CS 4000: Senior Capstone Design

Syllabus: Administrative Details and Course Objectives

Fall 2018

Course Staff and Schedule

Professor:	H. James de St. Germain
Office:	3190 MEB, TBD
Email:	germain@cs.utah.edu
Office Hours:	After class or by appointment
Web Page:	utah.instructure.com
TA:	TBA
Lectures:	MWF 10:45-11:30, WEB 2230

Course Overview and Objectives

The purpose of the Capstone Project is for seniors to bring all of their knowledge and abilities to bear to work in a team environment to create an excellent, professional, and polished, software-product. This project should form the cornerstone of your portfolio when applying for a job. In order to not only achieve this goal, but to do so with excellence, a great deal of planning, team building, research, review, and formalization must take place. The purpose of the Senior Capstone Design course is to address this need.

The course focuses on all of the pre-development steps necessary for defining and developing the final project. These steps include: team formation and ideation, presentations and critique, final idea formation, planning and design work, prototyping, etc. Throughout we will focus on the "soft skills" (e.g., teamwork, public speaking, writing, etc.) necessary to create an effective team, design, and project.

The class will initially meet three times a week, but this will drop as we move on to latter parts of the course where team-faculty meetings will take place. There will be numerous homework assignments used to develop your ability to design, communicate, and strengthen your soft skills. There will be several presentations and opportunities for critique and improvement. There will also be a final prototyping phase.

You should plan to spend a commensurate amount of time on this class as your other CS electives.

Please note, that at the end of this process, we want you to have produced a software project that will resonate with and impress members of industry and academics. Something that industry would be willing to sell or use internally. Something that will impress your favorite professor. Something that demonstrates your talents and ingenuity. When we push you, we are pushing you to realize this goal.

Definition of Capstone

"Capstone" is defined as *the crowning achievement or final stroke; the culmination or acme.* This is the proper way to think of your CS 4000-4500 experience. Therefore, strongly consider taking all CS core and elective courses needed to complete your Senior Project ***prior to this coming spring***. During implementation of the Senior Project, it is expected that you are applying CS concepts and skills you ***have already learned***. For example, if your team is involved in a web-based application, at least one member should have already completed Web Software Architecture, Databases, and perhaps Human Computer Interaction.

It is required that students enrolled in CS 4000 enroll in CS 4500 during the following semester. Therefore, it is also required that all students currently enrolled in CS 4000 be ready to graduate in the spring.

Learning Objectives

At the end of the capstone experience, students will effectively be able to:

- Conduct oral presentations, describing projects in various states of completion
- Form new teams and work with team members to accomplish large scale goals
- Design, research, plan, and deliver new software projects
- Create well written, complex and significant software taking multiple months
- Create parts of a software system that must be integrated with code produced by other developers
- Create well written project documentation
- Identify and utilize the SE design tools and methodologies
- Provide reasonable constructive feedback to peers
- Function in an Agile process including working with Sprints, Design Goals, and Tasks.
- Apply past learning as well as develop new skills in the process of creating the software project

Web System

Please use the utah.instructure.com web page for the course. Please make sure you update your profile to include information about yourself (useful in team building) as **well as a clear and professional photo** (useful for the course staff to know who you are).

Lectures

We will meet for lecture on Mon/Wed/Fri at 10:45-11:30. The course meets in WEB 2230. **On some presentation days, we will use multiple rooms!** The class meetings will consist of formal lectures, student presentations, group activities, meetings, etc. **Attendance is required.**

Prerequisites

All students in CS 4000 must have completed all but four CS electives or required courses (not counting CS 4000/4500) and completed their upper division writing course.

Importantly students must:

1. Be expecting to graduate in spring 2018
2. Have completed their upper division writing course (or in progress – with approval).
3. Have completed CS 3505 – Software Practice II

4. Have four or less CS Elective, Theory, or Required courses left.
5. Have a strong desire to achieve an excellent Senior Project.

Text

There is no text that you have to buy. Occasional readings may be suggested.

Hardware/Software/Displays

Your senior project can be written in any language and run on any hardware, as long as your project is approved by the course staff. Often projects will run on a variety of platforms. You are welcome to program on CS lab computers or to use your own. It is recommended that students, where appropriate, utilize cloud resources for hosting their projects. Students will be expected to procure computational resources[1] including web and database servers (e.g., Azure, AWS). Teams will be required to use a Git Cloud server for project management. For the final demonstration day, you will be required to provide display equipment[2].

