1 Basic Information

Instructors
Prof. David Chesney (am lecture), 4221 EECS, chesneyd@umich.edu, Office Hours: T TH 3:00-4:00pm
Prof. Farnam Jahanian (pm lecture), 2229 EECS, farnam@umich.edu, Office Hours: T 12:30-1:30pm, TH 3:00-4:00pm

Teaching Assistants (all office hours in MU3NE)
Graduate Student Instructor: Jianhui Wu, jianhuiw@eecs.umich.edu, Office Hours: MW 5:30-6:30pm
Instructional Assistant: John Bielawski, jbielaws@umich.edu, Office Hours: TTH 5:30-6:30pm
Instructional Assistant: Rachel Lamb, rmllamb@umich.edu, Office Hours: W 1:00-2:00pm, F 2:30-3:30pm
Instructional Assistant: Andrew Myrick, andrewmy@engin.umich.edu, Office Hours: M 4:00-5:00pm, W 7:00-8:00pm

Course web page (all sections): http://www.eecs.umich.edu/courses/eecs281/f04
Course forum/news group (all sections): umich.eecs.class.281

2 Course Overview

EECS 281 is an introductory course in data structures and algorithms at the undergraduate level. The objective of the course is to present a number of fundamental techniques to solve common programming problems. For each of these problems, we will determine an abstract specification for a solution and examine one or more potential representations to implement the abstract specification, focusing on those with significant advantages in time/space required to solve large problem instances. When appropriate, we will consider special cases of a general problem that admit particularly elegant solutions.

3 Prerequisites

Students must have obtained a grade of C or better in each of EECS/CS 203 and EECS/CS 280, or have equivalent knowledge of discrete mathematics, C++ programming, and familiarity with UNIX. Also, students are expected to have some experience with ‘make’ and ‘gnuplot.’ Students with questions about whether they have sufficient preparation for this course should speak with the instructor(s) as soon as possible. Due to the overwhelming number of students interested in this course, we will strictly enforce the prerequisites.

4 Reading List

This course has one required text: Data Structures and Algorithms in C++, by Michael T. Goodrich, Roberto Tamassia, and David Mount. There may also be some handouts that the faculty will provide. You are required to read the contents of the course website and visit the course forum regularly where we will post important course-related information.

5 Grading Policy

Your work in this course is composed of: attending lecture and discussion sections, reading assigned material, completing homework assignments, completing projects, taking a midterm exam, and taking a final exam. Final grades will be based on the total points earned on the homework, projects, and exams. Factors such as class participation may be used to adjust your final grade, especially if it falls on a borderline. The weight assigned to each category is as follows:

4 Homework assignments 10%
3 Projects 40%
Midterm Exam 25%
Final Exam 25%

There is a ‘gray area’ of several points around each specific numeric grade, within which a ± system is used. Two people getting the same numeric grade might therefore receive different letter grades for the course. For example, if a student is in one of these gray areas, her/his grade may go up depending upon demonstrated improvement in exam performance or exceptional class participation.

The grades of C- and D+ will not be given in this course.

5.1 Grading Errors

We make every effort to grade correctly, however we do sometimes make mistakes. Arithmetic errors can be corrected in person by your GSI. If you believe something was graded incorrectly, you may submit it for a regrade. All regrade requests must be made in writing (not email) explaining the technical reason(s) that would make a regrade necessary. All regrade requests must be made no later than five working days after the graded work is returned to the student. The work in question will be regraded carefully in its entirety, with consideration given to the written request. As a result, your grade might go up, or it might go down. This second evaluation is final.
5.2 Incompletes
Incompletes will generally not be given. In accordance with university policy, doing poorly in a course is not a valid reason for an incomplete. If you are having problems in the course or experiencing an unexpected circumstance, please talk to the instructor(s) as soon as you are able to do so.

6 Homework Assignments
We will assign four homework assignments over the course of the semester. Homework submissions must not include external materials (e.g., web downloads) unless specifically requested in the assignment. Homework is typically due in hard-copy in the 281 lockbox in room 2420 EECS. Due dates and times will be clearly indicated on the posted homework assignments. Late homework assignments will not be accepted for any reason. However, we will drop the lowest homework score from your grade.

7 Course Projects
Three projects will be assigned during the term. Project submissions must not include external materials (e.g., web downloads) unless specifically requested in the assignment. These projects will require substantial time commitment on every student’s part. However, we expect that effort spent on programming projects will help the student to gain a conceptual understanding of the material. We strongly recommend that students begin working on assignments early, both to lower stress and have more time to ask questions.

7.1 Project Grading
Projects are worth 40% of the course grade. The breakdown of points for each project grade will be specified in the project assignment. Categories include:
- functional correctness
- execution efficiency to obtain correct solution
- principles and practices (readable code, efficient algorithms, efficient implementation, comments, use of topics from class)
- documentation of your solution
- group and individual evaluations

7.2 Project Groups
Project 1 will be individual effort. Projects 2 and 3 will be completed by groups of 3 students.

7.2.1 Quitting
If a student feels that they are completing a disproportionately high portion of a project, then they have a right to ‘quit’ the group at a project boundary (e.g., after project 2 is turned in, but before project 3 is assigned). The student receives the group grade for the completed project. The ‘ quitting’ student will be reassigned to a new group of three or fewer students, at the discretion of the course staff. The resigning member must document the resignation in email or hardcopy, with cc to all group members and Prof. Jahanian. It is extremely important that you bring issues and problems concerning your team to Prof. Jahanian’s attention immediately.

