Success Guide for COMP-1410

Fall 2022

Introduction

Welcome to COMP-1410, the second programming course offered by the School of Computer Science at the University of Windsor. This will be your first dive into the real "heart and soul" of computer science. More than just a second course in programming, in this course we start putting the ***science*** into computer science. For CS majors, the material in this course will be foundational to your future courses—in order to succeed in your degree you will need to understand it thoroughly. For non-CS majors, being able to effectively apply the material in this course can provide you a competitive edge in nearly any career.

Students tend to find this course more difficult than COMP-1400, because more topics are introduced, they are studied in greater depth, and new ways of thinking are introduced. As a result, this is not a "bird" course. The good news is that the course is well within the ability of all university students to complete—***if*** you develop the right habits. There is no reason why you should not succeed if you study with the proper strategy. This guide was written to help put you on the proper path.

Logistics

Regardless of your past history, if you want to do well in university courses your best strategy is to develop successful habits. University is a time in your life when you—perhaps the first time—will be given the freedom to set your own habits. No one will chase after you if you don't attend the lectures or submit your coursework. It is all-to-easy to slip into the tempting habit of thinking "I can skip today and catch up later". Regardless of the truth of this, experience shows that the students that *consistently* make an effort do the best in the course.

Each part of the course is designed to help you learn the course material: the lectures and textbook explain the concepts; the labs and assignments give you an opportunity to practice and receive feedback; and the midterm and final exam allow you to demonstrate your understanding.

Lectures

Lectures introduce the primary material in the course. Attending lecture *in and of itself* will not help you much—computer science is not something you can pick up by mere listening—but this is not an excuse to skip lecture.

> Going to lectures reserves three hours of dedicated time each week during which you are thinking about course material, with the aid of someone who understands it thoroughly. It's a rare student who is disciplined enough to skip lectures and adequately substitute for that experience. Of course, not every minute of lectures is "quality time", but you won't know what you're missing if you don't attend.
>
> *Survival Guide*, University of Waterloo

Lectures in this class tend to be interactive—I ask many questions of the class throughout the term. The students that do well in the course tend to be the ones that attend lecture, week after week, participate by providing their input, and even ask their own questions about the course material.

After lectures you should spend a significant amount of time reviewing the material and practicing the concepts on your own. The amount of time is different for everyone but a good rule of thumb is to spend at least 2–4 hours outside of lecture for every hour in lecture. You should aim to understand everything on the slides and be able to write *by yourself* all the functions we implement in class.

Labs

Labs are run weekly and give you the opportunity to put into practice the material covered in lecture. Labs are run by graduate students and upper-year undergraduate students. Each week there will be a "lab assignment" to submit. The purpose of the labs is to help you understand and apply the course material. They typically can be solved by a direct application of the material from lecture, so if you can't complete the lab it is a sign you need to catch up. Lab instructors are also happy to answer your questions about the material and assist you while you complete the lab assignments.

Assignments

Assignments are approximately monthly and they also provide you the opportunity to put the material you've learned into practice. Compared to the labs, assignments will be longer, require more thought, and you may have to apply the course material in creative ways. Assignments help you solidify your understanding by testing if you can apply the material in a more general setting. A fact that may be counterintuitive is that ***the solution to the assignment is usually the least important part***. The concepts and techniques that you study and practice in the process of completing the assignment are far more important. This means that looking for solutions online is a counterproductive strategy for success in this course.

> Most of the learning in CS takes place through your working through problems on your own. In many cases the final answer isn't particularly relevant; it's going through the process of arriving at the final answer that's important. Doing the assignments is the key to doing well in the course. Skipping assignments would be like trying to learn to play a musical instrument without practicing, attempting to master a language without speaking it, or claiming to be an outstanding artist or athlete based solely on theoretical understanding of the underlying principles of the field.
>
> *Survival Guide*, University of Waterloo

A student once approached me with a concern about their coursework. They said they spent many hours on the assignment and were frustrated that they couldn't solve it. Eventually, they were dismayed to find that the solution was very short. They worried that because it took them so long to come up with something so short that they were missing something. In fact, this experience is totally normal and to be expected—nearly everyone learning this material for the first time has a similar story. When a related question was asked on the final exam, whom do you think answered it correctly, the students that spent hours wrestling with the material or the students that found the answer online?

