The University Of Michigan, Ann Arbor
EECS285, Fall 2016
Course Syllabus

General Course Information

Course Schedule

Mondays, 3:00 – 5:00pm, 1109 FXB

Instructors

Instructor: Andrew Morgan
Email: morgana@umich.edu
Note: When emailing, please include "EECS285" in the subject
Office Hours Location: GFL107
Office Hours: Monday, 5:00 – 7:00pm

IA: Todd Goodall
Email: goodallt@umich.edu
Office Hours Location: TBD
Office Hours: Please see Canvas site for detailed schedule

IA: Patrick Eschbach
Email: pjeschba@umich.edu
Office Hours Location: TBD
Office Hours: Please see Canvas site for detailed schedule

Textbook

No Required Text – Pick one that suits your learning style, or use available web resources along with lecture notes, etc…

Course Policies

Attendance

You are expected to attend every class period. Additional material not included in the lecture notes may be discussed during lecture. Important discussions regarding course projects may also be held during lecture, including updates, clarifications, and guidance. Finally, quizzes will be given during lecture at unspecified times.

Course Grades

Your final grade in this class will be determined using the following criteria:

In-class quizzes:	20%
Programming projects:	80%
Total:	**100%**

As this is a *practical* programming course, we will be grading far more than program correctness. Specifically, programming style (always) and design (when applicable) will be considered heavily during grading. Programs will be graded such that style and design issues will be deductions from your correctness grade, and can be significant. Given the high percentage set aside for projects, it is in your best interest to ensure good style and design throughout the semester. In addition, other factors may result in deductions as well – for example, while we are not typically focused on efficiency in this course, a deduction could be taken if your code is overly complex or inefficient to the point of being a bad design.

Final Grades and Curve

The *goal* is for final grades to be assigned as follows:

90% to 100%:	A- to A+
80% to 89.9%:	B- to B+
70% to 79.9%:	C- to C+
60% to 69.9%:	D- to D+
Under 60%:	F

However, if this seems unreasonable at the end of the course, the scale may be adjusted, in the favor of the student. Under no circumstances will the scale be adjusted such that the "curve" is detrimental to the student's final grade.

Participation

While participation is not a part of your grade, class participation is encouraged. Interaction during lecture makes for a more informative and enjoyable class.

Unacceptable Collaboration (Cheating)

All quizzes and individual programming projects are required to be done individually. Students may not share project-related code with other students *for any reason*. To make the policy clear: No student may view any other student's code or specific algorithm design. Also, no student may provide, either on purpose *or* inadvertently, his or her code or specific algorithm design to any other student.

If any violation of this policy is suspected, it will immediately be submitted to the proper disciplinary body of the LS&A or Engineering colleges, which will make an independent decision and take action that is deemed appropriate after a full investigation.

Cheating is *not* tolerated in this course. Be extremely careful when printing your project to a public printer, and be sure to always pick up every page of every printout. Do not leave your code or design (soft copy or hard copy) accessible on any machine, lab, study room, etc.

If a project is designated as a group project, students within a group may freely share code and algorithm design. However, project information may not be viewed by or provided to any member of another group.

Programming Projects and Submission

As the programming projects make up the majority of your course grade, ensure that you provide enough time to successfully complete the project, prior to the due date. Lack of setting aside enough time for one or more projects is the **single most common reason** for poor performance in this course.

Programming project points, weights, and duration will vary, depending on the difficulty and amount of work required to complete the assignment. Project specifications will be posted when appropriate throughout the semester, and due dates will be set to allow more than enough time to complete the project. To encourage starting early to maximize the chance of success, bonus points can be earned by submitting early, and to allow for complex and busy student schedules, late submissions are allowed with a small penalty. Details of these policies are provided below:

- **Early Submission:** If you submit more than 48 hours prior to the assignment deadline, and score at least 80% of the maximum points, a "bonus" of an extra 5% of the maximum points will be given. For example, if you submit 49 hours before the deadline, and receive 16 out of 20 points (80%), your grade will be recorded as 17 (16 + 20*0.05). However, if you submit 49 hours early, but only score 15 out of 20 (75%), no bonus points are given.

- **Late Submission:** You may submit your assignment up to 48 hours later than the deadline, however, a deduction will be taken. The deduction will be 10% of the maximum points for the assignment. For example, if you submit 15 hours past the deadline, and score a 48 out of 50 (98%), your grade will be recoded as 43 (48 - 50*0.10). Note: Submissions are NOT accepted later than 48 hours after the deadline. You will NOT receive credit for any such submission.

