1 Preface

The goal of CS15 is to provide an introduction to object-oriented programming and computer science in a way that is accessible, fun, and in-depth. Every year, we make changes to the course based on student and TA input with the aim of better accomplishing this goal, which always means looking at ways to build community, cut down on lines at TA Hours, and reduce student stress. While we implemented many new strategies this year to help with these objectives, including loosening the collaboration policy from last year, the extensive lines at hours and feedback on the midterm survey have caused us to rethink this even further. After discussing extensively amongst the course staff, we’ve decided to change the collaboration policy further starting with Tetris.

2 Policy

Starting with Tetris, you will be able to collaborate on the mini-assignment for each project and all material covered in the mini-assignment over the course of the entire project, to the extent of the mini-assignment. Thus, if the mini-assignment asks for a containment diagram and two algorithms in pseudocode, you may collaborate on what the containment diagram will look like, and pseudocode (though not actual Java code) for those algorithms. You cannot discuss portions of the assignment not covered in the mini-assignment, nor can you go further than the scope of the mini-assignment questions— if the question requires pseudocode, then you may discuss pseudocode, but you may not discuss how to write it in Java code, i.e., how to translate the pseudocode to Java. Note that this policy does not allow collaboration on everything discussed in design discussions; rather, it allows collaboration only on the questions on the mini-assignment (both written questions and “Questions to Think About”).

At no point can you look at another student’s Java code, and all collaboration on mini-assignment questions and concepts has to be done without any code on screen – essentially “laptops should be closed” when discussing the mini-assignment questions and concepts. That being said, you are allowed to take away notes from your discussions with other students. Your notes should not go beyond the scope of the mini-assignment; as a guideline you should be comfortable showing your notes to a TA.

Additionally, you may discuss the specifications of the project. That is, a question like “Should my square piece be able to rotate?” for Tetris is entirely acceptable; a question like “How do I prevent my square piece from rotating?” is not, unless that concept was covered in the mini-assignment.

Collaboration at the pseudocode and design level within the confines of this policy should still produce actual Java code that does not look identical to that a student with whom you collaborated under the above restrictions. For instance, in higher-level CS courses,
pseudocode is given in slides for 5-10 line algorithms that are on later coding assignments, and yet 200 students in a course can and do still turn in 200 distinct sets of code for those algorithms.

**All written work and code must still be your own**, and all code will still be run through MOSS, a program measuring similarity in code, to help identify potential cases of violation, each of which is then checked manually.

In keeping with the department’s standard procedure, **we are requiring that you put down the logins of any other students that you worked with on the project within the confines of the policy**. This is not to trick you, but to help us understand, if code similarity occurred, what may have occurred because of acceptable collaboration—thus, putting down logins of students that you worked with can only help you. **If you do not put down logins of other students, you are stating that you did not discuss the project with anyone, so any potential collaboration violations found will be evaluated under the original collaboration policy**. In order to utilize the extra mini-assignment level collaboration, you must include the logins of the students that you worked with.

When collaborating under this new policy, we urge you to think critically about how you are communicating with the person you collaborate with. Especially during introductory courses, collaboration can be abused in ways that impede learning. For example, simply giving a friend the pseudocode for an algorithm will not help them learn how to think about and debug algorithms in the future. If you are helping each other on pseudocode, make sure you are having a thoughtful discussion about it, as opposed to one person telling the other how to do it.

In order to receive a grade on the remaining projects, you must agree to abide by this policy by filling out the updated collaboration policy form.