SE490+CS493+GENE403 Capstone Design F2023

Prof Derek Rayside drayside@uwaterloo.ca

September 23, 2023

Course Descriptions

SE490 Course Description

Continuing from SE 390, students undertake a substantial customer-driven group project. Project groups establish and maintain project control processes, delivering a series of iterations on their initial SE 390 prototype. Adaptive methodologies are encouraged and supported.

https://ucalendar.uwaterloo.ca/2324/COURSE/course-SE.html#SE490

CS493 Course Description

Students work in teams on substantial open-ended computer science problems as part of the CS 493/494 course sequence. Lectures describe project management fundamentals and ethical and legal issues in computing. Students form teams, select projects, define project goals, perform risk assessment, establish a project plan, and develop a prototype. Possible project topics can include development of software systems, analysis of extensions to existing systems across the field, and experimental computer science.

https://ucalendar.uwaterloo.ca/2324/COURSE/course-CS.html#CS493

GENE403 Course Description

This is the first course in a two-course sequence where students from more than one engineering plan work together to solve an interdisciplinary engineering problem. In the course they will formulate and communicate an interdisciplinary engineering problem or opportunity; develop a feasible design project proposal and plan; generate feasible solutions that address the formulated problem; evaluate alternatives and identify preferred solution; address safety, regulatory, sustainability and professional ethics requirements, as appropriate; effectively manage design project technical and non-technical risks using project management tools and techniques; deliver a report and/or a presentation that summarizes the work completed; work effectively as an interdisciplinary team member and/or team leader.

https://ucalendar.uwaterloo.ca/2324/COURSE/course-GENE.html#GENE403

URLs & Repositories

Course Website	https://ece.uwaterloo.ca/~se_capstone/
Course Notes (Handbook)	https://ece.uwaterloo.ca/~se_capstone/se-capstone-handbook.pdf
Git Project Metadata	https://git.uwaterloo.ca/secapstone/se2024/capstone-fall2023 *(clone this)*
Course Discussion	MSTeams

Bootup (this is your first Learning Activity, worth 2.5%)

Due Sept 22

1. git clone https://git.uwaterloo.ca/secapstone/se2024/capstone-fall2023
2. cd capstone-fall2023
3. mkdir teams/team-OurTeamName
4. cp team-template/*.* teams/team-OurTeamName/
5. userids.txt — uw userids for all team members
6. meta.tex — project name, people display names, *etc.*
7. abstract.tex — Describe your best one. You can change.
8. make a merge request with your files, assigned to TA Ahmed El Shatshat ar2elsha@uwaterloo.ca

Mentor Meetings: Twice per Month

SE490: Ahmed will be your mentor.
GENE403: Derek will be your mentor.
CS493: Derek will be your mentor.

- Everyone should attend. (Perhaps on some occasions there is a good reason one person cannot make it.)
- Make the meetings as meaningful and as short as possible.
- Ideally in person. If online, then cameras on.
- Discuss your project progress and learning activities.

Weekly Workflow

1. Work on project.
2. Submit progress report on LEARN
3. Do learning activity.
4. Submit learning activity report on LEARN
5. Meet and discuss.

Learning Objectives

The capstone project is an opportunity to explore a significant subset of the learning objectives of the undergraduate degree in an integrated project. In broad overview, for capstone courses we might say the learning objectives are:

- To develop *technical leadership and judgement*.
- To do a project of *enduring value*, that has meaning to someone (including yourself) after you graduate.
- To develop *broad familiarity* with a wide range of techniques, tools, skills, concepts, and professional practices. These will come from a wide range of other courses, which you might not have taken, as well as from co-op experiences, industry, *etc.*.
- To be able to *select and apply appropriate* techniques, tools, skills, concepts, and professional practices for a particular project.
- To *communicate with colleagues* in a clear, correct, complete, and concise manner.
- To *give and receive constructive feedback*, both within your team and with other teams.
- To work in a *team*.
- To demonstrate appropropriate engineering *process maturity*. For anyone working on software-intensive projects, this will include using a version control system for your code. In other disciplines there are other ways of demonstrating process maturity.

Handbook §1

Learning objectives for the Bachelor of Software Engineering degree have been identified by the Canadian Engineering Accreditation Board (CEAB), the Canadian Information Processing Society (CIPS), and the UWaterloo Software Engineering Curriculum Committee. These are listed in the *Handbook* §1. Similar bodies have identified similar sets of learning objectives for other degree programs. Your project should align with the learning objectives of your degree.

