

# EECS 285 Fall 2020

# Syllabus

## Instructors

Kristopher Smith

[krjgsmth@umich.edu](mailto:krjgsmth@umich.edu)

## Staff

Brendan Lichter

[blichtle@umich.edu](mailto:blichtle@umich.edu)

Meera Shivakumar

[mshivaku@umich.edu](mailto:mshivaku@umich.edu)

## Contact

Please direct technical questions to our Piazza forums. For administrative questions, you can reach the course staff at [eeecs285staff@umich.edu](mailto:eeecs285staff@umich.edu).

## Schedule

Wed

1:30 pm - 3:30 pm

Remote

You are expected to attend/watch every class period, and you are responsible for all material discussed during lecture. Furthermore, we will have online quizzes throughout the semester.

For office hours, please refer to the calendar on the course website. In order to ensure fair access to all students, we will not provide any help to students outside of scheduled class meetings, office hours, and Piazza.

## Textbook

We do not have a required or recommended textbook. Use any textbook you like, or use web resources in addition to lecture material. UM students get free access to [O'Reilly Safari Books](#), and you can find several Java textbooks there.

## Overview

This course covers practical programming in Java. The following is a tentative list of topics: Java syntax, expressions and statements, control flow, objects and types, inheritance, polymorphism, arrays, exceptions, interfaces, nested and local classes, generics, input and output, graphical user interfaces, threads, and mobile programming. This list is subject to change.

The prerequisite for EECS 285 is some programming experience. You do not need to have any experience in Java.

## Website (eecs285.org)

The first place to go for any course materials or resources is our course website at [eecs285.org](http://eecs285.org) (you'll need to sign in with your @umich.edu account). All course materials and assignments are made available there, and are considered **required** reading. Other resources such as Piazza and the Autograder are linked from the site.

Minimal additional information can be found on the course Canvas site. Make sure you are set up to receive course announcements on Canvas, since we may post critical information there.

## Forum

We will be using Piazza to host a course forum. You are required to read this regularly; it is the venue we will use for important course announcements and project clarifications. In addition, it will be a significant source of help and hints on the assignments and projects.

We do not answer technical questions via email. In order to save everyone time, we want all students to have the benefit of seeing each question and its answer, so please use the forum.

Do not post your own solutions, assignment or project code, test cases, or output to the forum. Also, **please search the forum before posting** to avoid questions that have already been asked and answered.

## Projects

Over the course of the semester, we will assign four programming projects. The first is a short project intended to give you some familiarity with Java. The latter three projects are more substantial Java programs, requiring you to design, implement, and test a solution.

## Corrections and Clarifications

We will post a Canvas announcement in the case of any critical corrections or clarifications. It is your responsibility to ensure you are able to receive Canvas announcements.

## Programming Environment

All assignments in this course will use the Java 8 programming language. You are free to develop your programs on any platform you like, but you may use only standard features in Java 8. You will need to install the [Java Development Kit](#) for your platform. We also recommend using an Integrated Development Environment (IDE) for writing your code. We recommend the [IntelliJ Idea Community IDE](#).

Later projects may use Android Studio instead. More information will be forthcoming when those projects are released.

Programming assignments **must** be submitted to the **course autograder**. We do not accept submissions over Canvas or email.

## The Autograder

We use a web-based autograder for project submissions. We provide 2 submissions a day per student with feedback; partnerships will receive 4 submissions with feedback per day. After you have exhausted your submissions with feedback for a particular day, you may submit any number of times, but you will not receive any feedback for those submissions.

For Projects 1 and 2, the feedback you receive will be which autograder tests passed and which ones failed. The autograder will show full output for the **first** test that fails, but every other test will just show pass or fail. The autograder tests are not intended to be a substitute for testing and debugging on your own, which is why we limit feedback.

Projects 3 and 4 are expected to have a significant user-interface component, which would be very difficult to grade automatically. As such, the only feedback we expect to be able to give you on the autograder is whether or not your code compiled.

We will grade your **last** submission for a project. It is your responsibility to ensure that your last submission is complete and working. We suggest downloading your files after making your final submission and rerunning your tests to make sure that you submitted what you intended. **Under no circumstances will we grade any submission other than your final one.**

Several of the projects will also be graded for good programming practices, both by hand and using automated tools. Specifically, programming style (always), design (when applicable), and test cases will be considered during grading. In addition, other factors may be considered as well – for example, while we are not typically focused on efficiency in this course, a deduction

can be applied if your code is overly complex or inefficient to the point of being a bad design.

You should **always** use good programming practices; do not wait until the last minute to "fix your style," as this will likely lead to your code breaking.

A program is correct if it behaves as specified in the project handout, for example, by generating the correct output. A program adheres to course principles when it uses techniques taught in lecture and asked for in the specification. Finally, a program should be readable by other programmers. We recommend adhering to common style guidelines, such as [Google's style guide for Java](#). We require that you adhere to the style and programming guidelines on the course website.

