EECS 280 Fall 2019 Syllabus

Instructors

Jonathan Beaumont   jbbeau@umich.edu
Kevin Fu   kevinfu@umich.edu
Nicole Hamilton   nham@umich.edu
Sofia Saleem   sofias@umich.edu

Contact

Please direct technical questions to our Piazza forums. For other questions, you can reach the course staff at eecs280staff@umich.edu.

Schedule

Mon, Wed	9:00 am - 10:30 am	1013 DOW	Fu
Mon, Wed	12:00 noon - 1:30 pm	1013 DOW	Hamilton
Mon, Wed	1:30 pm - 3:00 pm	STAMPS	Beaumont
Mon, Wed	3:30 pm - 5:00 pm	2505 GGBL	Beaumont
Mon, Wed	4:00 pm - 5:30 pm	1210 CHEM	Saleem

For office hours and lab sections, please refer to the calendar on the course website.

Textbook

Course notes are available via our website at eecs280.org.

Optional: C++ Primer, by Lippman, Lajoie and Moo. 5th edition.

C++ Primer is available in print form, electronic Kindle edition, and for free through the University Library. If you decide to rely on the library version, please be aware that the number of simultaneous users is limited, and sometimes there may be a wait to access it.

Overview

This course provides a foundation for programming, the science of "how to" as a discipline. Students will learn a variety of techniques and principles to quickly write correct programs that are easy for others to understand and adapt to new purposes. This course focuses on programming in the small, writing short programs or program components that are easily understood by a single programmer. The concepts covered in this course are not just about C++ programming, but we will see how they can be concretely realized in the C++ language.

By the end of this course, a successful student will be able to:
- Take a problem and consider various possible approaches for solving it
- Select an approach or algorithm that provides for a simple, clean, well-structured solution
- Convert the algorithm into C++ code, using good design and style
- Test and debug the program using rigorous techniques
- Understand the concepts of top-down design, data encapsulation, information hiding, and procedural and data abstraction
- Design, implement and use complete classes, including constructors, destructors, and operator overloading
- Implement dynamic data structures for stacks, queues and lists
- Be able to quickly design, implement, test and debug a large scale project independently (1000+ lines of code).

Topics

- Functional abstraction
    1. Specification
    2. Recursion
    3. The recursion/iteration duality
    4. Recursive/iterative invariants
    5. Functional generalization

- Data abstraction
    1. Formal notions of type and type hierarchies
    2. Simple and aggregate types
    3. Abstract Data Types: ADTs
    4. Abstraction functions and representation invariants
    5. Polymorphism

- Dynamic resource management
    1. Creation and destruction

2. Rules of safety
3. Container types
4. Container iteration

In our exploration of these topics, we will discuss many elements of the C++ language, such as:
- Arrays (1-dimensional and multidimensional)
- File I/O
- Structs, classes, and objects
- Overloading of functions and operators
- Friends
- Strings (C-style and C++ string class)
- Pointers and dynamic arrays
- Templates
- Linked lists, stacks, and queues
- Iterators
- Exceptions
- Recursion
- Functors

Prerequisites

The prerequisite for EECS 280 is EECS 182 or EECS 183 or ENGR 101 or ENGR 151 or permission of instructor. The first project should be straightforward if you have satisfied the prerequisites.

Website (eecs280.org)

The first place to go for any course materials or resources is our course website at eecs280.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. A detailed schedule, including lecture topics, assignment due dates and exam dates, is also available there. Other resources such as Canvas, 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 encouraged to read this regularly and post technical questions as it will be a significant source of help and hints on the 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.

We prohibit posting your own solutions, project code, test cases, or output to the forum. Doing so is considered a violation of the Honor Code. 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 five programming projects. The first will be short and is based only on things we assume you have covered in prerequisites. The other projects will be longer and cover major course topics.

