CS271: Data Structures

Course Syllabus

Professor	Matt Kretchmar
Office	207 Olin
Office Hours	Mon & Tues 2:30-3:30, Wed & Fri 2:30-4:30
Phone	740-587-6721
Email	kretchmar@denison.edu
Webpage	`www.denison.edu/~kretchmar/cs271`
Textbook	*Introduction to Algorithms, 3rd Ed.* by Cormen, Leiserson, Rivest and Stein

General Info:
CS271 is a gateway course in the computer science major. It is the bridge from the developmental programming and analysis techniques in earlier classes to the more robust software design, implementation and analysis of later classes and electives. Students will acquire a toolbox of the fundamentally important data structures and analytic techniques that form the foundation of the discipline. The volume of programming will be larger so as to provide an opportunity to not just learn these important tools, but to truly master them in a way that makes their use in subsequent classes second nature. This is also a class that features problem solving in a more predominent way than previous classes. Projects will be of a larger scale and will require intuition and imagination in their solutions. Students should also view this course as an opportunity to further refine professional skills, a solid work ethic, and time management strategies that will allow them to be successful in this course and beyond; there is a very strong correlation between mastery of these professional skills at this level and the student's ability to graduate on solid ground for their post-graduate careers.

Grades:
This course will utilize an alternative, competency-focussed grading scheme. The central idea is to move the emphasis from performance on exams and projects to demonstrated proficiency in the important course skills. CS271 features four important primary competencies for the entire class, and a selection of unit-based course competencies:

CS271 Core Competencies:
There are six overall course competencies. Some of these competencies focus on specific data structures or programming techniques, while others relate to analytical thought and professional management of the course.

1. **Stacks and Queues** The proper use and understanding of the stack and queue ADTs.

2. **Heaps and Hashing** The proper use and understanding of the heap ADTs and its use as a priority queue. The full understanding of a hash function and its use in an algorithmic setting. The understanding and analysis of how hash functions impact hash performance.

3. **Graphs and Trees** Understanding the fundamental ways to search paths through graphs using DFS, BFS and Dijkstra's algorithms. The use and implementation of binary search trees. The use and implementation of red-black trees and other balanced tree structures.

4. **Programming Schemes** The demonstration of the core concept of dynamic programming and its implementation in at least one significant problem. Demonstrate the use and understanding of functional programming and the LISP language in particular.

5. **Theory and Analysis**. Students must develop proficiency in proving the correctness of various algorithms as well as analyzing their resource consumption.

6. **Professionalism**. This competency encompasses the development of professional work habits including managing time effectively, working effectively in a small team, taking pride in producing quality work, contributing to class discussions and activities, being prepared to do so by staying on top of the assigned readings, and otherwise by being a good department citizen. Students are asked to develop effective communication skills including commenting on code, email communication with other students and the professor, and care in submitting the various written assignments. Students are expected to learn and use the LaTeX typesetting system.

Student achievement in each competency will be assessed on a three part scale:

- **Proficiency**. In the standard of proficiency, students must demonstrate they fully understand the underlying concept and are able to apply that concept in the correct situations. Proficiency is the expected minimum standard for each competency.
- **Mastery**. Mastery achieves a level of polish and expertise above proficiency. The standard of mastery is noted by work that is of higher quality, the infrequent appearance of errors or other mistakes, and a level of deeper understanding and insight.
- **Mastery with Distinction**. The standard of Mastery with Distinction is truly unique and special. It indicates an especially elegant solution, a substantially insightful level of understanding, or a noteworthy independent initiative to explore an area in more depth than what is expected.

The course will still feature a final letter grade using the standard scale. Students who are successful in completing the course (earning a high grade) should seek balance across the broad assortment of competencies rather than strength in a few at the expense of others. There is no formulaic translation from the competency scale to letter grades. Students who fail to demonstrate competency in one or more areas should expect to earn lower than a C grade for the course. Students who achieve competency in all areas should expect to earn between a C and B-. Students who achieve some degree of mastery, should expect to earn grades in the B range. And A for the course signals "extraordinary accomplishment" and represents multiple masteries and/or Mastery with Distinction.

Portfolios

In general, the above unit goals correspond to the outline of the major topics covered in this course. Topics will typically feature a traditional mix of traditional assessment tools: written assignments, programming projects (group and individual), quizzes, midterm exams, and a final. Students will receive back all submitted work with comments, suggestions, and reflections, but no numeric scores or letter grades. Students are required to complete each of these "assessment tools" but not to be evaluated on them. Instead, each tool is an opportunity for the students to grow in their comprehension of the material and to demonstrate their ability to understand the critical concepts. In this grading system, each assessment tool has a course-long lifespan; students may always go back to work more on an exam question, re-engage a challenging homework problem, improve an algorithm in a programming project, or to revise another draft of a paper.