## Successfully Completing the Projects

We have found through many years of teaching experience in various courses that the most common reason for poor project performance is **not starting early enough**. Plan to do some work on the project every day and try to have it finished a few days ahead of the due date, because many unexpected problems arise during testing. In addition, the computing sites can become very crowded, making it difficult to get a computer to use, so plan for these things to happen.

The second most common reason for not doing well on the projects is not asking for help when you need it. We offer help in office hours and on the class forums. When you come to office hours, please be ready to provide access to your code, preferably electronic. Another good way to get help is to post a question to the course forum. Remember, if you find that you are stuck on a piece of your project for an undue amount of time, please see us!

An important skill in programming is to be able to test and debug your programs independently. As such, we will not debug your programs for you. Instead, we will help you try to figure out how to test and debug your program yourself. We will also ask you to demonstrate what testing and debugging techniques you have already tried, and what the results were, before providing any advice.

## Due Dates and Extensions

We expect you to turn in your projects by the due date and time (generally **8pm** on the day of the deadline), and we do not generally offer extensions. For example, we do not offer extensions due to crowded computing sites, long queue times or slow response times on the autograder, internet access problems, accidental erasure or loss of files, or outside conflicting commitments.

However, we recognize that unexpected events such as the above do happen. To account for them, we will allow each student a total of **3 late days** to be used on any project during the semester. That is 3 days total for the semester, not per project. These late days should only be used to deal with unexpected problems such as illness. They should not be used simply to start

later on a project or because you are having difficulty completing the project. Once your late days are used up, we will not accept submissions past the deadline.

**A late day is 24 hours.** This means that if the deadline for a project is 8pm, one late day gives you until 8pm the day after the deadline. **Submitting past 8pm on the day after the deadline consumes two late days.**

Submissions made by a partnership after the deadline use up late days from both partners. It is your responsibility to ensure that you and your partner agree on whether or not to use a late day. In the case that one partner has a late day remaining while the other doesn't, a late submission will only count for the partner who has a late day. For the partner who does not have a late day, we will use the last submission made in the time period they were allowed to submit.

To request an extension beyond the three free late days, you must discuss your situation with an instructor before the deadline and provide written documentation. Extensions will typically **not** be granted, even for computer problems, illness, family emergencies, etc. You can avoid most problems by starting the projects early and keeping backup files. If a family/personal emergency causes you to miss a significant number of days, please see an instructor to decide the best course of action. If you are having trouble understanding the material or starting a project, please come to office hours for help right away.

## Early Submission Bonus

For Projects 2 - 4, if your **last submission** is 2 days (or more) before the due date, you will receive bonus points calculated at 5% of your overall score for the project (including both autograder tests and hand grading). If your **last submission** is between 2 days and 1 day before the due date, you will receive bonus points calculated at 2.5% of your overall score for the project. For purposes of early submission, we use 11:59pm as the cutoff for each day. That is, if your last submission is by 11:59pm on Wednesday you will earn 5% bonus points for a project due on Friday two days later. If your last submission is by 11:59pm on Thursday, you will earn 2.5% bonus points.

Since submissions made by a partnership count for both partners, be sure to coordinate with your partner on whether or not to submit after the deadlines for early submission.

## Academic Integrity

You may give or receive help on any of the concepts covered in lecture or reading assignments. You are allowed to consult with other students in the class to help you understand the assignment specification (the definition of the problem).

You must complete Project 1 alone. You may complete Projects 2 - 4 either alone or with a partner. **For those retaking the course: if you submitted a project in a previous term, you may**

**not partner on that same project this term.** All programming assignments in this course are to be done by you or your partnership.

You may not collaborate in any way with people outside your partnership when constructing your solution; your partnership working alone must generate the solution to a programming assignment and must submit your code for grading together (i.e. your partnership may only submit one version of your code for grading). You are not allowed to work out the programming details of the problems with anyone outside your own partnership or to collaborate to the extent that your programs are identifiably similar. You may not derive your solution in any way from other solutions. (If you are retaking the course, you may reuse your own code.) **You may not share code, including making it publicly available in any form (e.g. a public GitHub repository), whether during or after the semester.** **You may not share test cases outside of your partnership as we consider your test cases part of your solution.** If you have any questions as to what constitutes unacceptable collaboration, please talk to the instructor right away. You are expected to exercise reasonable precautions in protecting your own work. Do not let other students borrow your account or computer. Ensure that the computers you use to access project code are password protected. Do not leave your program in a publicly accessible directory, neither during the semester, nor after. Take care when discarding printouts. You are still responsible for following these rules even after finishing the course.

We report suspected violations to the Engineering Honor Council. To identify violations, we use both manual inspection and automated software to compare present solutions with each other and with prior solutions. The Honor Council determines whether a violation of academic standards has occurred, as well as any sanctions. Read the Honor Code for detailed definitions of cheating, plagiarism, and other forms of academic misconduct.

## Rules for Partnerships

Working in a partnership is optional. For Projects 2 - 4, you may work either alone or in a partnership. The following rules apply to those who choose to work in a partnership.

