Check for updates

PANEL DISCUSSION

Plagiarism in Computer Sciences Courses

Philip L. Miller, Moderator
Carnegie-Mellon University

William Dodrill
West Virginia University

Michael Shamos
Carnegie-Mellon University

Doris K. Lidtke
Towson State University

Mary Dee Harris Fosberg
Loyola University

Cynthia Brown
Indiana University

With the dramatically-expanding use of computing systems has come the need to provide introductory programming courses for greater numbers of students and for students with increasingly diverse backgrounds, interests, and capabilities. Computer programming is not a skill which can be taught effectively. It is an ability which is acquired and enhanced through practice. Programming courses must, therefore, provide an opportunity for students to write, test, and perfect meaningful programs.

The acquisition of skills in computer programming can be, and often is, a challenging and rewarding experience. Unfortunately, the need to teach larger classes consisting of a wider variety of students has introduced many problems. Outstanding among these is the tendency of students to resort to unorthodox means in fulfilling course requirements. In other words, students cheat.

Considering the current nature of programming courses, this is not surprising. Many students must take computer programming whether they have an interest in the subject or not. In order to complete course requirements, they must spend many hours writing and debugging programs. It is little wonder that students resort to such tactics as copying programs, stealing programs written by other students, and paying to have programming assignments written for them rather than accepting the challenge of meeting course requirements through their own efforts.

What constitutes cheating on programming assignments? What methods can be used to detect cheating? What should be done with offenders? How can cheating be eliminated in programming courses? These are all pertinent questions, but they are directed more towards treating symptoms rather than towards correcting some very fundamental problems.

The primary difficulty in teaching computer programming is not necessarily centered around detecting and punishing cheating cases, but rather on how to teach a discipline with the unique characteristics of computer programming in a way that will encourage individual effort and reward individual achievement. Examples of questions which might be posed in order to improve teaching methods include: How can student interest in computer programming be stimulated? What can be done to reduce the frustrations inherent in writing and debugging code? What should be expected (and what should not be expected) of students taking introductory programming courses? How can individual performance and achievement be measured effectively for grading purposes?

If answers to questions like these can be found, it is quite natural to expect that the prevelence of cheating in introductory programming courses will diminish.

William Dodrill

Plagiarism and cheating do exist and do present us with special problems which differ from those confronted by instructors in many other disciplines. Traditionally the most common preventive measure against these infractions has been to insist that students work alone, produce their own unique programs, and on the whole avoid collaborative efforts. However effective this may be in preventing cheating (and is cannot always be effective), in practice it creates contervaling obstacles to effective learning.

In computer science it is particularly valuable for students to work cooperatively. Throughout their professional careers they will be working in teams and it is a poor educational system which does not prepare them for this. We should foster teamwork, rather than isolated individual activity; we should train students to work together, rather than looking upon it with suspicion; and we should encourage the sharing of ideas, rather than a jealous secrecy. There is nothing inherently unethical about such collaborative work. At the same time it is encumbent upon instructors and the profession in general to encourage mutual honesty, open frankness about how results have been achieved, and enthusiasm for a subject which can be approached cooperatively. If these principles are realized, then the problem of cheating and plagiarism diminishes substantially in importance.

Doris K. Lidtke

Plagiarism on programming assignments has been a persistent problem for Computer Science educators, but has not been faced squarely by the profession as a whole. Yet as educators we have a responsibility to deal with the problem. Students should be given a sense of values regarding their chosen discipline, and employers hiring Computer Science graduates should be able to trust that a student's knowledge and ability in the subject, not his proficiency at deception.

The motives of students who copy programs are many. The most common motive is probably the desire to "get something for nothing," to get a good grade (or at least a passing grade) without the effort or the talent required for the grade.

Another fairly common cause of plagiarism is ignorance, or naivete, among students. Everyone is taught in grammar school about identifying sources of information in written work--use of quotation marks, footnotes, etc., but when are students taught the same guidelines about wirting programs? Usually never, although it should be at the time they write their first programs. There are other, less frequent motives for plagiarism. Some students plagiarize for "oneupsmanship," attempting to prove to the teacher that they can "pull a fast one" on the teacher and get away with it. Other students cheat only on assignments that they consider "busy work". One example would be a student required to write a COBOL program for the survey of programming languages course. He said, "Writing a COBOL program is a waste of time", and copied another student's program instead of writing his own.

Prevention of plagiarism must begin with an explicit definition of plagiarism and consequences. This must be done before the student has a chance to fail because of ignorance of the rules. Drawing the line between working together and copying is difficult but necessary. Discussion of alternative algorithms or data structures between students seems reasonable, whereas cooperative creation of a program is taboo. Conferring with a lab assistant must be allowed, yet some lab assistants provide more actual program statements than they should.

The consequences of plagiarism should be reasonable--yet severe enough to make the point that plagiarism will not be tolerated. Assigning no credit for the program or sharing the grade among the guilty students seem to be appropriate penalties for first offenders. Penalties must be applied fairly and firmly, so that students understand the instructor's policy. Repeat offenders must be dealt with more severely with penalties appropriate to the offense.

With critical problems of computer fraud and software theft increasing all the time, making Computer Science students aware of the ethics of the computer industry seems not only appropriate but necessary.

Mary Dee Harris Fosberg

I consider cheating on program assignments to be collusion or copying in the writing of code. (The extent to which general discussion of the nature of the problem, or help with debugging, is allowed should vary with the level of the course.) Plagiarism is mostly found in lower level courses; if it is easy to plagiarize (as is usually the case) then the problem may become endemic, leading to demoralization and cynicism on the part of students. The large size of most sections of elementary computer science courses mandates the use of automatic methods for detecting similar programs. The ideas of Halstead on defining software metrics seem the most promising in this regard. The penalty for plagiarsim should be sever (much worse than the penalty for not handing in the assignment). Cheating probably cannot be eliminated entirely without imposing an unacceptable police-state-like system, but a good automatic detection method and imposition of severe penalties for offenders should reduce it to a tolerable level.

Cynthia Brown