Academic Integrity in COSC326

The general principles on academic integrity for COSC326 are laid out in the paper's manifesto (and repeated below). The main purpose of this document is to add some illustrative examples that may help in understanding the principles – these are not intended to be definitive or complete.

It's worth noting a few points.

- In COSC326 we have to restrict collaboration in a way that would be unusual and counterproductive in the professional world. That's because we're trying to develop and assess your individual skills, and so we need to be certain that the work coming from you is coming from you as an individual. Ideally we would, at the very least, have completely different individual etudes for every student but that would not really be practical!
- While much of the general framework regarding academic integrity is common to all papers in Computer Science, there is also individual context. Do not assume that something which was, or was not, permitted in another paper is, or is not, necessarily permitted in COSC326 (and vice versa).
- Do be aware that we monitor submissions for plagiarism as they arrive. A "complete and correct" outcome for an étude is not a certificate that there may not be (now or later) a concern about academic integrity.

General principles

In a paper such as COSC326 maintaining the standards of academic integrity is vitally important. General information on the standards and procedures connected with academic integrity and academic misconduct are available at the University's pages on academic integrity and academic misconduct., but there are a few issues specific to COSC326 as well. These will also be discussed in the first Town Hall session.

For the purpose of the following, "you" means "just you" for individual études, and "your pair/group" for pair/group études.

You may not:

- Ask questions on internet forums specific to any étude.
- Search for code online that's specific to an étude (or closely related to it).
- Provide access to any documents that you've produced in working on an étude to any other student or group (with one exception, test data – see below).
- Write code or reports while discussing them with anyone else (other than an instructor for COSC326).

You may:

- Share test data with any other students in COSC326.
- Carry out general research online for the purposes of understanding a problem, finding appropriate data structures, etc.
- Discuss études with other members of the class.

There are two overriding general principles:

- Work that you submit for an étude must be your own, and any exceptions to that (e.g., if using a third-party library) must be clearly indicated.
- In case of any doubt ask permission rather than presuming that something is permitted/forbidden (if you don't ask, you should presume that it's forbidden!)

Examples

A friend asks you if it would be appropriate to use a HashMap to represent the cost of the different types of widgets.

Answering that question is perfectly fine. Having a discussion about what the alternative representations might be is also fine.

A friend sends you a block of code that includes the use of a HashMap to represent the cost of different types of widgets and says "This doesn't seem to be working, can you see what's wrong?".

Answering that question, as posed, is inappropriate. You shouldn't even look at the code. Replying along the lines of "I'm happy to help, but only in terms of a general discussion of the étude" would be fine, but if you think that there might be a specific technical point that your friend is overlooking an alternative would be "That's probably a more appropriate question to ask one of the instructors".

In the lab, a friend runs their program on some test data. It crashes with the message:

```
Exception in thread "main" java.lang.NullPointerException at
   Widget.main(Widget.java:8)
```

They have their program file open in an editor window and ask you to look at line 8. You think you spot an obvious oversight.

As presented, it's fine to point out the oversight ("If you get a value from a `HashMap` and the key isn't present, it returns `null`"). You should not suggest how to fix it.

You set up a repository for working on COSC326, including your individual études. When you start working on a group étude you add the other members of your group to the repository with full access.

This is a clear violation of the Academic Integrity guidelines. Don't do this. For safety's sake keep your individual work, pair work, and group work in completely separate repositories (or use one repository per étude). If someone else whom you've granted access to copies your files then you, as well as they, are guilty of academic misconduct.

A friend asks "Should I use Java or Python for this étude?"

Answer, "What about Haskell, or R, or C, or C++?". Alternatively, proceed to an appropriate social venue with a group of like-minded friends and have a full and frank discussion of the features and limitations of various programming languages or paradigms. Seriously, this is perfectly fine.

You carry out a search with some fairly generic terms, e.g., "depth-first search arithmetic expressions", and among the results is a public repository of a former COSC326 student which they're using as their portfolio. Based on the text you can see in the search result you realise it includes a solution to the étude you're trying to solve, or a closely-related one.

Please don't click that link. Please do bring it to the attention of the instructor. We make a serious attempt to introduce new études, vary the conditions on existing ones, and make other changes, but sometimes (particularly for really good or valuable études from the learning point of view), this sort of thing happens. If you use that code you will have committed academic misconduct and the consequences will potentially be very serious.

You're reminding yourself about how depth-first search (DFS) works and come across some pseudocode, accompanied by a remark "This can be translated directly to python, and all you need to do is change the objective function" along with a link to a blog post. You read the blog post and it includes a python framework for DFS. Should you use it?

Why not? Absolutely no problem, provided you attribute it in your submission. Do be aware though that lots of our études have a few wrinkles or twists and while, for instance, it may basically be a DFS that's needed there could be some local changes required – it'll be up to you to find them of course.

A friend asks "I'm having trouble understanding the widgets étude. Can you please send me your code for . . . "

Stop there. Don't even look any further in the email, text message or whatever. No.

You're stressed and under pressure, or have been ill, and are really having difficulty coming to grips with the widgets étude. You start a message to a friend "I'm really under pressure, can you please send me your code so that I can see how it works?"

Don't. Think of the uncomfortable situation you're putting them into if you send that message. The instructors are there to help - we understand that circumstances can intervene and make things difficult. Contact us instead.