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

For this course, you must have a CAEN account. Anyone enrolled in EECS 280 is eligible for one, but it takes a little while to get it set up. (http://caen.engin.umich.edu/accounts/obtain) Our grading system uses the same version of the g++ compiler as on the CAEN Linux systems. At the time of this writing, CAEN Linux provides version 4.8.5 of the g++ compiler.

You are free to develop your programs on any platform you like, but you may use only ANSI/ISO standard C++, and are responsible for any differences between your preferred platform and the grading platform. We recommend testing your code on CAEN Linux before submitting.

The Autograder

We will grade your programs in an autograder system that is similar to the CAEN Linux environment. Your code must be submitted, compile, and run correctly on the autograder before the deadline.

We use a web-based autograder for project submissions. You will receive 3 submissions per day. Each submission will receive feedback from the results of the public test cases released with the project. Public test cases contain incomplete feedback; just because your code passes the public test cases, that does not mean it is 100% correct!

After the deadline, the autograder also runs a second set of private test cases that will be used to determine the correctness score for your submission. Your final project grade will be the score of the submission that received the combined **best** score on the public and private tests. If

multiple submissions share the best score on the tests, we will grade the last such submission.

The autograder also includes public tests for code style and good programming practices. For some projects, the autograder will also evaluate the thoroughness of test cases you submit.

Successfully Completing the Projects

We have found through many years of teaching experience 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!

One major goal of this course is for you to learn to test and debug your programs independently. We will not debug them 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.

Finally, always make multiple backup copies of your work! If you somehow lose your work, it is your responsibility. If you use source control, DO NOT make your code available in a public repository. This is considered a violation of the Honor Code.

Project Evaluation and Grading

Each project will be evaluated for behavioral correctness, adherence to course principles, and general style. Some projects may also be evaluated on the quality of test cases you supply.

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. For example, respecting the ADT interfaces in project 2. Finally, a program should be readable by other programmers. We have posted general guidelines for good style on the course web site.

Due Dates and Extensions

Programs turned in after the exact time and date on the assignment will receive a zero. 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.

Students adding EECS 280 after the Project 1 or Lab 1 deadline but before the Drop/Add deadline may petition to turn in Project 1 and Lab 1. Email the course staff immediately after adding the course if this is the case.

We will consider extension requests made in person and at least two weeks in advance, for example, for religious holidays or planned medical procedures.

Additionally, we will consider requests for extensions due to documented, unanticipated medical or personal emergencies. If you can't see the instructor in advance due to the emergency, then contact them as soon as you possibly can. In all cases, we require written proof of the emergency.

Academic Integrity

You must complete programming assignment 1 alone. You may complete programming assignments 2 - 5 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 give or receive help on any of the concepts covered in lecture, lab, or the textbook, and on the specifics of C/C++ syntax. You are allowed to consult with other students in the class to help you understand the assignment specification (the definition of the problem).

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, presuming it was wholly written by you and/or your partner and not derived from another source.) You may not share code outside of your partnership, including making it publicly available in any form (e.g. a public GitHub repository). 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.

Guidelines for Partnerships

Working in a partnership is optional. For programming assignments 2 - 5, you may work either alone or in a partnership. The following guidelines 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. However, you are free to work individually as much as you like or collaborate as much as you like, as long as it is with your partner.

DOs
- Do READ THESE GUIDELINES CAREFULLY before programming with another student. You must follow these guidelines, 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 guidelines.
- Do not partner on an assignment with someone who has already solved the problem. Students who do this will not learn as much as those who pair with someone at a similar skill level.
- Do not share code or test cases with anyone other than your partner, or a staff member.
- Do not split the work in half. It is important that both partners work on all parts of the program. Splitting the work may harm your or your partner's understanding of that part 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 280.

Lab Sections

In lab sections, you will gain practice with the material from lecture, with an emphasis on short programming tasks with a partner. The exercises assigned during lab section are intended to be completed during lab time. The work from lab will be turned in to the autograder and graded for

correctness. There is no submission limit for lab assignments, and the autograder only has public tests for labs. You may work with a partner for all labs except lab 1.