That's not to say you should spent an unlimited amount of time working on an assignment if you aren't getting anywhere. It's better to spend an hour or two on it (and then put it aside and come back to it tomorrow) then it is to try to solve it in one sitting. This is one of the primary reasons why starting *early* on assignments is a good habit to have. You're handicapping yourself if you try to solve them in one sitting.

Midterm and Final

There are two tests in the course, a midterm exam that takes place around the 5th week of the course and a final exam. Together they form the vast majority of your grade in the course. The best way of doing well on the exams is to complete all the labs and assignments by yourself—and to read, study, and understand the lecture notes in detail. The exams are designed to test an understanding of the course material. “Surface-level” understanding is dangerous, and puts you at risk of failing. Thus, the exams serve two purposes: to measure how well you are doing in the course, and to motivate you to complete (and understand) the labs and assignments.

The labs are assignments are *crucial* for doing well on the exams. Even though they are worth a small amount of your final grade they are of oversized importance when you consider that doing well on the exams is largely a result of understanding the labs and assignments.

Grading System

The grading system for this course is unique and not commonly used at the University of Windsor. It is designed to be less “high-stakes” than the typical grading system and to allow labs and assignments to be *learning opportunities* as opposed to ways to earn marks.

The system works as follows: You can submit labs and assignments at any time prior to the deadline and you will get feedback from the markers as soon as possible. Markers will make an effort to leave feedback back within 24 hours, or you can get immediate feedback during the lab itself or during the markers’ office hours. You are allowed to resubmit as many times as you want before the deadline and your grade for the assessment will be based on your final submission. The “catch” is that every function you submit will be graded 0 or 1 (i.e., no part marks) and in order to get 1 it has to have no mistakes and meet *all* the requirements specified in the grading scheme.

This grading system allows you to make improvements to your work before the due date and therefore allows you to focus on understanding rather than assessment. It also encourages early submissions rather than waiting to work on your assignment until the final day. If you only make a submission immediately before the deadline, then you will not have enough time to get feedback and you will get 0 for any function with even a small mistake. Working on assignments early is one of the habits that the best-performing students tend to have and is well worth the the effort that it takes to develop.

Cheating

It takes a significant amount of time and effort to handle cheating cases, and some instructors may not do so. However, in the courses that I teach, ***I do take cheating seriously***—just ask the students that have previously taken courses with me and had to meet with the head of the school or the Dean of the faculty.

Cheating is submitting anything that you didn't write yourself. The majority of students are honest, but around 10–20% of students cheat consistently. Those students almost always fail the class even ignoring the penalties. In a previous term, students that submitted copied code scored an average of 15% on the final exam and failed the course even before being penalized by the Dean's office.

Everything you submit in this course must be solely written by yourself alone. Getting any part of your submission from a website or another person is grounds for being reported to the Dean's office. You may not look online for help on the course assessments or get help from your friends; if you need help the course staff will be happy to provide it.

For better or worse, code duplication is easy to detect. There are an unlimited number of ways of writing any given function, so it is obvious when two complex functions were copied from each other even if there are some differences. It is like if 15% of the games of a chess tournament were exactly the same with a few changes. Yes, some games may start the same way—but at some point legitimately played games become entirely different.

If you think you have to cheat to pass this course, ***something is wrong***. It is better to hand in an incomplete submission or nothing at all instead of using copied code. When you submit copied code it is far too easy to fool yourself into thinking that you understand something that you do not—and past experience shows that you are very likely to fail the exams. Instead of cheating, you should talk to the course staff as soon as possible so that we are aware of the situation and can do our best to help offer you guidance and support.

The most tragic part of cheating for me has nothing to do with the course or academics at all. I wish you success, and a lack of integrity will ultimately hurt how successful you become. When you start your career no one will care what grade you got in this course—but people will care greatly about your integrity. Human beings are mostly creatures of habit, and the habits you build now determine whom you will become. Put yourself on a successful path by having solid integrity now.