**MCS 253P: Advanced Programming and Problem Solving**

[**Schedule**](#qwbmcnhoxviw)[**Dr Klefstad's Office Hours**](https://docs.google.com/document/d/1nZhpZKntLqV5vW3gVPprwKUCalWObBAl9AsJqY81tlE/edit?usp=sharing) **TA/Tutor Office Hours** [**Advice to Students**](https://docs.google.com/document/d/1xjO9A3PEfej-a54YX-mIgCFI_Z71o22K2yvjbIzrXDU/edit?usp=sharing)[**Lecture Rules**](https://docs.google.com/document/d/1gbvOIEuU4P1tZKCylBKUG153MYX84KJJfW35-VXTDrY/edit?usp=sharing)

This course provides in-depth preparation for industry interviews requiring demonstration of problem solving and programming skills. Emphasis is on understanding problem statements, considering edge cases, developing effective test cases, designing correct solutions, explaining these clearly, analyzing their time and space complexity, suggesting improved algorithms, and implementing a solution correctly. [Watch this](https://www.youtube.com/watch?v=ko-KkSmp-Lk)

**Overall Goals**

This course will ensure that students are prepared for success in technical job interviews. This course will also strengthen programming and problem solving skills as well as both oral and written communication skills.

**Course Objectives**

Students completing this course will be able to:

1. Read and understand programming problem statements.
2. Identify edge cases for the problem.
3. Define effective test cases and expected results for a program solving the problem.
4. Design one algorithmic solution on paper (or whiteboard).
5. Analyze the time and space complexity of their solution.
6. Write nearly correct code on paper (or whiteboard) to solve each problem.
7. Explain their algorithm and/or program to others.
8. Implement, test, and demonstrate correct function of their solution.

### Textbook

* Ace the Software Engineering Interview: An Interview Preparation Framework to Land the Job You Will Love, Ryan Ylitalo
* Coding Interview Ninja: 50 coding questions with Java solutions to practice for your coding interview, Ekim Ouye
* Programming Interviews Exposed: Secrets to Landing Your Next Job 3rd Edition, John Mongan
* The Complete Software Developer's Career Guide: How to Learn Your Next Programming Language, Ace Your Programming Interview, and Land The Coding Job Of Your Dreams, John Sonmez

Recommended on-line reading:

* [Get that job at Google](http://steve-yegge.blogspot.com/2008/03/get-that-job-at-google.html)
* [Get that job at Microsoft](https://blogs.microsoft.com/jobs/how-to-land-a-job-at-microsoft/)
* [Get that job at Apple](https://www.forbes.com/sites/ciocentral/2011/03/08/10-secrets-to-getting-a-job-at-apple-google-or-microsoft/#1d38f67fe771)
* [Get that job at Facebook](https://www.facebook.com/notes/facebook-engineering/get-that-job-at-facebook/10150964382448920)

**Student Assessment**

Students will be assessed based on two criteria:

1. ***Weekly Homeworks*** – Students will prepare for weekly lab projects by solving homework assignments at their own pace within the week. Each weekly homework assignment will require solving two or three programming problems.
2. ***Weekly Lab Projects in Discussion –*** Students will solve one or two problems within a time-constrained lab session (dicsussion). Students will be given multiple problem statements at the start of each lab. They must read, understand, select a problem to solve. Then, in writing, they must identify edge cases, define test cases and expected results, design an algorithm, verify the algorithm will solve the problem correctly. On their own time after the lab, they will implement (on Openlab using Linux and gcc) and test a program which correctly solves the problem. Students will explain their algorithms, program, and test cases to their partner or TA. The details of how this will happen are still under development and subject to change over the quarter.

### Lecture

Attending lecture & discussion and participating in them is required. I will not repeat lecture material in office hours or on the discussion forum. Lecture is your one chance to experience the new material. I do not allow recording of any lectures. Students who miss lecture must make it up by doing extra assigned problems.

**Email to instructor**

Do not email me with questions about the course or course material. Post all course related questions on Piazza. This gives other students, TAs, and the instructor a chance to answer for you and for everyone else at the same time. If you have a \*personal issue\* or \*emergency\*, please \*do\* email me about it as soon as possible. I typically have 600 to 700 students each term and can get overwhelmed by email when I get too much. If you do send email, be sure to say which course you are in. I’m always teaching two or more classes.

