Academic Code Committee Report for 2009 – 2010

Thomas W. Doeppner
Associate Professor (Research)
Department of Computer Science
Brown University

This report has two parts. The first describes the committee and summarizes its actions over the past academic year. The second contains my opinions on the reasons for the dramatic imbalance in the number of code cases coming from Computer Science courses and other courses, what this may tell us about academic-code violations at Brown in general, and what might be done about it.

1 Facts

The Standing Committee on the Academic Code meets when necessary "to hear cases involving charges of Academic Code violation, to determine guilt or innocence, and to determine penalty for the guilty." Its AY 09-10 membership is:

- Thomas Doeppner – Associate Professor (Research) of Computer Science (Co-Chair)
- Keith Brown – Associate Professor (Research) of International Studies (Co-Chair)
- Michael McKeown – Professor of Medical Science
- Alex Zaslavsky – Professor of Engineering
- Pauline Luong – Associate Professor of Political Science
- Leonor Simas-Almeida – Senior Lecturer in Portuguese & Brazilian Studies

Karen Krahulik, Associate Dean of the College, is the case administrator.

Note that Roderick Beresford, Professor of Engineering, served as a Co-Chair for the fall semester of 09-10.

The committee has heard 30 cases involving plagiarism accuses that occurred this academic year and has (at least) 14 pending cases. Note that, because of departures in the dean's office, we have no information about cases for which the alleged offences took place last year, but were heard this year. Of the 30 cases we have heard, the following penalties have been meted out:

- Directed No Credit with Transcript Notation: 15
- Directed No Credit: 1
- Loss of Credit in Exercise: 2
- Reprimand: 3
- No Action Taken[*]: 7
- Warning[†]: 2

[*] "No action taken" applies to students whose works were copied by others and were deemed not to be implicated in code violations.

A list by department of the number of students referred to the committee follows (these numbers include cases yet to be heard, but do not include the departments of the cases that resulted in warnings):

- Computer Science: 29
- Psychology: 3
- Chemistry: 4
- History of Art and Architecture: 2
- Education: 1
- Engineering: 1
- French: 1
- Graduate School: 1

The Organizational Review Committee recommended that the non-academic and academic disciplinary systems be merged. Dean Krahulik is drafting a plan to put this into action for next academic year. The Standing Committee structure for the academic code and its oversight by an academic dean will not change.

2 Opinions and Recommendations

That the vast majority of cases come from one department, Computer Science, seems staggering. However, Computer Science is unique in two, perhaps three ways:

1) It is the only department that uses technology to check for collaboration among students.
2) It has more teaching assistants per student than any other department, and consequently student assignments may be more carefully scrutinized than in other departments.
3) Particularly for the introductory (sub-1000-level) courses, there is a department culture of zero tolerance for cheating. All credible cases of collaboration are referred to the academic code committee.

Thus it is not surprising that Computer Science refers more cheaters to the academic code committee than any other department. However, it’s still surprising that it refers more students than all other departments put together. One can think of a number of possible reasons for this:

1) The sort of person who takes a Computer Science course is more likely to cheat than other students.
2) Computer Science does not spell out well enough what forms of collaboration are allowed and what forms are not, so that students are not aware that their actions might be considered improper.
3) It is the nature of Computer Science assignments that causes students to cheat, i.e., it is an inherent property of Computer Science.
4) Roughly as many students are cheating in other departments as in Computer Science, but they aren’t getting caught.

I briefly address each of these potential reasons in turn.

[†] “Warning” applies to students who had been referred to the committed, but for whom charges were later dropped

1) The vast majority of students caught cheating in Computer Science courses are taking introductory (sub-1000-level) courses. The number of students caught cheating in more advanced courses is far smaller, roughly similar to the rest of the university. Most of the students caught cheating in the introductory courses do not go on to more advanced CS courses. Thus the strongest assertion one might make is that the sort of person who takes just one CS course is more prone to cheating than other students. The vast majority of CS students do not get caught cheating.[‡]
2) Students in all CS courses must read and sign collaboration agreements at the beginning of each course. These agreements spell out exactly what is and what is not considered improper collaboration. Students in many of the introductory courses also must complete homework assignments in which they must answer questions about the collaboration agreements, to ensure that they fully understand them.
3) CS assignments are often both challenging and time-consuming. Students who wait until the last minute to do them might certainly be tempted to copy someone else's work. But courses in other disciplines certainly give challenging and time-consuming assignments. A History paper requires much research, thought, and time. Math and Engineering problem sets can be extremely tough, particularly at 4am. It seems unlikely that CS courses are more likely to cause cheating than those of other courses.
4) No Brown department other than CS employs a systematic approach to detecting cheating. In many of the non-CS cases brought to the Academic Code Committee it was bad luck or incompetence on the students' parts that resulted in their getting caught. A faculty member finds a passage suspicious, so he or she Googles it and gets a hit (some students have actually plagiarized passages from Wikipedia). For example, one student's entire master's thesis was found, looked up on Google, to have been a plagiarized copy of a thesis by another student at another university. And another student was caught because he accidentally handed in his "cheat sheet" with the exam.

There are no good data on how much cheating goes on at Brown. However, if we believe that most cheaters in CS courses get caught, and that it is unusual for a cheater to get caught (and prosecuted) in other departments' courses, then the percentage of students cheating in CS courses is probably a good indication of the percentage of students who cheat campus-wide. If this is indeed the case, we have a serious problem.

In general, Brown takes the ostrich approach towards cheating — we want to pretend it doesn't happen. The outcomes of academic-code cases are kept as confidential as possible. For the sake of the individual students, this makes sense. But we also keep confidential the fact that cheating occurs at all at Brown. Thus the facts that students do (occasionally) get caught and that punishments are meted out have no deterrence value. *I think we should consider publicizing, on either a semester or yearly basis, summaries of the actions of the Academic Code Committee.*

The technology CS uses to catch cheaters is MOSS (Measure of Software Similarity), a free service provided at Stanford University (http://theory.stanford.edu/~aiken/moss/). It is used in courses with programming assignments: all student hand-ins, as well as those from previous years, if relevant, are uploaded to MOSS, and MOSS returns its analysis indicating suspicious similarities. The program deals well with attempts by students to disguise copied code such as

[‡] One could, of course, interpret this fact as meaning that most CS students are simply better cheaters than those who get caught. This seems unlikely, considering the high quality of the tools used to detect collaboration.

renaming identifiers, changing spacing, etc. It can handle only computer code, and only code in certain computer languages, but for most introductory courses that involve programming it works very well.

Would it make sense to use something similar to MOSS for papers? At least one such service exists — turnitin.com (http://turnitin.com/static/aboutus.html) — though it is not free. There are clearly valid concerns about using such a service, such as privacy and intellectual property issues[§]*. However, if the level of cheating, campus-wide, is similar to that in CS, it is important that we at least discuss the possibility of making such a service available to all departments.*

Finally, perhaps some sort of study could be commissioned to determine the number of students who actually do cheat at some time during their years at Brown.

[§] There was a recent court case concerning turnitin.com and copyright violation that was settled on turnitin.com's favor, see http://www.plagiarismtoday.com/2008/03/25/iparadigms-wins-turnitin-lawsuit/.