Due to the flexible submission policy as described above, submissions later than 48 hours after the project deadline are not accepted for ***any*** reason, except for extended hospitalization that would prevent sufficient time to complete the project as documented by a doctor.

Note: Accidents and failures occur fairly often (disks crash, people accidentally delete their source code, files get corrupted, etc.). I recommend you use a version control system, such as Subversion, to maintain versions of your code in case something unexpected happens. Save often, and backup occasionally too. Accidents and failures will not be accepted as reasons for extensions or late submissions – again, submissions after 48 hours past the deadline will *not* be accepted for any reason.

To avoid tough decisions about "gray areas" all of these policies are set as described and will not be deviated from for "special cases".

Regrades

If you believe your grade on any quiz or project is unfair or incorrect, you may request a regrade. Your project will be graded a second time, with special attention to the area of concern. Regrade requests must be made within 1 week of the project grades being distributed and must include a written description of what you believe was unfair or incorrect so that the course staff can focus on that concern. For example, if project 1 is returned on a Monday, you have until the following Monday to request a regrade. Note: The regrade period starts when project grades are distributed, *not* when you receive or review your grade.

This semester, all regrade requests must be made on the course Canvas site. For a class this large, our course staff will include two IAs and multiple graders, and keeping track of regrade requests via email becomes very difficult and error prone. Therefore, the common Canvas site will be used so that all course staff can access the single set of regrade requests. Regrades via email will not be accepted.

Quizzes

When quizzes are given in class, they will likely be given at the beginning or end of the lecture period. Quizzes *can only be taken* at the time set aside in class. Quizzes can not be made-up for any reason, and extra time will not be given to students who are late. A minimum of one quiz grade will be dropped during final grade computation, to allow for unavoidable absences from class, since make-ups are not allowed.

Exams

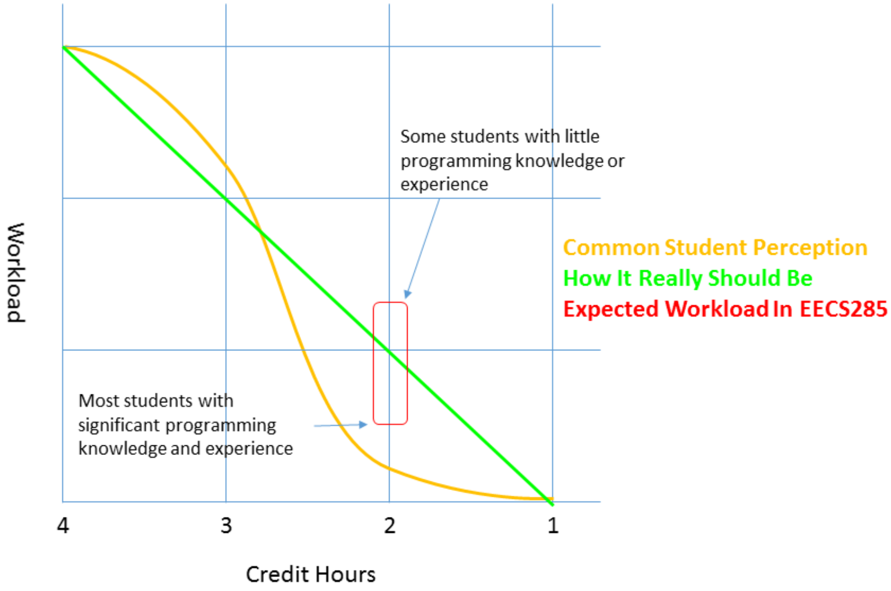
No exams will be given in this course.

One More Word On Grades

It is the student's responsibility to keep copies of all grades they received in the course. Without supporting proof of grades, the instructor's records are final.

Course Workload

As you all know, this is a 2 credit course – please realize that this does not imply the course is not a serious course that requires serious commitment. While most students perform well, every semester some students prioritize this course below other courses they are taking and end up performing poorly, being unable to obtain the grade they "need". The following nominal plot is based solely on my experience teaching this course, and is meant to reinforce the point that the projects in this course (especially the later ones) can be a lot of work in order to provide practical projects that demonstrate the importance of features of the language and of object-oriented concepts in general.


Workload
Some students with little programming knowledge or experience
Common Student Perception
How It Really Should Be
Expected Workload In EECS285
Most students with significant programming knowledge and experience
Credit Hours