Collaboration Policy and Registration Form

Introduction to Computer Graphics, Fall 2019

1 Collaboration Policy

Since CS123 does not have any examinations, your grade is based solely on your homework and programs. In order to evaluate you in the course, we must be sure that your assignments are your own work. This policy adheres to Brown's academic code: *"Academic achievement is evaluated on the basis of work that a student produces independently. A student who obtains credit for work, words, or ideas that are not the products of his or her own effort is dishonest and in violation of Brown's Academic Code."*

For every assignment you should do your own thinking, designing, coding, and debugging. Under no circumstances should you let yourself be led by another student or receive an amount of help which makes an assignment easier to implement. Conversely, you should never assist another student in a manner that would provide any details as to how an assignment can or should be implemented. **Just "helping a friend out" is okay only if the help does not breach this collaboration contract. Otherwise, it is considered as serious a violation as cheating and will be considered a breach of the collaboration policy.**

We strictly enforce this policy. Every submitted solution will be personally inspected by TAs and run through sophisticated code similarity software (MOSS) specifically designed to catch instances of cheating. All violations of this collaboration policy will be discovered and brought forth to Brown's academic counsel.

1.1 Permitted Collaboration

The following forms of collaboration are allowed (even encouraged) and are not considered collaboration policy violations.

1. **A team final project.**
   Details will be provided when final projects are discussed in November.
2. **Discussing solutions to the "algorithm" assignments before each project.**
   If you choose to discuss the algorithm assignments with another student you must **erase/throw out** all notes from the discussion before writing up the solution on your own, and you must **write the logins of the student(s) you collaborated with** next to each problem. Your handin should not look identical to another student's and should only contain your own work. However, keep in mind that giving/receiving a significant amount of help on the "algorithm" assignments will be considered a violation.
3. **Discussing solutions to labs.**
   Labs are meant to give you hands-on experience with material covered in class, and you are encouraged to work together with others and help each other debug them.
4. **Discussing material covered in lectures or the textbook.**
   Example: "What are the different transformation matrices that were covered in lecture last week?"
5. **Discussing the requirements of an assignment.**
6. **Asking general knowledge questions about syntax and C++.**
   Examples: "How do I make something public? How do I set up debugging in Qt Creator?"
7. **Discussing general techniques of designing, coding, or debugging.**
   Example: "When I get a segmentation fault, can you show me how to use gdb to find the line where the program crashed?"
8. **Exchanging render results, scene files, and other test data.**
   Example: "I exchanged {sample renders, scene files} with lcohen2."

There is no penalty for permissible collaboration with another student, as long as you clearly document the students with whom you collaborate, as noted above.

1.2 Prohibited collaboration

The following things are **not** allowed under any circumstances.

1. **Discussing project design or code.**

Projects are meant to be done entirely on your own, with the sole exception of the final project.

a. **Copying project code or test cases.**
   You should not be writing down anyone else's project code (including code from the internet), or allowing anyone else to write down your code. Remember, we have software designed explicitly to look for undue similarity of code and we check all submitted programs with it.
b. **Discussing project pseudocode.**
   Pseudocode in CS123 is close to discussing the code itself. If you've gotten beyond discussing the solutions to the algo, you're ready to work on your own.
c. **Discussing project code.**
   Project code must be written entirely on your own. Even discussing support code can lead into implementation details, so absolutely no discussion is allowed.

2. **Debugging a project with another person.**
   Sitting at the same computer with someone else and trying to fix a bug is not allowed. Describing your problem to someone and asking for advice on debugging techniques (e.g. "How can I debug segfaults using Qt Creator?") is okay as long as the advice is **completely unrelated** to the project.
3. **Asking for help on something you haven't thought about yourself.**
   Always make every attempt to tackle a problem yourself before asking another student or a TA. It will help you become a better programmer, as well as a better student.
4. **Having incorrect file permissions/being careless with your source files.**
   We require that all students maintain appropriate permissions on their coursework. Other students should not be able to access, view, or copy your files. If you don't know how to do this, ask a Sun Lab consultant or see a TA. **This also includes having your code publicly available on GitHub or other online code repositories. If another student copies your work, you will be held accountable for negligence and be referred to Brown's Academic Code Committee.**
5. **Using previously published solutions or otherwise found course material.**
   Students retaking the course may have access to course materials and solutions published the year prior. You may not use any assignment materials (solutions or otherwise) that were not published on the course website before the assignment was due. Students may not consult their own solutions from the previous year when completing the current year's work.

On a general level, you are not allowed to let yourself be led by another person to the extent that your task becomes significantly less challenging because of your discussion with that person. Always be careful not to venture into design and implementation specifics.

In addition, don't sit down with someone else before you've analyzed the problem in-depth on your own. You should do your own problem solving, do your own program design and decomposition, and design your own algorithms and data structures. If you are discussing what algorithms or functions you wrote to solve the problem, or describing header files or specific lines of code, then you are breaking the Collaboration Policy.

1.3 Conclusion

We believe that this policy is explicit enough to guide your judgment and that we have not left you much gray area. If you are ever in doubt about the legality of your actions, be sure to clear them with a TA, even after the event has already occurred. When we confront a student with a case of suspected violation, an answer of "I didn't know that this is wrong" will not let you off the hook. Brown's academic code is very specific on this. Suspected cases of disallowed collaboration will be referred to the Academic Code Committee and typically result in a "directed NC with transcript notation" and parental notification for the first offense, suspension on a second.

Again, note that you are expected always to initially approach a problem on your own, and put in serious effort to find a solution. You are honor-bound to preserve your independence of thought. Remember that the TAs and lecture slides should always be your first resource when you have a question or problem.

2 Registration

By signing below you are bound to these terms and will be officially registered for CS123. You will be added to the CS123 student group (and, if necessary, your account will be created) before your first lab section.

Name: _____ Date: _____

Signature: _____ CS login: _____