

## CSCE 310J Data Structures & Algorithms

Dr. Steve Goddard  
[goddard@cse.unl.edu](mailto:goddard@cse.unl.edu)  
Kauffman 133, Avery 357,  
and Suite 304, 121 S. 13<sup>th</sup> St. (Miller & Paine)  
472-9453 and 472-9968

August 24, 2004

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

## CSCE 310J Data Structures & Algorithms

- ◆ Data Structures & Algorithms
  - » 2:30-4:30 TuTh
  - » Kauffman 112
- ◆ Instructor: Dr. Steve Goddard
  - » Office hours: 12:30-2:30pm TuTh
  - » Office: Kauffman 133
  - » e-mail: [goddard@cse.unl.edu](mailto:goddard@cse.unl.edu)
- ◆ TA: Matt Johnson
  - » email: [msjohnso@unlnotes.unl.edu](mailto:msjohnso@unlnotes.unl.edu)

## Textbook

- ◆ *Introduction to the Design and Analysis of Algorithms*, by Anany Levitin. Addison Wesley.
- ◆ *Just Enough UNIX, 4<sup>th</sup> Ed*, Paul K. Andersen, McGraw-Hill

## 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
  1. Study classic data structures and algorithms that solve common problems.
  2. 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.

## Broader Objectives

- ◆ Learn critical thinking
- ◆ Learn how to learn (on your own)
- ◆ Learn how to communicate technical concepts
- ◆ These objectives will be met through
  - » interactive classroom discussions,
  - » challenging team assignments,
  - » written reports,
  - » regular quizzes, and
  - » a final examination.

## Topics Covered

- ◆ The basics of algorithm analysis
- ◆ Algorithmic techniques
  - » Brute Force
  - » Divide-and-Conquer (Decrease-and-Conquer, and Transform-and-Conquer)
  - » Space and Time Tradeoffs
  - » Dynamic Programming
  - » Greedy Techniques
  - » Distributed Algorithms
- ◆ An overview of the theory of computing

## Prerequisites: 155 and 156

- ◆ **Mastery** of data structures including list, stacks, and queues.
- ◆ **Familiarity** with recursion.
- ◆ **Exposure** to complexity analysis.

7

## Grading

- ◆ Assignments 30%,
- ◆ Class participation, peer and instructor evaluation 5%,
- ◆ Programming project 15%,
- ◆ Bi-weekly (15-20 minute) Quizzes 25%, and
- ◆ Final examination 25%.
- ◆ Note: There will be no midterm!

8

## Grading

- ◆ A minimum grade of C is required for the course to count toward a CS/CE major or minor.
- ◆ 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 |

9

## Assignments

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

10

## Programming

- ◆ Computing platform: Linux
- ◆ Computing language: C++
- ◆ All programming must follow the JDE Coding Standard (see the course Web page).
- ◆ Program correctness is assumed
  - » Less than 50% of the grade
  - » Program design, style, analysis, documentation, and conformance to specifications account for a majority of the grade!

11

## Quizzes

- ◆ Quizzes, given approximately every two weeks, will take the place of a midterm exam.
- ◆ 15-20 minutes in length.
- ◆ Format:
  - » Brief explanation of concepts based on the reading assignment for pending lecture
  - » At least one open question on material already discussed in class.

12

## Late Homework

- ◆ Late homework is “OK” but...
  - » Only if it’s not too late
  - » You don’t miss class to get it done
  - » You’re not late too often

13

## Late Homework Details

- ◆ 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 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.
- ◆ The penalty for late assignments is 25% per day they are late.
- ◆ **Weekends count in evaluating the lateness of an assignment.**

14

## How to get an “A” in CSCE 310J

- ◆ Attend class regularly
  - » Ask questions!
- ◆ Read the book
- ◆ Do the homework
- ◆ Study!

15

## How to get a “D” in CSCE 310J

- ◆ Do not read the text and lecture notes in advance
- ◆ Assume getting copies of handouts is sufficient
- ◆ Don’t take notes in class
- ◆ Miss class
- ◆ Waste time playing on the Web

16

## Course Conduct

- ◆ You may work in groups in *understanding* assignments,
- ◆ developing *approaches* and *strategies*
- ◆ *learning* to use the UNIX/Linux tools
- ◆ You may not
  - » develop joint solutions with other teams
  - » share code between teams
  - » copy anything
- ◆ All assignment solutions must be authored in full by your team!
- ◆ Individual assignments constitute a team of size one!

17

## Summary

- ◆ We will
  - » study classic data structures and algorithms that solve common problems, and
  - » learn standard approaches to solving new problems.
- ◆ Assignments will involve team work, programming, analysis, critical thinking, and writing.
- ◆ Assignments will be based on business problems and integrated with your business class.
- ◆ This course will be a lot of work.
- ◆ Hopefully, it will also be fun!

18