Consulting

All of the course staff will be available outside of formal classes to answer your questions and help with problems, but in general we expect that you and your team will strive to function as independently as possible. For sensitive issues, please come see Prof. de St. Germain and/or email him directly. For normal questions about grades/assignments/etc., a course based communication platform will be set up.

Class Announcements

Information and announcements for the entire class will be posted on the class web page.

Academic Misconduct

While it is highly unlikely at this point in your career that you are unaware of what is appropriate academic behavior, please be aware that the standard School of Computing policies apply. Individual assignments are to be done by the student alone. Team assignments are to be done only by the team. Team written assignments should be completed and reviewed **as a team**. Teams are required to document (and get approval for) any code/libraries/assets/etc. that are used in the project but not directly created by team members. Any material you leverage for your project must be documented.

Failure to abide by the highest ethical standards will be grounds for removal from the class with a failing grade.

The Pledge of the Computing Professional

All students in the course will be encouraged to take this oath.

---

[1] Some computational resources will be available via the CADE lab.

[2] We ask that you bring large screen monitors/TVs to help show off your work. Student teams without access to additional displays should contact the course staff about options.

Grading

Homework

Homework in CS 4000 will be a combination of written assignments, individual and group oral presentations, team meetings, self-evaluations, prototyping, etc. Most weeks you will be producing documentation and/or presentations about either yourself, your team, or your project. During the last month of the semester, your team will produce a prototype of your proposed software. The goal of the prototype is to show us the scope of your vision and convince us you can get there (in a more effective way than a static "design document" can). You will be graded on how well you perform in the total Software Development Cycle. *Final grades will be not only determined by your regular assignments, but will be greatly influenced by the quality of your prototype and your individual contribution to your team over the semester.* Students will be required to evaluate their team members as well as self-evaluate.

Exams

There will not be exams in this course. Teams will present their final projects to the class and to the course staff.

Grades

The Senior Project Class grades will follow the official University of Utah guidelines:

- A: Excellent performance, superior achievement
- B: Good performance, substantial achievement
- C: Standard performance and achievement
- D: Substandard performance, marginal achievement
- E Unsatisfactory performance and achievement

The final scores will be used to assign the final grade for the course. The instructor may curve the grades as necessary, but you can ideally expect the standard 90-100 A, 80-89 B, 70-79 C, etc., grading brackets. The following list will show the rough worth of each graded component of the course:

- 30% Presentations
- 20% Design Document
- 20% Prototype
- 10% Other Misc. Assignments
- 10% Effective Participation (classroom; team; assignments; peer grading; progress reports, etc.)
- 10% Overall Modifier

Overall Grade Modifier

During the final letter grade computation, the course staff will evaluate each team member based on the overall quality of the team project, the quality of their contribution, and their contribution as evaluated by their team peers. This will include the team member's contribution ***across the entire semester***. Individuals on teams that see all members contribute equally and which produce high quality work can expect 10 points in this category. Individuals on teams where the final project is not superior and/or

where certain team members produce less than others[3], can expect to see their final grade one or two steps lower than their teammates.

Questions you can ask yourself as you contemplate what your Overall Modifier grade will be:

1. Is our project demonstrably one of the top ones produced by the class?
2. Did I take a leadership role on the team?
3. Was the code I produced a significant portion of the overall code?
4. Throughout the semester, did I communicate to the team, make all the team meetings, use the team tools, and get my work done on time?
5. Did I contribute a significant amount to the non-coding requirements of the team?
6. Was my work, effort, and contributions at the top of my team?

Other Course Miscellaneous

Development vs. Research, Apply Knowledge vs. Gaining It

It should be stressed once again that the Senior Project is about utilizing the skills you have painstakingly accumulated over your years in the CS program. While it is expected that there will be "growing pains" as you develop your senior project, the "learning of new technologies" is not an excuse for not producing excellent work.

Further, the capstone project should not be about **research**. There should be a very high likelihood of successfully writing software to solve the project problem (e.g., a good project would be implementing a well-known algorithm; a poor project would be developing a new algorithm to solve a problem).