Grading and Exams

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

Lab exercises	5%
Programming projects (p1 4%, p2-p5 9%)	40%
Midterm exam	25%
Final exam	29%
Participation in course surveys, etc.	1%

There are no letter grades for individual projects or exams. The final course letter grade is based on the total weighted score earned. To pass EECS 280, your weighted average project score must be a passing score, and your weighted average exam score must be a passing score. If you score 60% overall, and your project average is above 60%, and your exam average is above 50%, you can expect to pass the course with a C or better. Final grades will be assigned based on the distribution of earned scores, consistent with past incarnations of EECS 280. In the past, the median student has received a grade in the B range. We may adjust our distribution and the threshold for passing if this semester's exams are more difficult than in the past. In particular, if everyone does great work, we'll assign everyone a great grade! Please note, we do not offer the opportunity for "make-up" or extra credit work to improve your grade.

iClicker Exercises

We expect to incorporate some iClicker exercises into our lectures this semester. This is new in EECS 280. Students really seem to enjoy these opportunities to participate, voting on an answer and then seeing whether they got it right and how they came out relative to others in the class. So, we encourage you to get an iClicker and bring it to the lectures. You'll have fun.

But we won't be grading the exercises. This will be the first time we're trying it with five sections, so it's still a bit experimental. Also, we were concerned that grading the exercises would amount to an attendance requirement we haven't traditionally had in this class.

Exams

We expect you to take both exams at the scheduled times. If you miss an exam, and a medical or personal emergency is not involved, you will receive a zero for that exam. If you anticipate an exam in another course or a religious holiday that conflicts with our exam time, you must notify the course staff during the **first** three weeks of the term using the alternate exam request form, which will be made available at the beginning of the term. The exam dates are given at the beginning of the term so that you can avoid scheduling job interviews or other commitments on exam days; hence job interviews, etc. are not considered valid reasons for missing an exam.

> *Midterm:*   *Wednesday, October 30, 7:00pm - 9:00pm*
> *Final:*   *Tuesday, December 17, 10:30am - 12:30pm*

Alternate Exams

We provide alternate arrangements for students who have a valid conflict with the midterm exam time, such as a conflict with a required activity in another course or university-affiliated athletics. As mentioned above, you must notify the course staff about such a conflict in the first three weeks of the semester.

For the final exam, the university schedules an alternate exam time for the course. The alternate exam is **not** elective: it is only available to students who have a final exam that conflicts with the regular time. You must notify the course staff during the **first** three weeks of the semester if this is the case, using the alternate exam request form. Aside from SSD accommodations and emergencies, we do not provide any other alternates. We do **not** provide another alternate exam if you can make neither the regular nor the university-scheduled alternate times. It is your responsibility to make sure this is not the case before registering for the course.

> *Alternate Final:*   *Tuesday, December 17, 2019, 1:30pm - 3:30pm*

We do **not** consider an exam on Central Campus that ends half an hour before our scheduled exam to be a conflict. Our exams start 10 minutes after the scheduled time (once called "Michigan Time"), so you will have sufficient time to make the transfer to North Campus for the exam.

Regrade Requests

While we work hard to grade accurately, we sometimes make mistakes. If you believe we graded an exam of yours incorrectly, you can submit a regrade request by the announced deadline for that exam. We will then regrade your entire exam, which can cause your grade to 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.

Projects and labs are automatically graded and are thus not open to regrade requests.

Please refer to course announcements for the regrade procedure and deadline for individual 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 and lab sections (note that many tips and hints about projects are given during class!)
- Read the provided readings (course notes, handouts, Piazza 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 during the first three weeks of the semester. 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. For a listing of other mental health resources available on and off campus, visit: http://umich.edu/~mhealth.