**Giving up**

Courses often get hard while you are taking them. You may feel like giving up. Giving up will certainly result in you getting behind and getting an F course grade. I want to encourage you now, do not give up! You can do it, but you must persist. If you feel like giving up, come see me in office hours as soon as possible. If you wait till the end of the term it will be too late to do anything about it.

### Discussion Forum

Students should expect to spend significant time writing and debugging programs related to the homework assignments. All students are expected to read and post on the class on-line discussion forum. You can sign up [here Piazza Sign-up](http://piazza.com/uci/winter2019/mcs253p). You can visit the forum [here Piazza forum](http://piazza.com/uci/winter2019/mcs253p/home).

### Homework

I recommend students schedule three to four 4-hour blocks per week to work on programs from homework and lab. If you have a question about homework or course material, do not email it to the the instructor or the TAs. Instead, post your questions to Piazza. This allows everyone in the class a chance to answer it first, so you are likely to get a very quick answer. The instructor and TAs will also read Piazza and may answer there, but everyone will be able to see the question and answer which reduces duplicate questions. If you have a personal issue or want to discuss your homework or quiz score, come see your instructor or TA during office hours or email them if you can’t make office hours.

Helping other students learn is encouraged. Most interesting stuff happens in Q&A. You may post small program fragments on Piazza related to your question or your answer, but please do not post answers to homework problems anywhere.

I only answer primary questions on Piazza and usually only when the TA asked me to answer directly. I generally do not answer follow-up questions. Students and TAs: please do answer questions on Piazza if you know the answer.

**Homework Submission and Grading** are due one week after assigned. Homework will be assigned in each lecture and in each discussion section and is due within one week of the time it was assigned. HW is due Thursday before lecture following the week that it was assigned, for example, HW1 is due Thursday of week 2 before lecture. HW0 is important set-up and practice. Do it as soon as you can before the end of Week 1.

Never copy and paste code when you are learning. Writing/typing all of it is a valuable experience and there is no shortcut for it. I also recommend avoid taking pictures to take notes in lecture either even though it appears like a handy shortcut. After struggling, you will be tempted to look at that picture sooner than you should. Lecture slides, your notes, Internet sources, pictures may be helpful to get past a difficulty but, but you should not use them until you are absolutely stuck. What are you going to do when you can’t find help? You need to be able to construct programs yourself. One goal of education is to have you develop this skill.

### Submitting your Homework and Report

[**Here**](https://docs.google.com/a/uci.edu/document/d/1rp3-oBsKPFDFC3IWcL3zYHRIJYS_aTgfG1hq0O3PXD0/edit?usp=sharing) **is how to submit your homework assignments to the EEE/Canvas dropbox.**

[**Here is the sample format to use for your report for gradescope.**](https://docs.google.com/document/d/1iS4DxMOrhErOHHX2DizFkgpvLR7VVbZ_kuUnL74dd1U/edit?usp=sharing)

It is your responsibility to ensure that your submission is what you intend to submit. Every term we have students who submit the wrong file. We will no longer allow you to resubmit at a later time without penalty if you submit the wrong files. You should verify that what you sumitted is correct. You may submit multiple times, but only your last submission will be graded and the time of that last submission will be used to determine when it was submitted for on-time or late credit.

**Quizzes (Q)** will be given weekly in lecture (usually on Thursday at the end of lecture) as scheduled above. Quiz coverage is the homeworks and lectures since the previous quiz. Quiz 1 will be given Thursday of Week 2 at the end of lecture. Quiz 0 is a practice quiz to give you a sample of what a quiz will be like. I suggest you practice by printing it, then doing it at home with a 15 minute timer. Also, each week when you start on your homework, write your entire first draft of each program on paper while trying to get the logic and syntax correct. Enter what you wrote, then notice the errors as the compiler tells you about them. Note what mistakes you made and correct them.

You must bring valid picture ID with you to all exams (quizzes). We will check everyone’s ID during the final quiz. Anyone without a valid picture ID may be photographed during the quiz and must bring a valid photo ID to office hours to get credit for the final quiz. I prefer a valid drivers license, but a UCI student ID **that is clearly readable** is also acceptable. If your UCI ID is faded so that any of these three - ID number, name, or picture - is not clearly readable, it is not acceptable for ID verification.

There are no make-ups on regular quizzes unless you are seriously ill and under care of a doctor. You must provide official, dated documentation from your doctor saying you are too ill to take the exam on the day it is given.

