Collaboration and Sharing

In line with MIT's policy on Academic Integrity (https://integrity.mit.edu/handbook/writing-code) , here are our expectations regarding collaboration and sharing of work.

Collaboration

We encourage you to help each other with work in this class, but there are limits to what you can do, to ensure that everybody has a good individual learning experience. This section describes those limits.

Problem sets

Problem sets are intended to be primarily individual efforts. You are encouraged to discuss approaches with other students but your code and your write-up must be your own.

You may not use materials produced as course work by other students, whether in this term or previous terms, nor may you provide work for other students to use.

During code review, you will see classmates' solutions to a problem set. While it is fine to take inspiration from their approach, do not copy their work.

It's fine to use material from external sources like StackOverflow, but only with proper attribution (https://integrity.mit.edu/handbook/writing-code) , and only if the assignment allows it. In particular, if the assignment says "implement *X* ," then you must create your own *X* , not reuse one from an external source.

It's also fine to use any code provided by this semester's 6.005 staff (in class, readings, or problem sets), without need for attribution. Staff-provided code may not be publicly shared without permission, however, as discussed later in this document.

Examples

1. Alyssa and Ben sit next to each other with their laptops while working on a problem set. They talk in general terms about different approaches to doing the problem set. They draw diagrams on the whiteboard. When Alyssa discovers a useful class in the Java library, she mentions it to Ben. When Ben finds a StackOverflow answer that helps, he sends the URL to Alyssa. **OK.**
    - As they type lines of code, they speak the code aloud to the other person, to make sure both people have the right code. **INAPPROPRIATE.**
    - In a tricky part of the problem set, Alyssa and Ben look at each other's screens and compare them so that they can get their code right. **INAPPROPRIATE.**
2. Alyssa and Ben haven't been working together on the problem set so far, but Ben is now struggling with a nasty bug. Alyssa sits next to him, looks at his code, and helps him debug. **OK.**

    - Alyssa opens her own laptop, finds her solution to the problem set, and refers to it while she's helping Ben correct his code. **INAPPROPRIATE.**
3. Louis had three problem sets and two quizzes this week, was away from campus for several days for a track meet, and then got sick. He's already taken two slack days on the deadline and has made almost no progress on the problem set. Ben feels sorry for Louis and wants to help, so he sits down with Louis and talks with him about how to do the problem set while Louis is working on it. Ben already handed in his own solution, but he doesn't open his own laptop to look at it while he's helping Louis. **OK.**
    - Ben opens his laptop and reads his own code while he's helping Louis. **INAPPROPRIATE.**
    - Ben has by now spent a couple hours with Louis, and Louis still needs help, but Ben really needs to get back to his own work. He puts his code in a Dropbox and shares it with Louis, after Louis promises only to look at it when he really has to. **INAPPROPRIATE.**
4. John and Ellen both worked on their problem sets separately. They exchange their test cases with each other to check their work. **INAPPROPRIATE.** Test cases are part of the material for the problem set, and part of the learning experience of the course. You are copying if you use somebody else's test cases, even if temporarily.

Note that in the examples marked inappropriate above, *both* people are held responsible for the violation in academic honesty. Copying work, or knowingly making work available for copying, in contravention of this policy is a serious offense that may incur reduced grades, failing the course and disciplinary action.

Group projects

You should collaborate with your partners on all aspects of group project work, and each of you is expected to contribute a roughly equal share to design and implementation.

You may reuse designs, ideas and code from your own work earlier in the semester (even if it was done in a group project with a different partner). You may also use any code provided by this semester's 6.005 staff.

You may also use material from external sources, so long as: (1) the material is available to all students in the class; (2) you give proper attribution (https://integrity.mit.edu/handbook/writing-code) ; and (3) the assignment itself allows it. In particular, if the assignment says "implement *X* ," then you must create your own *X* , not reuse someone else's. Finally, your group may not reuse designs, ideas, or code created by another group, in this semester or previous semesters.

Public sharing of work

People often want to share their code publicly, e.g., on Github, in order to show off a portfolio of code they've written to potential employers. Building a portfolio is a great idea, but 6.005 is *not* a good class to use for it, because the problem sets and projects are fixed by the course staff, not chosen by you. Personal projects, hackathons, and IAP contests are much better ways to build up your portfolio.

The policy for public sharing of 6.005 code is described below.

Problem sets

Copyright for the starter problem set code is held by the 6.005 course staff, and does not allow redistribution of derived works without prior permission. Your solutions are a derived work, so you may not distribute your problem set solutions publicly. This means you cannot post them on Github, in a public Dropbox folder, or on a public server accessible to others.

Keep in mind that when work on an individual problem set is copied, both the provider and the consumer of copied materials are violating academic honesty standards, as described above.

Group projects

For group projects, before you can post any code for public distribution, all **authors** of the code must **agree** to public distribution, and their authorship must be **acknowledged** .

- **Authors** means anyone who worked on the project at any time, including previous phases of the project. Suppose Alice wants to post her project code. Alice worked on project 1 with Bob and Carol, and then she uses code from project 1 during project 2 with Yolanda and Zach. Yolanda and Zach also use code from their own project 1, when they worked with Morris, Nancy, Ollie, and Patricia. Alice, Bob, Carol, Morris, Nancy, Ollie, Patricia, Yolanda, and Zach are *all* authors of Alice's project 2.

  Even if Alice only wants to post one file from the project, *even if she wrote that code herself* , she must ask all of her teammates from all phases of the project, and all of their teammates from previous phases of the project. Projects are a team effort, and code ownership is shared.

- **Agree** means all authors must be asked whether public distribution is OK, and they must answer yes. In our example, if Alice wants to post any part of project 2, then Bob, Carol, Morris, Nancy, Ollie, Patricia, Yolanda, and Zach must *all* be asked and answer yes.

- **Acknowledged** means the names of all authors appear either in a comment at the top of each source file, or in a `README.md` (or other prominent README file) at the root of the project. If you're posting on a site like GitHub, keep the commit history – and for projects that use code from a previous group, import that code with its commit history. This history demonstrates each person's individual contribution, and shows off your software development process.

These rules are derived from the standard practice for open-source code projects, in which it's necessary to clarify the origin of all the code and obtain Contributor License Agreements (//en.wikipedia.org/wiki/Contributor_License_Agreement) from all identified contributors. You should follow this standard ethical practice of the software development community even for your own projects. If you want to publish code from a hackathon where you worked with other people, then make sure all the authors agree and are appropriately acknowledged.

MIT EECS