CIS 27 – Data Structures and Algorithms
Laney College – Spring 2013

Syllabus Addendum

The following are excerpts from course information given in recent CS courses at University of California at Berkeley:

Policy on Collaboration and Cheating

Cheating on a homework, lab, or project will earn you the maximum negative grade on that assignment. For example, if you cheat on a project worth 20 points, your grade on that project will be –20. Cheating on an exam, or cheating twice in any way, will earn you an F in the course. I reserve the right to assign an F in the course to anyone who cheats on a project, though I might not exercise it. All incidents of cheating will be reported to the Office of Student Conduct, who will maintain records of your academic misconduct throughout your undergraduate career.

We encourage you to help each other learn the material by discussing the work *before* you do each assignment. Explaining the meaning of a question or offering advice on what a compiler error message means are interactions that we encourage. On the other hand, you should **never** have another student's solution or code in your possession, either electronically or on paper. (We will call this the **No Code Rule.**) If you are not sure whether a particular interaction is appropriate, talk to your TA or the instructor.

If you receive a significant idea from someone in the class, explicitly acknowledge that person in your solution. Not only is this a good scholarly conduct, it also protects you from accusations of theft of your colleagues' ideas.

Presenting another person's work as your own constitutes cheating, whether that person is a friend, an unknown student in this or another class, or an anonymous programmer on the web who happens to have solved the problem you've been asked to solve. Everything you turn in must be your own doing, and it is your responsibility to make it clear to the graders that it really is your own work. The following activities are specifically forbidden in all graded course work:

> Possession (or theft) of another student's solution or partial solution in any form (electronic, handwritten, or printed).
>
> Giving a solution or partial solution to another student, even with the explicit understanding that it will not be copied.
>
> Working together (with someone other than your partner for the assignment) to develop a single solution and then turning in copies (or modified versions) of that solution under multiple names.
>
> …

You will do some of the projects in teams of two or three students. Any assignment that is not designated as a team assignment must be done individually. On team assignments, you share everything with your teammates, but the rules for individuals given above apply to teams. You may not work with another team or share solutions between teams. Each individual in a team is responsible for the entire project, which means that you will be held responsible if your partner uses another team's solution to produce part of your team's solution. Once you've begun coding a project, you may not change the size of your team or exchange partners without our permission. If your team has irreconcilable conflicts after beginning a project, you must speak to me before breaking up or reforming your team. Only one of the new teams (at most) will be allowed to keep the code developed thus far.

Cheating will be policed by advanced cheating-detection software. If you share code with another team, you will be caught, even if you take steps to hide your cheating.

In my experience, nobody begins the semester with the intention of cheating. Students who cheat do so because they fall behind gradually and then panic. Some students get into this situation because they are afraid of an unpleasant conversation with a professor if they admit to not understanding something. I would much rather deal with your misunderstanding early than deal with its consequences later. Even if you are convinced that you are the only person in the class that doesn't understand the material, and that it is entirely your fault for having fallen behind, please overcome your feeling of guilt and ask for help as soon as you need it. Remember that the other students in the class are working under similar constraints—they are taking multiple classes and are often holding down outside employment.

And Another Reminder (below)

"…Before you develop your solutions to each problem you are encouraged to discuss it with other students, in groups as large or small as you like. When you turn in your solution, you must give credit to any other student(s) who contributed to your work. Working on the homework in groups is both a good way to learn and a lot more fun!

…

Working cooperatively in groups is a change from the traditional approach in schools, in which students work either in isolation or in competition. But cooperative learning has become increasingly popular as educational research has demonstrated its effectiveness. One advantage of cooperative learning is that it allows us to give intense assignments, from which you'll learn a great deal, while limiting the workload for each individual student.

Another advantage, of course, is that it helps you to understand new ideas when you discuss them with other people. Even if you are the "smartest" person in your group, you'll find that you learn a lot by discussing the course with other students. For example, in the past some of our best students have commented that they didn't really understand the course until they worked as lab assistants and had to explain the ideas to later students."

…….

"Copying all or part of another person's work, or using reference materials not specifically allowed, are forms of cheating and will not be tolerated."

The following is an excerpt from the course information given in a recent CS course at Stanford University:

"… In computer science courses, it is usually appropriate to ask others — The TA, the instructor, or other students — for hints and debugging help or to talk generally about problem-solving strategies and program structure. In fact, I strongly encourage you to seek such assistance when you need it. The important point, however, is embodied in the following rule:

Rule 1: You must indicate on your submission any assistance you received.

If you make use of such assistance without giving proper credit, you may be guilty of plagiarism.

In addition to providing proper citation—usually as part of the comments at the beginning of the program—it is also important to make sure that the assistance you receive consists of general advice that does not cross the boundary into having someone else write the actual code. It is fine to discuss ideas and strategies, but you should be careful to write your programs on your own. This provision is expressed in the following rule:

Rule 2: You must not share actual program code with other students.

In particular, you should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it. Similarly, you should not discuss your algorithmic strategies to such an extent that you and your collaborators end up turning in exactly the same code.

Discuss ideas together, but do the coding on your own.

The prohibition against looking at the actual code for a program has an important specific application in computer science courses. Developing a good programming assignment often takes years. When a new assignment is created, it invariably has problems that require a certain amount of polishing. To make sure that the assignments are as good as they can be, Stanford's department—like most others in the country—reuses assignments over the years, incorporating a few changes each time to make them more effective. The following rule applies in all computer science courses:

Rule 3: You must not look at solution sets or program code from other years.

Beyond being a clear violation of academic integrity, making use of old solution sets is a dangerous practice. Most assignments change in a variety of ways from year to year as we seek to make them better.

Each year, however, some student turns in a solution to an assignment from some prior year, even though that assignment has since changed so that the old solution no longer makes sense. Submitting a program that solves last year's assignment perfectly while failing to solve the current one is particularly damaging evidence of an Honor Code violation.

Whenever you seek help on an assignment, your goal should be improving your level of understanding and not simply getting your program to work. Suppose, for example, that someone responds to your request for help by showing you a couple of lines of code that do the job. Don't fall into the trap of thinking about that code as if it were a magical incantation—something you simply include in your program and don't have to understand. By doing so, you will be in no position to solve similar problems on exams.

The need to understand the assistance you receive can be expressed in the following rule:

Rule 4: You must be prepared to explain any program code you submit.

Although you should certainly keep these rules in mind, it is important to recognize that the cases that we bring forward to Judicial Affairs are not those in which a student simply forgets to cite a source of legitimate aid. Most of the students we charge under the Honor Code have committed fairly egregious violations. Students, for example, have rummaged through paper recycling bins or undeleted trash folders to come up with copies of other students' programs, which they then turn in as their own work. In many cases, students take deliberate measures—rewriting comments, changing variable names, and so forth — to disguise the fact that their work is copied from someone else.

Despite such cosmetic changes, it is easy to determine—and we have tools for doing so—that copying has occurred. Programming style is highly idiosyncratic, and the chance that two submissions would be the same except for variable names and comments is vanishingly small.

…."

The following is an excerpt from the course information given in a CS course at Columbia University (eon ago):

“**Cheating**

Cheating makes me extremely unhappy. It’s dishonest, unfair to other students who are doing their own work fairly, and very unpleasant to deal with as an instructor. And since misery loves company, I will do my best to make cheaters extremely unhappy too …”