Contribute Cover Art!

See past abstract booklets on the course website.

Calling all artists! Please contribute your artwork for the cover of our class abstract booklet. We have used student artwork on the cover for the last few years.

Modes of Assessment: Formative & Summative

In HCI (Human-Computer Interaction) we also use the concepts of *formative* and *summative* assessment of the software. For example, a *thinkaloud* is a formative technique where we learn about the software by listening to a user express their thoughts out loud as they are using the software. This gives us qualitative insights into how to improve the software.

A *summative* assessment of the software might measure the time it takes the user to do a task, or the user's task error-rate while using the software. This gives us a quantitative measure of how well the software works, but doesn't give us any insights into how to improve the software.

FORMATIVE ASSESSMENT: qualitative feedback to identify and help resolve misconceptions, struggles, and learning gaps. Helps student to achieve intended learning outcomes and prepare for future summative assessments.

SUMMATIVE ASSESSMENT: measure the degree of success demonstrated with the learning. Often quantitative. Tests and exams are the classic examples.

Modes of Instruction: Didactic & Dialectic

Didactic: a teacher-centered approach in which the teacher talks and students learn by listening.

DIDACTIC is the mode of instruction and learning for most engineering / science / math courses, and is characterized by:

- regularly scheduled lectures
- textbooks
- assignments / problem sets
- tests / exams
- most/all assessment is *summative*

Dialectic: discussion and reasoning by dialogue as a method of intellectual investigation. Everyone takes turns talking and listening. Students learn by talking, by listening to their peers, and by listening to the instructor.

DIALECTIC is the main mode of instruction and learning for capstone courses. It's about learning through discussion and dialogue: with teammates, with peers, with instructors, with ideas and texts.

- interaction and discussion
- self/team-directed learning
- broad reference materials
- presentations + Q&A
- much assessment is *formative*

Marking Scheme

Marks are divided 50/50 between formative and summative assessments. Marks are generally assigned to teams rather than individuals, unless otherwise indicated. In extreme cases, the contributions of individual students might be assessed individually.

There is a progression in the capstone courses. Earlier courses have greater emphasis on formative assessment, and later courses have greater emphasis on summative assessment.

Summative Assessments	*Weight*	*Due Date*
Individual quizzes	4%	
· on LEARN		
· open book (handbook)		
· not open friend — read on your own		
· at the knowledge/comprehension level		
Midterm demo	16%	Oct 20+27
· realization of some learning objectives		
· partial functionality / experimental prototypes		
Final demo	30%	Nov 24 + Dec 1
· realization of most learning objectives		
· MVP prototype or significant design analysis		
·	50%	·

Formative Assessments	*Weight*	*Due Date*
Team Learning Activities (8 × 2.5%)	20%	∼weekly/fortnightly
· bootup		Sept 22
· team health assessment + reflection		Oct 6
· team process assessment + reflection		Nov 17
· other activites you choose		
· other activites recommended to you		
Weekly Project Progress (8 × 2.5%)	20%	∼weekly/fortnightly
· progressX.md		
Peer Feedback (3 × 3.33%)	10%	
· midterm		Oct 20+27
· activity #2		Nov 3 + 10
· final		Nov 24 + Dec 1
·	50%	·

Shifting weight: Teams may choose to shift weight from formative activities to summative assessments. For example, skip a learning activity and shift the weight to the final demo.

Formative Assessment of Weekly Learning Activities

Everyone gets full marks for making an honest effort + follow through.

If you follow through an action, plan, or idea or follow through with it, you continue doing or thinking about it until you have done everything possible.

https://www.collinsdictionary.com/us/dictionary/english/follow-through

What we mean here by *follow through* is that you address any points or issues raised by your TA. The point of the learning activities is to stimulate your thoughts and your project and your interaction with your TA. So the learning activity might be the start of a conversation, not the end of one. If there are interesting ideas or perspectives that arise from the activity, then you should follow through and address them.

If there are challenging cases where your work lacks honesty, effort, or follow-through, then your TA can ask the instructor for help in assessing part marks.

Remember that the learning activities are not the project. They complement the project, and help you get more out of the educational experience. Let's consider the project itself next.

Formative Assessment of Weekly Project Progress

Everyone gets full marks for making an honest effort.

