

Improper Collaboration Policy

Prepared by Dr. Derek Harter, Dept. of Computer Sciences
Texas A&M University- Commerce

Description of Improper Collaboration

In grappling with course work, the sharing of ideas is educationally useful. The copying of ideas is destructive, fraudulent, and unacceptable.

YOU ARE STRICTLY FORBIDDEN TO COPY SOMEONE ELSE'S HOMEWORK OR PROGRAMMING ASSIGNMENTS, WHOLE OR IN PART AND SUBMIT THOSE ASSIGNMENTS IN YOUR NAME. YOU ARE STRICTLY FORBIDDEN TO MAKE TRIVIAL OR MECHANICAL CHANGES TO SOMEONE ELSE'S ASSIGNMENT SOLUTION AND SUBMIT THE TRANSFORMED VERSION IN YOUR NAME.

It is difficult to know where to draw the line between educationally useful sharing of ideas and the educationally destructive copying of ideas. I will quote Roger D. Eastman of Loyola College (ironically this is copied), who draws the line rather well for programming assignments:

“I encourage you to help each other with programming assignments, but I also want you to understand where the help should stop. Don't take someone else's program to copy, or give yours for copying. If you want to show someone your program to ask about a syntax or run-time error, that's fine; if you want to brainstorm about what the assignment requires and how to approach it, that's fine; if you want to share your knowledge of [C++], that's fine; but letting someone copy your program line by line, in fact or spirit, is not fine.”

The line between sharing and copying may still seem fuzzy. To put it another way, when you start writing code, it is time to stop sharing ideas. But what about when you are stuck with a syntax error? In that case it is all right to ask someone about the syntax error, but that is as far as your sharing of information should go. To put it still another way, it is wrong to show someone else a copy of your program for “reference.” Conversely, it is equally wrong to look at someone else's program for “reference.” A completed program should have a single easily and uniquely identifiable author.

Debugging code presents an even greater challenge. Typically, students who have completed a programming assignment become obvious targets for programming help. While we applaud the desire of students to help one another in the educational process, don't go too far. Once you look at someone's else's code, you are in danger of improperly collaborating. It is okay to tell the other person what is wrong, but do not offer a correct coding sequence. In addition, do not incorporate any of the code you may have seen into your solution. Above all, do not assume that helping someone fix their code means making it look like yours!

The normal procedure for cases of suspected copying or improper collaboration will be to notify you on the first instance that we suspect you have submitted copied materials. You will be given a chance to respond, and will receive a 0 grade for the first such instance of improper copying. If a subsequent instance of copying is detected, you will receive an F grade for the entire course.

THERE MUST NOT BE ANY COPYING OF ASSIGNMENTS.

Example Scenarios

Some scenarios are presented below as examples. These are meant to be general examples, but may not be inclusive of all possible situations.

- A group of students sit down for an hour to discuss how they each plan to attack the programming assignment. As long as specific code sequences and/or flow charts are not exchanged, this is acceptable collaboration.
- Two (or more) students each complete a programming assignment. When it is done, they share their source files, and each student imports the best code and ideas from the other file. This is improper collaboration. Submit your own work.
- Your friend tells you that he or she is having problems getting a particular function to work and asks to see your version of just that function. Even though this one function may only be a small fraction of the entire assignment, this is still improper collaboration since it is code sharing.
- You find a discarded printout, a floppy disk, or a file left on the hard drive or network in a computer lab and decide to use some of the code on it. This is improper collaboration. Even if the source of the code is anonymous, it is still tainted.
- You use a web search engine to find existing solutions to the problem. Re-using (parts of) those solutions as your own is improper. You must always give proper credit to the original author.
- The instructor provides sample code available to the class as part of the regular lectures or perhaps via a web page. Unless there were explicit instructions not to use the code, it is acceptable to include that as part of your solution. However, you should always give proper attribution to the original author.

```
/* subroutine doit()
   Code supplied as a class example
      by Professor J. Knowitall
      in CSci 123, Fall 2009
*/
```

More obvious examples of improper collaboration, i.e., blatant cheating, include submitting someone else's source code after modifying it by

- Adding, changing, and/or removing comments.
- Changing the names of variables, constants, subroutines, and/or functions.
- Rearranging the order of the declarations, subroutines, and/or functions.
- Applying any combination of the above and/or other simple transformations.

About this Document ...

This document has been adapted from a similar collaboration policy statement of Stephen Corbesero from the Computer Science Department at Lehigh University.