

# Collaboration Policy

*Fall 2017*

## 1 Introduction

In order to help the course staff evaluate each student in CS 33 as fairly and individually as possible, we have written a collaboration policy, which we expect all students to abide by. **Please read this policy carefully**, as it may differ from collaboration policies in CS classes you have taken previously. Please complete the [digital collaboration policy quiz](#) and agreement form by the end of the first lab. You will not be able to receive credit on assignments until you complete the form.

We strongly encourage teaching and learning with your peers. At the same time, your work must, in the end, represent your own understanding of the material. In that spirit, our overall policy is that you can talk about almost anything and work out solutions together, but your final handin must be your own work. Debugging is one area where we particularly encourage you to help one another. See the Testing and Debugging section for more information.

You may (and are encouraged to) talk to other students about the content of the lectures and of the textbook, and about high-level concepts in general, but please ensure that in doing so you are not violating any of the following policies.

## 2 Permissions

You must ensure that your course documents have permissions which do not permit other students to view them. Ask a consultant or a TA for help with permissions if necessary, or use `man chmod`. Failure to do this may be seen as a violation on your part. Do not post your code in publicly viewable repositories, such as github. Doing so is a violation of Brown's academic code that could result in your getting an NC in the course, even if the violation occurs after you've completed the course.

## 3 Projects

When completing projects, you are allowed to discuss how to approach an assignment with other CS 33 students. However, you should never sit down to discuss an assignment with someone else before you've analyzed the problem in depth on your own. In addition, we ask that you write up your code independently and separately from any joint discussions, and that you not take away any notes from such discussions (i.e. erase the whiteboard after discussion). When you are coding the assignment, you may not code along with another student. You may discuss your code or allow others to look at your code only after you have moved on to

debugging. While you may search the internet for documentation, such as looking up C syntax or how certain instructions work, you may not search the internet for solutions to problems. Copying code from the internet is a clear violation of the academic code. Additionally, **do not post code on Piazza publicly**. Lastly, **do not email the TAs your code**. Email is reserved for non-assignment-related questions.

## 4 Testing and Debugging

Testing ideas and a testing plan are part of project design, and you may collaborate to come up with them. Test code, however, must not be shared. Saying something like, “Oh, have you tried doing an add after a jump? I had a strange problem with that.” is fine, but “Try compiling this chunk of assembly code” or “Load this into your register” are not.

You may not look at or copy anyone else's code for the purpose of helping you write your own, and you may not permit anyone to look at or copy your code for that purpose. **You are permitted to work together on debugging**, and may look at someone's code or work together to modify code for the purpose of helping him/her debug, *as long as your code is not open concurrently*.

Note that this policy is significantly more lenient than that of most CS courses. You all have different skills when it comes to problem solving, so we encourage you to help each other. You can often learn just as much from your peers as you can from TAs.

**If you receive help on debugging, you must cite this help in your README:**

*ex: "I received debugging assistance from dj and dcorrea"*

## 5 Labs

Collaboration during lab periods is allowed, but each student must still turn in their own solution to the assignment. Discussion and debugging are permitted, but pair programming is not. Please note that, especially if you get checked off after your lab section is over, we reserve the right to do a “wire-pull test” (i.e., ask you to explain your program) and to use all of the tools at our disposal to compare your code to that of other students. This also applies to all other assignments and work for the course.

## 6 MOSS

Throughout the semester, we will be using a program called MOSS (Measure Of Software Similarity) to detect potential illicit collaboration. After each project, we will run MOSS, which compares each assignment handin pairwise against every other handin and reports ‘matching’ or highly similar code. Note that MOSS will not count code that is very similar across a high

number of students as matching - we recognize that there are portions of projects that everyone in the class will do in a similar manner, and what we care about are the ways in which pairs or small groups of students have highly similar code that differs from what the majority of the class did. We will examine the results by hand and look for cheating.

This year, we will be reporting your *highest similarity score with another student* on your grade rubric for each project, i.e., the highest percentage of your code that matches the code of another specific student in the class. We will also announce, in an email to the class or a Piazza post, the percentage similarity threshold above which we suspect unauthorized collaboration (different for each project and determined by hand by Tom and the HTAs). If your similarity score is above this threshold, we are likely investigating your project and may take action against you by writing up a case, though this may not happen immediately. We strongly suggest that if your score is above the threshold, you get in touch with Tom and explain what happened.

Late handins will be MOSS'd, but you may not be given similarity score information on your grade report.