7.2.2 Being Fired
Students are expected to participate wholly in their group to the benefit of the entire group. Students may be “fired” from a group by the majority vote of the remaining members. The process is as follows:
- Documented “gentle warning” of risk of firing in email or hardcopy, with cc to all group members and Prof. Jahanian, with specific work required for warned group member to remain in group;
- 3 calendar days elapsed time for compliance;
- Documented statement of firing in email or hardcopy, with cc to all group members and Prof. Jahanian.

The course instructors assign an appropriate grade to the fired group member. Fired group members must actively pursue and obtain membership in another group. The course instructors must receive documentation stating that student has been hired by another group. Students that don’t belong to a group do not receive a grade on the appropriate portions of the group project.

7.2.3 Evaluations
Each group will fill out a group evaluation at the end of each project. The group evaluation clearly evaluates the contributions of each group member to the project, and is signed by all group members. In addition, individual evaluations may be turned in for private dissection from the remainder of the group. Group and Individual evaluation forms will be posted on the web. Group members with disproportionately low group contribution will receive a maximum of 50% of the grade received by their group for the project. A significant project grade penalty is associated with not handing in group evaluation forms.

7.3 Turning in Projects
Each project is allotted 3-4 weeks. Projects will be handed in as described in the project specification. Late projects will receive a 10% penalty per day. A weekend counts as one late day. No project will be accepted after 3 late days.
7.4 Computing Facilities
All projects are to be written in C++. The student is free to use any environment to develop a program, but any submitted program must compile and run in the CAEN Solaris computing environment using the g++ version 3.3.1 compiler. You are responsible for making sure that your projects run correctly in this environment. Using CAEN Solaris computing as your development environment will reduce the chance of encountering nasty surprises when your project is graded.

8 Exams
There will be one midterm exam and a final examination. If you miss an exam due to a medical or personal emergency, you are expected to provide the appropriate documentation. If you need additional time to complete the exam because of a personal disability, talk to the instructor(s) at least 1 month before the exam date. We will hold joint examinations for all lecture sections. The exam dates are given at the beginning of the term so you can avoid scheduling job interviews or other commitments on exam days. Other than for medical or personal emergency reasons, no scheduling conflicts will be taken into consideration two weeks after the start of the semester. Outside commitments are not considered a valid reason for missing an exam. The midterm is scheduled for Thursday, October 21st, from 7:30-9:00 PM. The final examination is scheduled for Thursday, December 16th from 8:00-10:00 AM.

9 Getting Help
Your first and best option is to ask your question during the office hours of a member of course staff. The next best option is to post your question to the course forum which will be monitored regularly. Students are allowed/encouraged to post answers to forum questions; however, posted questions must not reveal solutions to the projects or homework questions. **You should not rely on private email to ask course-related questions. Members of the course staff may or may not answer questions sent by private email.**

We do understand that students sometimes have questions of a personal or sensitive nature. For such matters, we ask that you see Prof. Chesney or Prof. Jahanian during office hours. If you have a conflict with all posted hours, you should send us email to schedule another time.

10 Policy on Collaboration and Cheating
Professors Chesney and Jahanian are here to provide a nurturing environment for everyone enrolled in the course. However, acts of cheating and unacceptable collaboration will be reported to the Engineering or LS&A Honor Councils, as appropriate. Cheating is when you copy, with or without modification, someone else's work that is not meant to be publicly accessible. Unacceptable collaboration is the knowing exposure of your own exam answers, project solutions, or homework solutions; or the use of someone else's answers or solutions made public. This includes solution sets and student solutions from past incarnations of 281/380. This means that students cannot use previous solution sets, even if the solutions are your own.

At the same time, we encourage students to help each other learn the course material. As in most courses, there is a boundary separating these two situations. You may give or receive help on any of the concepts covered in lecture or discussion and on the specifics of C++ syntax. You are allowed to consult with other students about the conceptualization of a project, or the general approach for homework solutions. However, all written work, whether in scrap or final form, must be done by you or your partners, where applicable.

You are not allowed to work out the programming details of the projects or specific details of the homework problems with anyone or to collaborate to the extent that your programs/homework are identifiable similar. You are not allowed to look at or in any way derive advantage from the existence of solutions prepared in prior terms, whether these solutions are copies of former students' work or solution sets handed out by course staff. We will be using an automated program to correlate projects against each other and past solution sets.

If you have any questions as to what constitutes unacceptable collaboration or exploitation of prior work, please talk to the instructor(s) right away. You are expected to exercise reasonable precautions in protecting your own work. Don't let other students borrow your account or computer, don't leave your work in a publicly accessible directory, and take care when discarding printouts.