

Intro to Computer Science Midyear Project 2013-14

There is no midyear examination for this course; instead, a midyear project is required. Like a midyear exam, it is worth 10% of your course grade. You may submit any one of the three Standard Projects (see the separate Standard Project Description document for each) either individually, or in a team of two. Alternatively, you may propose your own project as an individual or team of two (see below). If you work in a team, you will each receive 70% of the points awarded for the project.

The due dates for the project are:

in class	Fri, Jan 3 (D) Mon, Jan 6 (GH)	Project Selection or Proposal and Partners Selected Final – no changes after this date!
in class	Tue, Jan 7	Project Proposals Modified and Approved
3 PM	Fri, Jan 10	Progress Report #1
3 PM	Tue, Jan 14 (DG) Wed, Jan 15 (H)	Progress Report #2
3 PM	Fri, Jan 17	Progress Report #3
High Noon	Wed, Jan 22 (D) Fri, Jan 24 (GH)	Final Project Delivery, including Final Progress Report

Scoring for the project is as follows:

Progress Reports (4):	10 each	40 points
Milestones 1-4 achieved:	15 each	60
Milestones 5-9 achieved:	10 each	50
Milestones 10-11 achieved:	10 each	<u>20</u> (not available for Alice)
Total possible:		170

At the completion of the project, you as an individual may elect to count any or all points you have earned over 80 toward your Q2 grade instead, at a rate of ½ Q2 point for every 1 Project point. If you complete your Project through at least Milestone 4, you may turn it in at that point with your Final Progress Report and receive full credit for any other Progress Reports not already overdue.

The score on Progress Reports 1-3, and on your final project delivery will be reduced by 10% if no more than 24 hours late, 30% for 48 hours, and 70% for 72 hours. No credit after that. There is a 5% bonus for final project delivery for each full 24 hours in advance of the due date, up to a maximum of 20%. Only school days are counted. Percentages are of the actual grade that would have been received; for example, what would have been a 90 if delivered on time is an 81 if delivered less than 24 hours late; the same project delivered 24 hours early is a 95.

The project should take between 8 and 10 hours for an individual to complete the Progress Reports and Milestones 1-4, and an additional 10-15 hours for Milestones 5-11. You will have all of the remaining class periods (plus your Exam block, of course) to work on your project.

You may submit completed Milestones for early evaluation through Wednesday, Jan 15. There are no specific due dates for any Milestone except the final due date. You should plan your own work, include your estimated completion dates on your first Progress Report, and then update them on subsequent reports. There is no penalty for missing your own estimated milestone dates.

Milestones are cumulative – each must include the functionality of all previous ones, whether you submitted the previous one or not. I will do my best to get you feedback within 24 hours, and almost always in 48. You may resubmit a Milestone for more credit once, but I will look at only the most recent Milestone submission when I evaluate your submission. So, it is a good idea to submit Milestones regularly, when you think they are complete.

See the MY Milestone Delivery assignment for your project on Canvas for specific submission requirements. Either partner in a team of two may do any submission.

Alternate Project Proposals

If you want to propose your own project, it should be very specific about what you intend to accomplish and how to assess whether you accomplished your goals or not. Your proposal should be similar to the Standard Project descriptions including the points per Milestone; feel free to cut and paste. It must have at least Milestones 1-4 clearly defined.

Academic Integrity

You may discuss algorithms and approaches with others in the class or elsewhere, but your code or circuits must be your own work. The same copyright standards apply to programs and circuits as to a written paper in an English or History class, or as in the computer industry; reading someone else's work to help you write yours is NOT permitted. I use a software evaluation program that is used to detect copyright violations and stolen code in the business world, and it's pretty good at spotting code sharing and the usual transformations code criminals use to try to hide their tracks (changing variable names, making non-functional changes to data or control structures, etc.). However, none of the underlying algorithms used in these programs or circuits will be considered "patented", as programs must inevitably implement similar algorithms to accomplish the given tasks.