RAIK 284H
FOUNDATIONS OF COMPUTER SYSTEMS

Spring 2010

TR 9:00 – 10:45am
Kauffman 112

Instructor:	Dr. Steve Goddard Avery 256 472-2401 *goddard@cse.unl.edu* *http://cse.unl.edu/~goddard*	Office Hours:	11:00am-12:30pm TR @ Avery 256 Or by appt.
TA:	Dylan Douglas Kauffman Lounge *dondyl88@yahoo.com*	Office Hours:	12:30-1:30pm MWF Or by appt.

Course Overview

The computer science goals of this course are to introduce the fundamental organization and structure of computer systems. Topics include Boolean logic, data representation, processor organization, input/output, memory organization, system support software and communication.

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 assignments, regular quizzes, and a final examination.

The following topics will be covered:

- Computer systems
- Data representation
- Machine language
- Processor architecture
- Memory hierarchy
- Linking
- Exception control flow
- Virtual memory
- System level I/O (as time allows)
- Network programming (as time allows)
- Concurrent programming (as time allows)

Course Materials

The textbook for this course is *Computer Systems: A Programmer's Perspective,* Randal Bryant and David O'Hallaron, Prentice Hall, 2002

You will be doing substantial programming in the C language and working with the Linux/Unix operating system. The book *The C Programming Language,* 2nd Ed. (ANSI C version) or later, Kernighan & Ritchie, Prentice Hall, 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 JDEP 284 home page on the Web at URL:

`http://www.cse.unl.edu/~goddard/Courses/JDEP284/`

Prerequisites

Prerequisite for this course is CSCE 156. 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%,
- Peer and instructor evaluation 5%
- Programming project 15%,
- Bi-weekly (30 minute) Quizzes 25%, and
- Final examination 25%.

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. Assignments will be a mix of individual and team assignments and will include programming exercises and analytical (pen and paper) problems. Teams will be randomly selected for team assignments. In some cases, you will be able to organize your own teams. All assignments will be due at 9pm on the day on which they are due. Assignments will be submitted via the Web-handin program or another Internet-based program.

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

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 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 4 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 assignments 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 programming grade. Some programming assignments will also require a document that describes the design; design tradeoffs considered; testing performed; and a detailed description of outstanding program defects.

Course Project: The course project will integrate most concepts covered during the semester, but it is more open-ended than regular assignments. An oral presentation and demonstration will be required, which must be completed before 9:00pm Wednesday, April 28, 2010 and within available presentation slots made available by the TA. Final source code will be due at the same time. 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 Spring Break.

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

Post Assessment Test (PAT): A "post assessment test" will be given for this course at the end of the semester to assess our undergraduate CS and CE programs. This assessment test will count as part of the 5% peer and instructor evaluation portion of your grade. The test will be delivered through Blackboard/Maple TA and will consist of about 30 multiple-choice questions covering topics in the course randomly pulled from a large test bank. You will be required to take this assessment test exactly once, at the Arts & Sciences Testing Center located in 127 Burnett Hall, during the one-week period that starts Wednesday of the fourteenth week of class (April 21, 2010) and ends on Wednesday of the fifteenth week (April 28, 2010). The test length will be approximately 30-45 minutes, depending on number of questions asked. More details and specific test availability times will be announced later in the semester.

Final Examination: The final exam for this course is scheduled for Monday, May 3 at 7:30 am! (Be sure you do not oversleep and miss this early morning examination time!) The final exam covers the entire course.

Expected Workload

- Regular Lectures: 4 hours per week
- Support Lectures: Labs and additional lectures (2 hours each) will be provided during the semester on selected Fridays.
- Study / Project: average of 8 hours per week

The tentative schedule of topics, readings, and assignments follows. The dates and number of assignments is likely to change. Please note, the last four weeks of the schedule are the most likely to change, based on input from the class.

Week	Topic	Readings	Support Lectures	Milestones
1	Computer systems	Ch 1		Pre-test: HW0, Q1
1-3	Data representation & Arithmetic	Ch 2		Q2, HW1
3-6	Machine language	Ch 3		Q3
6-9	Processor architecture	Ch 4		Q4, HW2
10	Spring Break	N/A		N/A
11	Memory hierarchy	Ch 6.1-6.3		Q5, HW3
12	Linking	Ch 7		Q6, HW4
12-13	Exception control flow	Ch 8 (8.1-3)		Q7
14	Virtual memory	Ch 10		HW5
14-15	System level I/O	Ch 11		Q8
15-16	Network and concurrent programming	Ch 12,13 (lightly)		
16	Review for Final Exam			Projects Due

Course Conduct and Academic Integrity

Both individual and team assignments will be given. The following text on collaboration is written in terms of teams. Individual assignments constitute teams of size one. *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. As stated above, 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 RAIK Director, in accordance with the CSE department policy on academic integrity. You are responsible to read the CSE 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.

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