

Integrity in My Computer Science Courses

Tom Kelliher

The rules below are my attempt at applying Goucher College's Academic Honor Code to my computer science courses. I have no desire to create a climate in which students feel as if they are under suspicion. The entire point of the Honor Code is that we all benefit from working in an atmosphere of mutual trust. In addition, I want to maximize your opportunities for learning and to ensure that the work that I'm assessing is your work, as assigned.

Collaboration

Paragraph II.B.1 of the Honor Code states

“Students may not collaborate with another student or any other individual(s) unless authorized by the instructor.”

Paragraph II.B.2 of the Honor Code states in part

“Unauthorized use of ChatGPT or other Generative Pre-Trained artificial intelligence tools to write papers, essays, reflections, or other assignments or exams/tests is strictly prohibited.”

The following three rules define unauthorized collaboration in my classes. Any other type of collaboration is therefore authorized.

Rule 1: You must not look at solutions to, or program code for, an assigned problem, lab, or project that are not your own.

It is an act of plagiarism to submit work that is copied or derived from the work of others and submitted as your own. For example, using a solution from the Internet or a solution from another student (past or present) or some other source, in part or in whole, that is not your own work is a violation of the Honor Code. The best way to steer clear of this possibility is not to search for online solutions to programming assignments. Looking at someone else's solution code in order to determine how to solve the problem yourself is also an infraction of the Honor Code.

Rule 2: You must not share your solution code with other students.

In particular, you must not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it. Similarly, you must not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in the same code. Moreover, you should take reasonable measures to maintain the privacy of your solutions. For example, you should not leave copies of your work on public computers nor post your solution code on a public website.

Rule 3: The use of ChatGPT or other Generative Pre-Trained artificial intelligence tools to write papers, essays, reflections, or other assignments or exams/tests is strictly prohibited.

In computer science courses, it is usually appropriate to talk to others — the instructor or other students — about problem-solving strategies, general program structure, and debugging tips. In fact, I strongly encourage you to seek such assistance when you need it. Discuss ideas together, but do the coding on your own.

What it means to build upon other's work — if your currently assigned project is to write a shell program it's fine to look at documentation for system calls and short code examples of system call usage. System calls are the foundation of a shell program and there's a lot of learning to be had by piecing those system calls together into your working shell program. That's a proper way to build upon the work of others. On the other hand, if you're writing a shell program and you look at other shell programs, there's nothing left to be learned. The only learning that you've demonstrated is the ability to locate shell programs. You're not building on the work of others, so it's inappropriate collaboration.

In a nutshell, discussing general ideas together, but coding on your own, ensures your learning, which is why you're here and is what I am assessing.

Plagiarism

Paragraph II.B.3 of the Academic Honor Code states in part

“It is expected that students will discriminate carefully between parts of their writing for which they are solely responsible and other parts for which credit is owed to someone else.”

Rule 4: You must indicate on your submission any assistance you received.

If you received aid while producing your solution, you must indicate from whom you got help (if that person is not the instructor) and what help you received. A proper citation must specifically identify the source (e.g., person's name, book title, website URL, etc.) and a clear indication of how this assistance influenced your work (be as specific as possible). This citation must appear in some prominent place. For example, in an email to me or a README file submitted with the assignment. As an example of a citation, you might write “I discussed the approach used for sorting numbers in the `sortNumbers` method with Mary Smith.” If you make use of such assistance without giving proper credit, you may be guilty of plagiarism.

It is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary into having someone else write the actual code or show you their code. It is fine to discuss ideas and strategies, but you must be careful to write your programs on your own, as indicated in Rules 1 and 2.

Acknowledgments

This policy is derived from, and uses much of the language from, the Honor Code of the Stanford University Computer Science Department, <http://csmajor.stanford.edu/HonorCode.shtml>, and the Purdue Online Writing Lab, <https://owl.english.purdue.edu/owl/resource/589/01/>.