CS344 Syllabus

Course: CS344 - Operating Systems I
Instructor: Tyler Lawson
Email: lawsonty@oregonstate.edu

Course Materials & Schedule

The course material is presented over the course of four Blocks, the culmination of each being a specific, detailed programming assignment geared towards the material covered. Our required material includes videos and articles, all written by the Instructor.

In lieu of a week-by-week schedule, please use the assignment list found at the bottom of this Syllabus: it lists the due dates of the five assignments, the Syllabus Quiz, and the Final.

Please see our Home Page for links to the material.

Here are a couple of good books which are entirely optional (we require neither homework nor readings from these):

William E. Shotts, Jr., *The Linux Command Line 13.07*, LinuxCommand.org (free download!)
Michael Kerrisk, *The Linux Programming Interface (TLPI)*, No Starch Press, 2010, ISBN: 978-1-59327-220-3

Assignments

There are five programming assignments in this course: four complex, and one simple. The programming assignments must be completed on our class server (see the Home Page for more) and then submitted to Canvas.

When you submit the programming assignments to Canvas, don't be alarmed that Canvas renames the files. This is just normal Canvas behavior. We have to rename them anyway, so don't worry about it.

I highly recommend that you read the programming assignment for each Block when it becomes available. *Start the assignments immediately* - they will take time to accomplish. I promise they're interesting! When submitting assignments, please be aware that neither the Instructor nor the TA(s) are alerted to comments added to the text boxes in Canvas that are alongside your assignment submissions, and they may not be seen. No notifications (email or otherwise) are sent out when these comments are added, so we aren't aware that you have added content! If you need to make a meta-comment about a submission, please include it in the submission .zip file itself, or email the person directly who will be grading it (see the Home page for grading responsibilities).

You must use *only* our class server, as described on the Home Page, to test your programs and homework for this course. You may of course use your own computer to do development work on, but everything must compile and be runnable on the course server to earn points. More importantly, do not use *other* OSU servers to run our class assignments on, as much of our software will crash the server; this is why we have been given our own machine! If you fail to heed this requirement, running our software on a non-class server will hurt your grade!

Tests

Here is some information about testing in this course:

- There are no mid-term exams in this course.

- Our Final Exam is an online Quiz here in Canvas. As with all of our assignments, you will not use *ANY* form of proctor for this test.
- Since the Final is online only, you can take it from anywhere you want!
- The Final is entirely multiple choice.
- In terms of materials you can have with you, the Final is open book, open notes, open compiler, open server, but NOT open group (i.e. you have to take it by yourself).
- The Final should take (far) under two hours.
- You'll be able to take the Final during a portion of Finals Week: in fact, you can already see the precise availability dates on the Assignments page here in Canvas. You can take it any time in between the listed end points.
- There is no particular study guide for the Final. The topics covered are everything in all of the lectures and homework, except Program Py (Python) which is NOT covered at all on the Final.
- To review, I recommend reading through all of the lecture slides.

Grading

Many of the assignments will be using grading scripts provided to you, and all have a set of specifications given that you must adhere to to both learn the material and earn the points possible. To assign grades to your submissions, the graders will be following a set of grading instructions (that you'll have access to). Note that the graders may run additional tests not communicated beforehand to verify that your program is adhering to the specifications, though nothing will be added to the specification. For example, if a grader suspects that a submitted program has been written in such a way that it passes a listed grading test, but does not adhere to the specification, then an additional test is likely to be run. However, any additional test done shall be easily justified as checking that the program adheres to the specification. The amount of points to be awarded or taken away by these additional tests is at the discretion of the grader. Any crashes, hangs, errors, infinite loops, etc. not covered in the grading instructions and/or grading scripts will cause your program to lose points. The amount lost depends on the severity, how much it affects the rest of the program, and how it is recovered from, if at all, all based on the discretion of the grader.

If you have grading questions about the homework, you MUST contact your grading TA, as the TAs do ALL of the grading (except for the final, which is graded automatically by Canvas). You can see the contact information for our TAs on our Home Page.

