Most programming you have done so far in school has been “throw-away” assignments, however, in the real world, you must live with the code that you write for a very long time—often multiple years. In this course, we focus on designing and writing code that you can work with over a long period of time. In the first part of the semester, you will build a moderate sized project, then spend the rest of the semester modifying it to meet new/changed requirements. As you do this, you will write about what you have done, how well your design choices worked, where and how they could have been improved, and why you think your current design will be effective going forwards.

Assignments and Grading

Your main effort in this course will be on semester long project with changing requirements. This project will be done in groups of four students. Once you set your groups, you may not change them, so please pick your groups carefully.

Your grade will be broken down into the following three parts:

Software Deliverables (45%) You will submit the code for your project at various times throughout the semester. The first deadline is posted on the course website. After the first deadline, you will be given your new requirements as well as a new deadline for those requirements. Your software deliverables will be graded based on their functionality and design.

Written Analysis (25%) Along with every software deliverable, you will turn in a written document which will cover two main points:

1. A retrospective on how your previous design choices impacted your work to meet the current set of requirements. This section should analyze not only where your good design choices made things easy, but also where your bad design choices made things hard. For both of these points, you should analyze how/why these design choices helped or hindered you. For bad design choices, you should discuss what you might have done differently in the past to avoid the problem this time around.

2. An evaluation of your current design, with an analysis of its strengths and weaknesses going forwards. This section should justify your current design choices, explaining why you think they will be beneficial to you in the long run. If you recognize weaknesses in your current design, you should discuss them—including an explanation of why they are there, and how you plan to fix them in future submissions.

These documents should not only deep analysis of the strengths and weaknesses of your design choice, but also be well written. Ideally, the retrospective section of submission N would connect back to the forward-looking analysis of submission N-1 (i.e., Did things you think would be beneficial actually end up helping you? Did the weaknesses you identified come back to bite you? Did you fix your weaknesses this time around?).
Individual Presentation (10%) Each group member will present one evolution of the project to the class (there are 4 evolutions, and 4 members per group). This presentation should be 11 minutes in length, and should cover similar information to your written document.

Class Attendance and Discussion (20%) You are expected to attend class regularly, and participate in class discussions. We will dedicate a significant amount of class time to discussion—in which we will discuss topics related to software design. Some of these will be general design considerations, others will be specific to the project stages. These discussions provide you with a chance to not only exchange technical ideas, but to hone your oral technical communication skills. Some discussions will include a reading; you will be expected to have completed the reading prior to class. For each class discussion about general design considerations, you are expected to bring 1–2 pages of notes that you have made in preparing for the discussion. These notes should be typed, and will be handed in hardcopy. Part of your participation grade will be from turning these in, part from actively participating in the discussion, and part from attending class. You have 2 free absences during the semester, after that your participation grade will be reduced unless you discuss the (valid) reasons for your absences with the instructor.

Note that in both the discussions and the written analysis, you are expected to address weaknesses as well as strengths. Claiming that your design is perfect and cannot be improved upon will lose you points, not gain them.

Note that there are no exams nor textbook for this class.

Academic Integrity

Academic integrity is very important, and misconduct will not be tolerated in this course. All students should already be aware of a few basic principles which govern academic integrity at Duke in general:

- I will not lie, cheat, or steal in my academic endeavors, nor will I accept the actions of those who do.
- I will conduct myself responsibly and honorably in all my activities as a Duke student.

If I suspect academic misconduct in my class, I will report you to the Office of Student Conduct. The Office of Student Conduct may ask us to attempt a Faculty/Student Resolution, which will typically result in a zero on the assignment in question if the offending student agrees they are responsible for the alleged misconduct. If a Faculty/Student Resolution cannot be reached, or if the student is ineligible for one due to a record of prior misconduct, the case will be adjudicated by the Undergraduate Conduct Board.

Concretely, the expectations for this course are that anything written you turn in (code + written analysis) will reflect the work of your group. I expect that you will not copy code/text from or share code/text with other groups, nor have people outside your group write any of this. You are welcome to (and encouraged to!) share design ideas during class discussions. If you choose to use ideas presented by other groups, you should give credit where credit is due: in your written analysis, when you discuss that element of your design, you should note who you got the idea from.

If you have any questions about what is an is not acceptable, you should ask me.