Collaboration Policy

This policy is specific to CS1680. Please read it, as it may differ significantly from other courses you have taken.

By submitting any assignment in this course, you acknowledge that you have read and agree to this policy. If you have questions about the policy, please consult the course staff.

1 Introduction

Our goal is to help you learn gain experience building networking systems and learn fundamental principles and techniques for navigating the complexity of other systems you will see outside this course. Collaboration is an important part of building networks and systems, as well as the learning process. At the same time, by the time you leave this course, we want to make sure you have internalized the material yourself—we want all of you to have worked on every project and homework and have encountered the intellectual challenges it poses.

Therefore, we have adopted a policy that generally encourages teamwork while establishing a few boundaries to help make sure you build your own understanding.

If you have questions about this policy, please ask the instructors or TAs. Violations of this policy may by considered violations of Brown's Academic Code policies and will be handled accordingly..

2 Projects

Our first project, Snowcast, is submitted individually. Other projects are submitted in teams of two students.

For the remaining projects, students on a project team are expected to collaborate freely. We highly recommend collaborating via pair programming—where both team members write code together, rather than a "divide and conquer" strategy—since you will be spending a lot of time thinking about system design. We will provide more guidance on this as we start each project.

For individual projects, or in terms of collaboration between teams, you are encouraged to discuss project topics with other students, so long as you are not directly revealing specific solutions to a problem. You may work out high-level solutions together at a conceptual level, or look at parts of others' code to help them debug minor errors or environment setup issues. However, as you collaborate with others, you must abide by the following rules:

1. **Any code you submit must be your own work (ie, written by you)**. You are responsible for understanding all code that you submit. Typing out a line of code verbatim that you see on a

whiteboard, or from an online source, when you don't know what it does is not acceptable, as this impedes your own learning process.

2. **You must not directly transmit code to another student, or permit someone else to use your code directly**. To help ensure everyone follows rule #1, you should not send code, or screenshots/photos of your code, to another student via any medium (email, text, Snapchat, Tiktok, `netcat`, ...), or allow someone else to take a photo of your computer screen. Taking paper notes or whiteboard photos from a group discussion (or during TA hours) is fine, but you should not be copying any notes/photos directly into your code. If you find yourself wanting to rely on this, you should consult with the TAs or the instructors to check your understanding of the concepts involved—we can work with you to help build any skills you may be missing.

3. **When turning in your work, you must list anyone you collaborated with and what you worked on together**. Your final submission should include a file called `COLLABORATORS.{txt,md,}`. In this file, you must list the names and CS logins of anyone you worked with and include a brief description of what part(s) of the project you worked on together. Collaborating with someone on part of an assignment is not an excuse for not understanding it. We reserve the right to ask you to discuss your solutions with us if we are concerned about your work.

4. **When collaborating, your discussions should generally take one of two forms**:

   (a) **"High-level" discussions about course concepts or how to approach problems**: we encourage you to work together to figure out how to get started and develop a strategy for solving a problem. After that, you are responsible for building your own implementation and understanding how it works.
   Note: Certain projects may restrict how you can discuss your approach to problems with other students, particularly where the assignment goal is to find and exploit vulnerabilities in project code—more details will be provided where applicable.

   (b) **"Low-level" questions about programming languages, tools/environment issues, debugging techniques, dealing with errors, Linux questions, etc.**: Working in security involves using a *lot* of different tools—we don't expect you to know every detail of how they work, and we encourage you to work with your peers (and course staff!) to learn how to use them! It's always okay to ask someone about specific details of a tool, language, or library you're using (eg. *"What does this error mean?"*, *"How do I do X in Go?"*). In general, for any non-assignment-specific issue you might search for online, it's also okay to ask someone else about it, too.
   For low-level issues like these, it's okay to look at someone else's code to help figure out a specific problem, or build a strategy for debugging it, so long as this does not i) directly reveal an exploit or solution, or ii) involve copying code without understanding it.

3 Homeworks

Similar to projects, you are encouraged to discuss any aspect of the homework problems with other students, so long as:

1. **The solutions you write down must be your own work (ie, written by you)**. You are responsible for understanding everything you submit—it is not sufficient to copy a solution from a whiteboard, group discussion, or collaborative hours, without understanding it. As a general guideline, it's okay to work on problems with someone, but the writeup you generate should be your own (even if you're typing separately at the same table)–if you find yourself copying words or phrases verbatim, you should stop and make sure you are understanding the problem well enough to express it in your own words.
2. **When turning in your work, you must list your collaborators**. In your writeup for each problem, list the names and CS logins of anyone you worked with and include a brief description of what you worked on together. Once again, collaborating with someone on part of an assignment is not an excuse for not understanding it. We reserve the right to ask you to discuss your solutions with us if we are concerned about your work.

4 EdStem Discussions

We use EdStem (aka, Ed) as a platform for discussion and asking questions. As a "public" (within the course) discussion forum, Ed can be an excellent resource for providing clarifications about various course topics and assignment mechanics.

Public posts We encourage you to ask *public* clarifying questions on Ed, so long as your questions to not reveal details about assignment solutions. For example, it's okay to ask about clarifications on assignment mechanics, high-level questions, help understanding error messages or debugging strategies, or questions about any course concepts in lectures.

In general, small amounts of code can be included in public posts if it's generic enough not to reveal solutions for an assignment. For help debugging specific errors, it's often helpful (both to you, and anyone reading the post!) to create a generic example that demonstrates a problem (similar to posts you see on StackOverflow).

If you're ever unsure if it's okay to make a public post, please post privately. If we believe your post would benefit the class as a whole, we will ask if we can make it public (as an anonymous post).

Private posts Can be used for questions that might reveal your solutions, or specific issues related to grading or logistics for review by the instructors and HTAs.

Please do not request extensions or request accommodations via Ed—this information is considered sensitive and should only be sent to the instructor.

5 Protecting your work

You must ensure that your solutions will not be visible to other students. All assignment code is distributed using Github Classroom, which automatically gives you a private repository you can use for your work.

If you store your work elsewhere, such as a private version control repository, you must ensure your account is configured so your solutions are not publicly visible. (For example, GitHub, offers free private repositories.) Leaving course projects in a place where they are visible to other students (such as in world-readable folders on the department filesystem or public GitHub repositories) constitutes a *Collaboration Policy* violation, and can result in penalties even if they are discovered after the course.

6 Responsible network security practices

As we learn about various networking protocols, we will often study security issues and ways to break things, which provides a good case study for understanding how protocols work and how they've evolved over time.

Students are expected to use the techniques they learn in this class in a responsible manner. Some of the techniques covered in this course for educational purposes are unethical and/or illegal to use and apply in contexts beyond the course itself. Breaking into, misusing, or harming computer systems or networks is illegal and punishable by law if done without the explicit authorization of the owner. Attacking technical systems (whether owned by Brown or by others), except as specifically assigned in this course, is a violation of Brown's Acceptable Use of IT Resources Policy and may lead to disciplinary action. The consequences of violating this policy can be severe—similar to, or greater than, a violation of the Academic Code.

If you have any questions about what kind of conduct is legal and/or ethical, please contact the instructor and/or the TA staff first.