The purpose of these marks is to help keep you on track. For years students have said, at Symposium Day, that they would appreciate if we could find a way to help them stay on track through the summer. Let's discuss some aspects of honesty and effort:

- The normal model for course workload at UW is 10 hours/week.
  - For example, in a regular technical course: 3 hours lecture + 3 hours lab + 1 hour tutorial + 3 hours studying.
  - Other universities, such as McGill and MIT, explicitly rank different courses based on the number of credit-hours per week. So then meeting graduation requirements is not just a question of having enough courses, but also of having enough credit-hours, since different courses are weighted differently. At Waterloo courses are supposed to be more uniformly sized.
  - The usual arrangement for most teams in most weeks is something like this:
    - 1 hour for a learning activity (e.g., watching an old project video and writing a response)
    - 1 hour for reading course announcements, notes, etc.
    - 1 hour for meeting with TA
    - 1 hour for team sync-up/planning meeting

    * 6 hours for actually working on your project (designing, analyzing, programming, etc)
  - Some weeks some teams will have different arrangements. For example, if you want to patch the Rust compiler you might need to learn the Rust language, and so you might devote significant time to learning activities on that. Those learning activities should have some structure and some identifiable output. Perhaps you complete a Rust tutorial.

- SE463 deliverables are not part of SE490, although they are part of your overall project.
  - Many reputable universities would consider it dishonest to submit the same work for credit in two different courses.
  - There are opportunities for other requirements activities that you can do as part of SE490+CS493 that are not SE463 deliverables.
- Natural variation occurs, and if we are honest then we acknowledge it. It's not going to be a uniform effort every week.
- Scrum self-assessment of effort in the past week:
  - above average > 10 hours
  - average ~10 hours
  - below average < 10 hours
- Full marks: make an honest effort over the term.

Summative Project Evaluation: Midterm + Final

Handbook §14

Project evaluation is aligned with the learning objectives. A central premise is that the software should perform its intended function properly and in a unified way. Generally speaking, capstone projects should, in all relevant ways:

- exemplify the learning objectives;
- demonstrate the skills expected of a graduate;
- make appropriate trade-offs and judgments; and
- not suffer serious oversights.

The general goal is to have software that users can try by the end of the term, while applying good software engineering practices. Then you can gather user feedback over the fall coop term and revise your project accordingly next winter.

For some projects this goal doesn't make sense, and that's ok. Discuss with your TA or instructor.

There is some description of this assessment in the Handbook. The best way to callibrate yourselves is by watching old project videos. If you have questions about what to do next in your specific project, then ask your TA or instructor.

University Policies

Intellectual Property: UWaterloo has the (fairly unique) policy that intellectual property is owned by its creators (rather than by the university). The university has resources to help you commercialize your project (if desired), as well as local incubators such as Velocity.

https://uwaterloo.ca/secretariat/policies-procedures-guidelines/policies/policy-73-intellectual-property-rights

https://uwaterloo.ca/secretariat-general-counsel/faculty-staff-and-students-entering-relationships-external

Academic Integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility.

http://uwaterloo.ca/academicintegrity/

AccessAbility: AccessAbility Services collaborates with all academic departments to arrange appropriate accommodations for students without compromising the academic integrity of the curriculum. If you require academic accommodations, please register with AccessAbility Services at the beginning of each academic term.

https://uwaterloo.ca/accessability-services/

Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

Policy 70, Student Petitions and Grievances, §4, http://secretariat.uwaterloo.ca/Policies/policy70.htm

http://uwaterloo.ca/academicintegrity/

Discipline: A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (*e.g.*, plagiarism, cheating) or about rules for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean.

For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, http://secretariat.uwaterloo.ca/Policies/policy71.htm

For typical penalties check Guidelines for the Assessment of Penalties, http://secretariat.uwaterloo.ca/guidelines/penaltyguidelines.htm

Appeals: A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals).

http://secretariat.uwaterloo.ca/Policies/policy70.htm

http://secretariat.uwaterloo.ca/Policies/policy71.htm

http://secretariat.uwaterloo.ca/Policies/policy72.htm

Reconciliation

We acknowledge that the University of Waterloo is on the traditional territory of the Neutral, Anishinaabeg and Haudenosaunee peoples. The University of Waterloo is situated on the Haldimand Tract, the land promised to the Six Nations that includes ten kilometres on each side of the Grand River.