If you work in a partnership, you and your partner will submit one assignment together. You will need to register your partnership for a project on the autograder before making the first submission for the project. You cannot change partners in the middle of one project. You may change partners only after a project is completed and submitted.

You and your partner must collaborate on all aspects of your solution. You may **not** split up the project and work on separate parts individually. **Both partners will be held equally responsible for anything submitted on behalf of the partnership.**

## DOs

- Do READ THESE RULES CAREFULLY before programming with another student. You must follow these rules, or risk being investigated for an Honor Code Violation.
- Do choose a partner from the current semester of this course.
- Do register your partnership on the autograder before making a submission for a project.

- Do submit one copy of the project together.

### **DON'Ts**

- Do not program with someone without understanding these rules.
- Do not partner on an assignment with someone who has already solved the problem.
- Do not share code or **test cases** with anyone other than your partner, or a staff member.
- Do not split the work in half. Both partners must work on all parts of the solution.
- Do not partner with anyone who is not currently enrolled in the course.
- **Do not partner with anyone on a project that you have submitted in a past term of EECS 285.**

## **Grading and Quizzes**

Your final grade is based on scores from programming projects and quizzes. The tentative point distribution is included in the following table. It is not likely that this will change, but circumstances might occur that would make changes necessary, at the discretion of the instructor.

Project 1	10%
Project 2	20%
Project 3	25%
Project 4	25%
Quizzes	20%

There are no letter grades for individual projects or exams. The final course letter grade is based on the total weighted score earned. Please note, we do not offer the opportunity for “make-up” or extra credit work to improve your grade.

### **Quizzes**

There will be several quizzes throughout the semester. These will be taken outside lecture hours. A minimum of one quiz grade will be dropped during final grade computation to allow for unavoidable absences from class.

### **Final Grades and Curve**

We expect to assign final grades 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 favor of the student. Under no circumstances will the scale be adjusted such that the “curve” is detrimental to the student’s final grade.

We do not round scores to the closest percentage when assigning grades.

## Regrade Requests

While we work hard to grade accurately, we sometimes make mistakes. If you believe we graded an assignment of yours incorrectly, you can submit a regrade request no later than one week after the graded work is originally returned. We will then regrade your entire assignment, which can cause your grade can go up, but it can also go down. Regrade requests should only be made if you feel a grading error occurred - not if you are unsatisfied with the grading criteria. For projects, we only allow regrade requests on the hand-graded portion -- we do not regrade autograder tests.

Please refer to course announcements for the regrade procedure and deadline for individual assignments and exams.

## Tips for Doing Well in the Class

You will maximize your grade, and learn a lot at the same time, if you:

- Attend all lectures (note that many tips and hints about projects are given during class!)
- Read the assigned readings (lecture slides, forum, web pages)
- Hand in your work on time (even if a program does not work, turn in whatever you have done)
- Start the programming projects early and come for help as soon as you need it
- Follow the program specifications carefully

## Accommodations for Students with Disabilities

If you think you need an accommodation for a disability, please let your instructor know **within the first three weeks of the term**. Some aspects of this course may be modified to facilitate your participation and progress. As soon as you make us aware of your needs, we can work with the Services for Students with Disabilities (SSD) office to help us determine appropriate academic accommodations. SSD (734-763-3000; <http://ssd.umich.edu>) typically recommends accommodations through a Verified Individualized Services and Accommodations (VISA) form. Any information you provide is private and confidential and will be treated as such.

## Commitment to Equal Opportunity



As indicated in the [General Standards of Conduct for Engineering Students](#), we are committed to a policy of equal opportunity for all persons and do not discriminate on the basis of race, color, national origin, age, marital status, sex, sexual orientation, gender identity, gender expression, disability, religion, height, weight, or veteran status. Please feel free to contact us with any problem, concern, or suggestion. We ask that all students treat each other with respect.

## **Students' Mental Health and Well-being**

University of Michigan is committed to advancing the mental health and well-being of its students. If you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. For help, contact Counseling and Psychological Services (CAPS) at (734) 764-8312 and <https://caps.umich.edu> during and after hours, on weekends and holidays, or through its counselors physically located in schools on both North and Central Campus. You may also consult University Health Service (UHS) at (734) 764-8320 and <https://www.uhs.umich.edu/mentalhealthsvcs>, or for alcohol or drug concerns, see [www.uhs.umich.edu/aodresources](http://www.uhs.umich.edu/aodresources). For a listing of other mental health resources available on and off campus, visit: <http://umich.edu/~mhealth>.

## **Research Disclosure**

Your class work might be used for research purposes. For example, we may use anonymized student assignments to design algorithms or build tools to help programmers. Any student who wishes to opt out can contact the course staff ([eeecs285staff@umich.edu](mailto:eeecs285staff@umich.edu)) to do so at any time up to seven days after final grades have been issued. This has no impact on your grade in any manner.

## **Right to Revise**

The course staff reserve the right to make changes to the syllabus at any time as they see fit. When a revision occurs, it will be announced, and it is your responsibility to be informed of such.