It should be noted that any project will no doubt require students to learn new techniques and acquire knowledge, but this should be ancillary to the project. A good example would be extending your knowledge of databases (gained from the DB course) to allow more tables with advanced joins or using an entity framework to tie code to data. A bad example would be to choose to learn a new language that you have never written in before.

A Note on Participation

Historically, students are known to miss deadlines, ask for extra time, "forget" minor assignments, miss appointments, etc. Often a professor will overlook these transgressions because he or she is focused on the academic learning that is taking place. As CS 4000 and CS 4500 attempt to imitate a real world environment (as much as possible) this behavior will not be rewarded; e.g., if you have a job, not showing up to the team meeting would be considered, at best, "bad form", not getting your boss a status update would be unacceptable, etc. Of course, sickness and accidents happen, but "spacing out" will result in a lowering of your grade.

Peer Grading

[3] It is strongly recommended that individuals join teams in which each member has the same level of skill, motivation, and available time. For example, teams in which one member spends 5 hours a week and another spends 15 hours a week will often find a discrepancy in final grades.

Some grading will be done by your peers using an aggregation system. Peer grading will often occur with evaluation of public oral and written communications. Also, once teams are formed, each week you will be asked to evaluate the performance of your teammates. Completing these evaluations will also be awarded credit. These grades will affect your final course letter grade[4].

Grading Scales

Grading in this class is holistic while attempting to match the official University of Utah categorizations. Below, the grading categories are listed, along with what an industry hiring manager might "think" about the candidate. These "bins" are used because at this point in your career numeric scores are somewhat arbitrary, and it is the high level category of performance that you will be known for. Grades are only listed because students like grades ;^)

- Superior (High A) - Hire this person today
- Good (Low A) - Hire this person
- Adequate (B) - Call in for another interview
- Needs Work (Low C) - Look for someone else
- Poor (Low D) - Something must have gone wrong
- Unacceptable (Zero) - Do not hire anyone from the UofU

Once teamwork begins, you will be asked to categorize your team members (and yourself) both by amount of work achieved and by teamwork/leadership qualities. The following list shows the general types of categorization that you will need to make of your fellow team members.

- Producing more than others on the team.
- Pulling their weight
- Working hard but some issues are keeping them from completely contributing
- Not working as hard as you would like
- Not effectively contributing to the project
- Not contributing to the project

The following bullets list the individual's leadership and teamwork contributions/relationships:

- Carrying more than their share, but in a way that is **not** conducive to the team's overall progress
- Taking a leadership role in the team (effectively supporting the whole team's efforts)
- Working well as a solid member of the team
- Causing some team relations issues
- Causing a complete distraction to the team

[4] It should be noted that the instructor has the final say on grades and, while unlikely, may override student given grades.

Team Work

The first third of the semester will be graded mostly on an individual basis. The second two thirds of the semester (once teams are formed) will be graded mostly on a team basis. In general, the same grade will be given to each member of the team, though as noted above, the course instructor will adjust individual grades of team members when an exceptional performance (good or bad) is given.

Students with Disabilities

Reasonable accommodation will gladly be provided to the known disabilities of students in the class. Please let the instructor know of such situations as soon as possible. If you wish to qualify for exemptions under the Americans with Disabilities Act (ADA), you should also notify the Center for Disabled Students Services, 160 Union Building.

Tentative Class Schedule

The class schedule will be posted on the class web page. Tentative weekly topic areas include:

Week 1: Introduction and Resume Creation
Week 2: Selling Yourself and Meeting your Potential Teammates
Week 3: Peer Interviews and Individual Ideation
Week 4: Team Formation
Week 5: Team Idea Creation
Week 6: Project Research and Discovery
Week 7: Team Presentations and Design Document Creation
Week 8: UI Sketches and Use Case Development
Week 9: Requirements Analysis
Week 10: Storyboard Walkthrough Presentation
Week 11: “First” “Complete” Design Document Due
Week 12: Prototype Stage 1 – Asset Accumulation
Week 13: Prototype Stage 2 – Required Infrastructure Development
Week 14: Prototype Stage 3 – Complete Path
Week 15: Final Presentation of Prototype Demonstration Day
Finals Week: Final Design Document and Prototype Due, Team Evaluation Meetings