https://uwaterloo.ca/arts/about-arts/territorial-acknowledgement

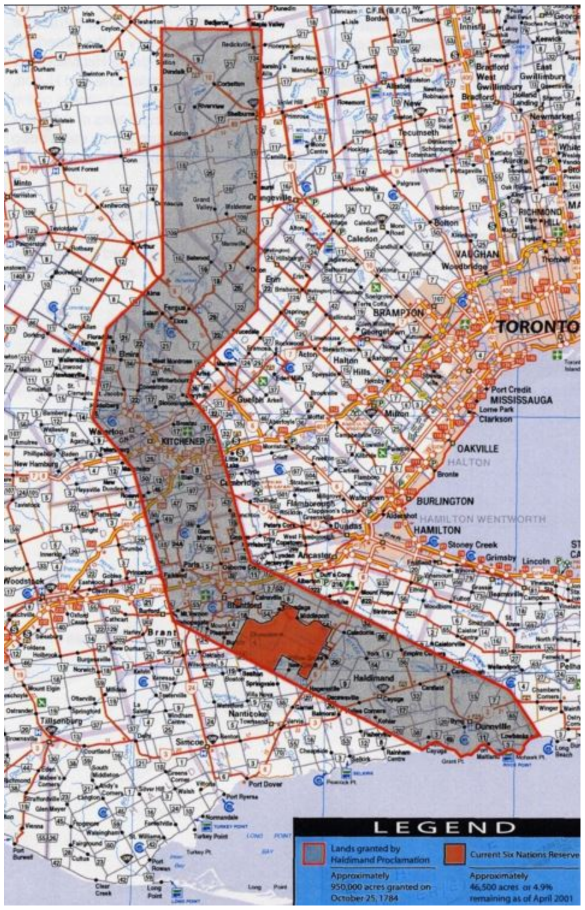
TORONTO
MISSISSAUGA
OAKVILLE
BURLINGTON
HAMILTON
KITCHENER
BRAMPTON
LEGEND
Lands granted by Haldimand Proclamation
Approximately 950,000 acres granted on October 25, 1784
Current Six Nations Reserve
Approximately 46,500 acres or 4.9% remaining as of April 2001

Figure 1: Contemporary map of the original Haldimand Tract and the remaining Six Nations Territory (red).

Midterm Evaluation Criteria

June 20

Baseline score is 9/12, for meeting a minimum standard:

Many marking schemes start from a baseline of 0 and then add for different categories. A problem with such marking schemes for open-ended projects is that the various categories might be more or less relevant for different projects.
This can result in a situation where the marking scheme incentivizes students to do work that's not important or relevant for the project. Already this term two teams have asked if they should do unimportant work because of how they imagine the midterm might be evaluated. Focus on what's actually important for the project.

- *Definition/identification* of problem/opportunity, solution strategy/-ies, potential benefits of solution strategy/-ies
- *Consistency*: between what you say (presentation), what you write (abstract), and what you do (code). Update your abstract.
- *Reasonable technical judgement.*
- *Reasonable project progress*: this could include a variety of things, including: code, tests, toolchains, wireframes, experimental prototypes, user interviews, legal agreements, *etc.*
- *Baseline process established*: using version control and one other tool (*e.g.*, JUnit, Jira, Trello, issue tracker, *etc.*). Quantify your usage of the tools in some way: *e.g.*, number of commits, number of merges, number of tests, test coverage, *etc.*
- *Discussing important rules of thumb.*

But don't waste time on unimportant things. How do you know if it's important? Does it influence significant decisions?

Three Positives: Describe three ways in which your project so far is beyond the baseline. +1 for each. Maybe you have something extraordinary that's worth +2, but that would be uncommon.
Talk about important and interesting things that you've done.

Common Negatives: These can cause point deductions, usually -1. There are many ways to demonstrate poor judgement, ignorance, poor communication, *etc.*, but these are some common ones.

- *Poor judgement:* There are many ways to demonstrate poor judgement. A classic example is using a key/value store (*e.g.*, MongoDB) — without explanation — when the data are obviously relational or time-series. The bigger problem is not being aware that the database technology is not suited to the task at hand.
- *Significant Inconsistency* between what you say (presentation), what you write (abstract), and what you do (*e.g.*, code). For example, if your abstract only mentions software, but it turns out that custom hardware is a major part of your project, that would be a significant inconsistency.
- *Ignorance of domain.* You should be the experts in the room in your project domain. If you appear ignorant of major issues in the problem/opportunity domain that's not good.
- *Spending too much time on unimportant things.*