Each of the course competencies will ultimately be assessed through a student portfolio. Each student must submit a portfolio for each competency. The purpose of the portfolio is to allow the student make a case for (demonstrate) their level of understanding in that particular competency. The portfolio is comprised of any combination of written work, programming work, exams, etc. that relate to that competency. Students, in consultation with the professor, may also include other items in the portfolio that may not have been directly assigned (be creative – oral exams, research papers based on assignments, corrected exams, interesting applications, etc.). Each portfolio should be accompanied with a carefully constructed reflection (about two pages) that provides an overview of the portfolio and why that portfolio demonstrates the student's desired level of competency.

Portfolios should be submitted in a timely fashion soon after we finish the material for that portfolio. The course webpage will provide a more detailed timeline, but generally we will finish (1) Stacks and Queues first, then (2) Heaps and Hashing. These two portfolios should be submitted, evaluated and returned to form the midterm grade. (3) Graphs and Trees will for the next collection of course units followed finally by (4) Programming Schemes. Competencies for (5) Theory and Analysis, and (6) Professionalism are really course-long competencies and thus these last two portfolios will likely be submitted at the conclusion of the course. The final exam will come too late to be integrated in the portfolios; students are still required to take the final and the performance therein may allow some small adjustment in borderline course grades. It is the student's responsibility to stay on track with the intended schedule and to submit portfolios in a timely manner.

In addition to regular office hour visits (which are highly encouraged), students will schedule three out-of-class meetings with the professor. In the first meeting, to take place during the second week of class, students should bring with them a written plan of what they intend to achieve in the various competencies as well as an implementation plan of how they intend to achieve those objectives. We will meet individually again in the week before midterm grades to assess the execution of the student's plan and to assess their achievements in the class to date. We will schedule the final meeting during the last week of classes to assess the overall class performance.

Other Academic Policies:
Academic integrity is of the utmost importance. A good rule to follow is to make sure the work you submit reflects your own intellectual achievements and not those of someone else. Since this course will feature team projects, it is important to keep in mind when a particular endeavor is intended to be shared among a group and when it is intended to be an individual effort; in general, assume that work is to be completed individually rather than in consort with others unless explicitly stated. In

the event of team projects, it will be important that each student contribute actively to the team's effort. Cases of academic fraud are required to be reported to the institution (where they may affect your permanent record) and will incur a course grade penalty such as failure for the assignment or failure for the whole course. If you have any doubts or gray areas, please first ask the professor.

Any student who feels he or she may need an accommodation based on the impact of a disability should contact the instructor privately as soon as possible to discuss his or her specific needs. The instructor relies on the Academic Support & Enrichment Center in 104 Doane to verify the need for reasonable accommodations based on documentation on file in that office.

Traditional Grading vs Competency/Portfolio Method
Some advantages of the Competency/Portfolio evaluate method:

- Separate learning from evaluation. The traditional assessment tools (projects, exams, hw) are not assessed until/unless the student submits them in a portfolio. Thus there is a clean distinction in the course between assessment and learning opporunities.

- Failure is permitted. Students may fail exams, do poorly on a paper, have substantial problems with a project, etc. and not have it impact their grade. In this system, failure is a learning opportunity rather than points subtracted from your overall grade. Students decide what is submitted in their portfolio for evaluation.

- More control in evaluation. Students play a greater role in planning for, implementing and making a case for a particular achievement (grade). For example, a student may not be especially strong in taking exams but be a better programmer. This system allows students to play to their strengths to maximize their evaluation. It also allows students to be more explicit in their achievements through planning and execution.

- Stronger correlation of learning and assessment. Professors often ask students to do things which are subsequently not factored in to the evaluation. In this scenario, students are given credit for those extras that form the complete portfolio of a student's performance.

- This scheme more closely matches the professional evaluation of post-graduate careers. Both in my experiences as a professional programmer at IBM and as a professor at Denison, I have undergone multiple evaluations which are structured around this model. Professionals are largely responsible for com-

piling their own ”portfolios” and stating a case for their performance and value-add to their organization.

Most of the drawbacks of this evaluation method concern more time requirements from students; the management of the course and creation of the portfolios are extra steps typically not required in a traditional grading scheme. However, the value in completing these activities more than makes up for the incremental effort. Furthermore, the professor recognizes the additional work and has subsequently scaled back some of the material. Some challenges of the Competency/Portfolio evaluation method:

- Students are required to perform additional steps, especially with the compilation and submission of the portfolios.
- There is less immediate feedback in tracking a numeric grade as a percentage of the total that often accompanies a traditional grading system.
- This method places a greater time management responsibility on the student to stay on top of their workload. Thus it is critical to both have a plan in place and to find the time to execute that plan.