections 2.5, 2.7, 4.5, 5.4, 6.2, 6.5, and 7.2; Section 8.4 of Chapter 8; Sections 9.1-9.3 of Chapter 9; Section 10.3 of Chapter 10; Sections 11.1-11.3 of Chapter 11; Appendix A and B. The actual sections covered and order may vary slightly. I will provide supplemental information for Distributed Algorithms, Finite State Machines, and the Halting Problem.

You will be doing substantial programming in the C++ language and working with the Linux/Unix operating system. The book, *Just Enough UNIX*, Paul Anderson, McGraw-Hill, 2003, will be helpful in learning the operating system and programming environment.

In addition to the texts, I will make available a set of lecture notes for each class. These notes are purposely incomplete and are meant to simply be a note-taking aid. The lecture notes, as well as copies of most class handouts, can be found on the CSCE 310J home page on the Web at URL:

`http://www.cse.unl.edu/~goddard/Courses/CSCE310J/`

Prerequisites

Prerequisites for this course are CSCE 156 (Introduction to Computer Science II). Prerequisites by specific topics are:

1. **Mastery** of data structures including list, stacks, and queues.
2. **Familiarity** with recursion.
3. **Exposure** to complexity analysis.

As part of our ongoing accreditation process, there will be an evaluation of your understanding of the material covered in the prerequisite courses. This evaluation will be graded as a homework assignment.

Grading

Your grade in this class will based on:[1]

- Assignments 30%,
- Classroom participation, peer, and instructor evaluation 5%
- Programming project 15%,
- Bi-weekly (15-20 minute) Quizzes 25%, and
- Final examination 25%.

[1] The percentages listed are only approximate and are subject to change (by no more than 5%).

A minimum grade of C is required for the course to count toward a CS/CE major or minor. Note: a grade of C- does NOT count toward a CS/CE major or minor. No incompletes (I) will be given. Letter grades will be assigned at the end of the semester, using the percentage of possible points, as follows:

A+: 101+	A: 93-100	A-: 90-92	
B+: 87-89	B: 83-86	B-: 80-82	
C+: 77-79	C: 73-76	C-: 70-72	
D+: 67-69	D: 63-66	D-: 60-62	F: 0-59

Assignments: Homework will be assigned approximately on a bi-weekly basis. Many assignments will be team assignments and will include programming exercises and analytical (pen and paper) problems. Teams will be randomly selected for team assignment. All assignments will be due at 9pm on the day on which they are due. Assignments will be submitted via the Web-handin program.

All homework submitted after its deadline is considered late. Assignments that are submitted within 24 hours after the original deadline are considered to be “one day late,” within 48 hours, “two days late,” *etc*. A late homework assignment, but not the final project, will be accepted without penalty if the following conditions are met:

- the total “lateness” of all homework assignments received to date (including the current assignment) does not exceed 3 days.
- the student does not miss class on the day the assignment is due or on the day after the assignment is due. Exceptions to this requirement must be approved by the instructor in advance.

Late programs should be handed in with the Web-handin program. The penalty for late assignments is 25% per day they are late. An assignment that is 4 days late will receive no credit. **Weekends count in evaluating the lateness of an assignment.**

Program correctness is assumed and only accounts for a portion of the assignment grade. Each assignment will also require a document that describes the design; design tradeoffs considered; an analysis of the program’s computational complexity (when requested); testing performed; and a detailed description of outstanding program defects. All programs will follow the JDEHP coding standard. The coding standard specifically addresses programs written in Java, C++ and C, but it can easily be extended to cover other programming languages.

Peer and Instructor Evaluation: Each student will be required to fill out and submit an evaluation form that describes the performance of the team members and the team as a unit for each assignment, including the course project. These evaluations will be combined with the instructor’s personal evaluation of each student’s performance in the team and in the classroom.

Course Project: The course project will integrate all concepts covered during the semester, but it is more open-ended than regular assignments. A written proposal and a final report will be required. The final report and program source code will be due 9:00pm Saturday, December 11, 2004. There will be NO late days allowed for the course project. A penalty of 25% per day will be assessed for late projects. Early submissions are encouraged. Consider this your notice of an assignment due during Dead Week. More details regarding the project will be distributed after Fall Break.

Quizzes: There will be NO midterm examination. Quizzes, given approximately every two weeks, will take the place of a midterm exam.

Final Examination: The final exam for this course is scheduled for Monday, December 13 at 3:30 pm. The final exam covers the entire course.

Expected Workload

- Regular Lectures: 4 hours per week
- Support Lectures: Additional lectures (1 hour each) may be provided during the semester
- Study / Project: average of 8 hours per week
- No classes on Tuesday October 19th and Thursday November 25th

Course Conduct and Academic Integrity

Many of the assignments will be team assignments that will require collaboration within your team. Individual assignments consist of a team of one (for the purposes of this section). *Acceptable collaboration* with students not on your team (or for an individual assignment) includes:

- discussing the assigned problems to understand their meaning,
- discussing possible approaches to assigned problems,
- discussing general programming principles in the solution of programming problems or UNIX system features that will help solve the problem.

In all cases you must *explicitly* acknowledge any and all substantive help received from other teams during the course of the preparation of your solution. That is, if you collaborate with other teams then you must include an explicit acknowledgment in your solution of the persons from whom you received aid. Individual assignments are defined as a team of one person.

Unacceptable collaboration, unless explicitly stated, on homework includes:

- copying (verbatim use) of physical papers or computer files.[2]
- submission of solutions that are jointly authored, or authored either wholly or in part by other individuals not on your team.

The general rule to be followed is that the strategy and approach of solutions may be developed jointly but *all* actual solutions (*i.e.*, the final solution) must be constructed and written up by individual teams. Work done with other teams should not be done in sufficient detail as to make it a solution. For example, the design of a program solution may be performed jointly, however, each team must write *all* the code they eventually submit as their solution. *No code may be shared between teams*. Similarly, for written assignments, solutions may *sketched* out with other teams, however each team must construct the final form of their solution individually and write-up their own solution. You will be held accountable if someone else copies your work, even if you are unaware of the event. Thus, you should make sure all of your files are properly secured.

Unacceptable collaboration will be considered a violation of the Student Code of Conduct, and will result in a failing grade for the course. In other words: if you cheat, you will fail! In addition, the incident will be reported to the CSE Department and to the JDEHP Director, in

[2] This includes computer files that are copied and then edited and/or reformatted.

accordance with the CSCE department policy on academic integrity. You are responsible to read the CSCE department policy and adhere to it.

Should questions arise during the course of working on a problem, please feel free to immediately contact the instructor either by telephone, electronic mail, or by an office visit. In principle, if you work with others in good faith and are honest and generous with your attributions of credit you will have no problems.

Special Needs

Any student in this course who has a disability that necessitates accommodation should contact the instructor as soon as possible to discuss the appropriate accommodations necessary to complete the course requirements.