If you believe a grade returned to you is incorrect, please submit proof to your grading TA within 48 hours of the grade being received by you. Your proof must consist of screenshots (not a copy/paste of the text) that clearly show you being in bash, on os1, and executing the grading method as proscribed. Your screenshots must show the places where the grading has been done incorrect, if any, and this grading must be done on the submission that you have made (i.e. redownload it yourself from Canvas to make these screenshots). If you don't provide proof, or if you ask for a regrade past 48 hours of the grade being given, a regrade won't be done.

We do not use any sort of proctoring for any assignment or test in this class.

All assignments must be submitted on Canvas, according to the posted due date and time, or they will be subject to penalties. All programming assignments that are submitted late by less than 24 hours will have 10% deducted from their grade (e.g. your program submitted at 12:01pm, if it was due at 12:00pm, will be worth 90% of its graded value). Programming assignments submitted late equal to or more than 24 hours, but less than 48 hours, will have 25% deducted from their grade. Programming assignments may not be submitted late past 48 hours, and will be worth 0 points. These late penalties are off of the total possible, not the amount you earn. For example, 10% off from Program 1 will always be -16 points, regardless of the points otherwise earned.

Note that Canvas has three types of dates in relation to Assignments: the "available" date, "due" date, and "available until" date. The due date is the date that the Assignment must be turned in by for full credit. The other "available" dates allow me to control when the Assignment can be accessed, which helps keep all of the students in the same place at the same time. The "available until" date is used to prevent submissions of Assignments past 48 hours; it is NOT the due date: it is *after* the due date.

The Final Exam and Syllabus Quiz cannot be submitted late.

To get an extension on an assignment, you'll need to have a major event occur in your life that will prevent you from timely completing your work, and you must then notify our Lead TA (listed on our Home Page) before hand, if possible. Extensions of these kinds are generally reserved for issues you can't control, such as medical reasons or family emergencies. Merely being busy does not count! If you cannot notify the Lead TA before the event occurs (sudden severe sickness, for example), then you must make contact as soon as possible to get an extension. We're fairly easy going about these, but you need to be upfront and immediate: don't wait!

There won't be a curve applied to the grading of this course, nor is there any rounding or weighting of assignments and/or the final. The points you see in the assignments are the points available, including in the table below. The grading scale is as follows, and will be adhered to strictly (I have already taken into account some pretty generous rounding), so please don't ask for "just a few more points":

91.5 ≤ A ≤ 100

89.5 ≤ A- < 91.5

87.5 ≤ B+ < 89.5

81.5 ≤ B < 87.5

79.5 ≤ B- < 81.5

77.5 ≤ C+ < 79.5

71.5 ≤ C < 77.5

69.5 ≤ C- < 71.5

67.5 ≤ D+ < 69.5

61.5 ≤ D < 67.5

59.5 ≤ D- < 61.5

0 ≤ F < 59.5

Note that when you take the final, you won't be able to see how you did on the individual questions and answers, nor will they be given afterwards. It'll simply give you a point grade, and that's it.

Academic Honesty

You must follow these rules and regulations (Links to an external site.)Links to an external site. while working on assignments at OSU and for this course.

I have no problems with you working together to solve problems, work through coding bugs, etc. However, I do require that final assignment and test answers be YOUR OWN WORK; do not complete the assignment as a group project.

You may *not* turn in work that has a substantial amount of someone else's program code, except for code that I give you. If you do submit such work anyway, your submission will be reported to the College of Engineering for disciplinary action, and a preliminary 0 grade will be entered in for that assignment, which grade will be finalized only when the College makes its ruling (which might not be for a few months).

We will *automatically compare* what you turn in against all other submissions, including this term, previous terms, other sections, and from online sources.

To be clear: do not download someone else's code, change it, and then submit it, even if you cite what you are doing. This constitutes cheating and your submission will be reported to the College of Engineering for academic dishonesty. Warning: we are ***very good*** at detecting cheating.

Relax

Really, this is a fun course. I do my best to be entertaining, and not overly boring. You'll find me easy to communicate with, and actively involved! The information we cover is *fundamental* to your education in computer science, and is actually pretty interesting. :)

If you're not having fun, you're doing it wrong!