There are no make-ups on the final quiz.

**Regrades:** If you have any issue with grading of your homework or quiz, you must contest it within one week of the grade being posted (e.g., on Gradescope). The only issue for regrade are clear mistakes in grading where a correct answer was mistakenly marked wrong (this does happen) or where points were incorrectly totalled (this is more rare).

You may contest your grade on each homework or quiz at most once per item through gradescope.

There is no re-grading after 10th week. Note also, I never second-guess the TA’s grading and amount of points deducted for a given error. That decision is up to the TA/Reader after discussing grading with me beforehand.

### Computing Platform (operating system, editor, compiler, debugger)

We use openlab.ics.uci.edu computers to write and debug your programs. These computers use the Linux operating system and the command processor is called bash. If you have never logged into the ICS machines before, you may need to authorize your account first <http://www.ics.uci.edu/~lab/students/acct_activate.php>

You can [set up ssh keys](https://swiki.ics.uci.edu/doku.php/accounts:ssh_keys) or use VPN with ssh to connect to openlab computers from off campus.

We use the gcc compiler invoked with the command g++. [Here](https://docs.google.com/document/d/1ixkx1elCOKUW-Kt7aB1EQ4Jr_dp6G_QDb6xNYckDA6k/edit?usp=sharing) is how to log into and use the C++ compiler on our openlab computers.

You can set-up a password reset [here](https://support.ics.uci.edu/ltb).

If you accidentally delete your files, you can recover them yourself [read about it here.](https://www.ics.uci.edu/computing/services/snapshot.php) If you just want a fresh start with your .bash\_profile, you can copy a new copy with cp

cp /opt/local/etc/skel/example.bashrc ~/.bashrc

If putty logs you out, look at the Keepalive values in Settings under Connection. Set them to longer to keep you alive longer while you are idle. [Here is how to](http://subin.me/blog/how-to-avoid-ssh-session-timeout-in-putty/) disable timeout on PuTTY - the annoying way it disconnects you if you let it sit idle for too long.

Use g++ on openlab.ics.uci.edu as that is the only platform where we will provide help. You can find excellent tutorials (cheat-sheets) on-line using Google. Example searches include “vim command summary” and “linux command summary” if you want to find quick reference sheets. [Here is one for vim that just gives basic commands.](http://www.radford.edu/~mhtay/CPSC120/VIM_Editor_Commands.htm)  [Here is another one for vim that is more exhaustive.](http://tnerual.eriogerg.free.fr/vimqrc.pdf) [Here is one for linux bash commands](ftp://ftp.psu.ac.th/pub/bash-howto/reference_bash-cheat.pdf). [Here is one for bash scripts.](http://www.disi.unige.it/person/MoggiE/PG1-13/bash.quickref.pdf) [Make tutorial.](http://mrbook.org/blog/tutorials/make/)

Valgrind is a valuable tool for detecting run-time errors with your C++ program. You can run it on Linux (to get basic checking) with the following command

valgrind *program\_name program\_arguments*

or to get very careful checking

valgrind --tool=memcheck --leak-check=yes --show-reachable=yes --track-origins=yes *program\_name program\_arguments*

You must run valgrind on y our program for part of your submission report.

You can use scp or ftp to transfer files between computers (like openlab and your laptop).

If your only programming language experience is in Python, I suggest you read this tutorial on C++ arrays [Tutorial](http://www.cplusplus.com/doc/tutorial/arrays/). There are many videos on C++ such as this one [Learn C++ Video](https://www.youtube.com/watch?v=Rub-JsjMhWY). Generally, [this site](http://www.cplusplus.com/doc/) is useful and where you should start when you want some help from the Internet. Don’t waste time on stackoverflow.com because it is mostly novices posting answers for other novices.

### No Group Work!

All programming work is to be completed individually. There is no group work or group collaboration allowed in this course. Discussing concepts with others at a high level is encouraged, but **it is cheating to work on homework programs together, or to copy parts or all of your programs from any source including another student (current or previous), or from the Internet, or from a book**. We will use MOSS to check for cheating on homework submissions and it doesn't amazing job of detecting when works were derived from a common source even if you rename all the variable names and function names or reorder the functions in the file. Anyone whose homework is flagged as a copy will receive a zero for that homework submission. Guard your programs and do not share them with anyone. If you copy any code from any source, MOSS will catch it.

### Cheating

Familiarize yourself with UCI Academic Integrity Policy <http://honesty.uci.edu/> Anyone caught cheating in this class will be submitted to the University for prosecution and if found guilty, will receive a course grade of F. Students can be expelled from University for cheating. [Here are some important examples of violations](http://www.ics.uci.edu/ugrad/policies/Academic_Honesty.php). DO NOT POST SOLUTIONS TO ANY ASSIGNMENTS ON-LINE - especially on github. This is facilitating cheating which is very serious form of cheating. Posting homework solutions on-line ruins the homework for other students. Homework is where students do most of their learning. If they see (or worse use) a solution, it ruins their chance to learn from that assignment.

**Schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Reading** | **Lecture** | **Lab**  **Assigned** | **Homework Assigned** | **Exam** |
|  |  | Before Start |  | [HW0](https://docs.google.com/document/d/1hadmThgYa0MiOHSF58vY8MQYm7Boowj7ftx84LGiihc/edit?usp=sharing) |  |
| 1 |  | What are top companies seeking?  [Refresh of C subset of C++](https://docs.google.com/document/d/1l1d5buJlWqpegRX0IcIIEATIa5LBkJqvz8plbOqlRZs/edit?usp=sharing) | [LAB1](https://docs.google.com/document/d/1fpNi0zcFLiTdqhos-EjhnwGMK0zm9LZ3k-NaDMmtPi8/edit?usp=sharing) | [HW1](https://docs.google.com/document/d/1JY1bUgJeSflwIZLNzodce9cvJWGXZ6bREhvOxPSBUvw/edit?usp=sharing) |  |
| 2 |  | String processing problems | LAB2 | [HW2](https://docs.google.com/document/d/1ghHcpcUQoLzXn-5d3oBSwjqE45G2JXq3lppYlNSbymQ/edit?usp=sharing) | Q1 |
| 3 |  |  |  | HW3 | Q2 |
| 4 |  |  |  | HW4 | Q3 |
| 5 |  |  |  | HW5 | Q4 |
| 6 |  |  |  | HW6 | Q5 |
| 7 |  |  |  | HW7 | Q6 |
| 8 |  |  |  | HW8 | Q7 |
| 9 |  |  |  | HW9 | Q8 |
| 10 |  |  |  |  | Q9 |
| Finals |  |  |  |  |  |

**Possoble topics**

|  |
| --- |
| Introduction: What are companies and interviewers seeking? |
| String processing problems |
| Numerical problems |
| Search problems |
| Big data problems: hashing and caching |
| Recursion problems: divide and conquer, dynamic programming |
| Multithreading problems |
| Network Communication problems |
| Handling difficult problems |

**Course Grades**

|  |  |  |
| --- | --- | --- |
| **What** | **Number** | **Percent Grade** |
| Lab Interviews | 2 each week | 20 |
| Lab Programs | 2 each week | 30 |
| Homework Programs | each week | 30 |
| Quiz | each week | 20 |
| **Total** |  | **100** |

Course letter grades will be assigned by the default straight scale in the EEE/Canvas gradebook.

Out of 100% scale:

A+ ≥ 97.5 > A ≥ 93.5 > A- ≥ 90

B+ ≥ 86.5 > B ≥ 83.5 > B- ≥ 80

C+ ≥ 76.5 > C ≥ 73.5 > C- ≥ 70

D+ ≥ 66.5 > D ≥ 63.5 > D- ≥ 60 > F

**IMPORTANT: There is no other extra credit in this course.  The university only allows grade adjustments for grading errors on our part.  Please do not ask us to allow you to do "anything" to improve your course grade after it is calcuated.  By asking for such special accomodations, you are putting us in a very difficult position.  All such requests will be ignored or pointed back to this note.  The time to think about doing "anything" is during the course when you can make changes like**

* **attending lecture faithfully and on time, reading assigned reading before lecture,**
* **paying attention in lecture and taking hand-written notes on things you don't already know,**
* **ask questions when you don’t understand,**
* **studying more effectively before exams (like asking yourself hypothetical questions),**
* **reading exam questions very carefully and giving clear, correct answers,**
* **starting homework & other assignments early and ensuring they are completed on time or early.**
* **Sadly, many students won’t read this, but if you do, you are on the right track.**