Informatics in Education, 2014, Vol. 13, No. 2, 179–195

DOI: http://dx.doi.org/10.15388/infedu.2014.01

Collaboration, Collusion and Plagiarism in Computer Science Coursework

Robert FRASER

University of Manitoba
Department of Computer Science, University of Manitoba, 66 Chancellors Circle
Winnipeg, Manitoba, Canada, R3T 2N2
e-mail: fraser@cs.umanitoba.ca

Received: May 2013

Abstract. We present an overview of the nature of academic dishonesty with respect to computer science coursework. We discuss the efficacy of various policies for collaboration with regard to student education, and we consider a number of strategies for mitigating dishonest behaviour on computer science coursework by addressing some common causes. Computer science coursework is somewhat unique, in that there often exist ideal solutions for problems, and work may be shared and copied with very little effort. We discuss the idiosyncratic nature of how collaboration, collusion and plagiarism are defined and perceived by students, instructors and administration. After considering some of the common reasons for dishonest behaviour among students, we look at some methods that have been suggested for mitigating them. Finally, we propose several ideas for improving computer science courses in this context. We suggest emphasizing the intended learning outcomes of each assignment, providing tutorial sessions to facilitate acceptable collaboration, delivering quizzes related to assignment content after each assignment is submitted, and clarifying the boundary between collaboration and collusion in the context of each course. While this discussion is directed at the computer science community, much may apply to other disciplines as well, particularly those with a similar nature such as engineering, other sciences, or mathematics.

Keywords: collaboration, collusion, plagiarism, computer science, academic dishonesty.

1. Introduction

This paper contains an examination of academic dishonesty with respect to assignments in computer science courses. In particular, this discussion is directed toward computer science courses that use the model where students are usually expected to work individually to complete their coursework. Typically, this includes introductory courses where students learn and make use of a single programming language as a means to understanding basic algorithms and data structures, as well as upper-level undergraduate and graduate courses with a strong theoretical basis. Our objective is to shed light on the problems that may arise in such courses due to dishonest behaviour, whether intentional or not. An understanding of such issues will facilitate the design of courses that provide better learning opportunities for students and consequently improve the course for all interested parties.

The topic of academic dishonesty is very broad, and so we refine the discussion to focus on coursework in the kinds of courses described above. There are many computer science courses that contain a significant project as a fundamental component of the curriculum, and so these courses may be designed to be group-oriented (such as software engineering or operating systems courses). While some of the issues discussed in this paper will apply to this type of course, there are a number of additional considerations that arise with respect to the nature of group work, see e.g. Waite *et al.* (2004). Furthermore, we restrict the discussion to coursework. Dishonesty on examinations is another important concern, but the framework and objectives are distinct from those for coursework, and we do not address these issues in this study. See Weber *et al.* (1983), Brimble and Stevenson-Clarke (2005), and de Bruin and Rudnick (2007) for some interesting discussions with respect to examinations. Finally, topics in this paper are usually discussed in the context of computer science courses, but many of the ideas and conclusions presented here apply beyond this scope and will hopefully be useful to instructors and researchers in other areas.

Computer science is subject to a significant amount of cheating, and we begin by looking at why the field may be prone to some of these practices. Primarily, the nature of many computer science assignments is that there is an ideal solution for each question, and as a consequence the best answers will be highly similar. Students are well aware of this property, and dishonest students will exploit it. We take a detailed look at collaboration, collusion, and plagiarism in turn, and determine where the lines are drawn between them. We show that the boundaries are difficult to define objectively, and that there is often confusion and even disagreement among students and also among instructors as to what should be considered acceptable behaviour.

We follow up with a discussion on mitigating dishonesty, where we review some of the reasons that students cheat in general, and we examine some of the practices that have been used to try to reduce dishonesty. The conventional detection and discipline approach is effective at deterring some dishonest behaviour, but it should be complemented with additional approaches which are also geared toward improving learning opportunities for the students. To accomplish this, we investigate some interesting roots of dishonest behaviour. Primary causes include a lack of foundational knowledge, an inflated sense of entitlement or apathy, or a lack of motivation. We discuss each of these problems in turn, and we review the effectiveness of various strategies for addressing them.

Next, we discuss policies for permissible collaboration, and we present ideas for improving the design of a course in computer science, both in terms of reducing academic dishonesty and improving student learning. By reinforcing the intended learning outcomes of an assignment, a student may better understand the purpose of the work, and feel more motivated to complete the task as a consequence. We discuss how tutorial sessions may be used to reinforce proper collaboration and improve student learning.

We show that student response systems (such as clickers) are a popular means to keep students engaged, and we present evidence showing that this practice will reduce collusion and/or plagiarism. Finally, we outline how a scenario-based discussion can be used to mitigate confusion regarding acceptable collaboration.

2. Academic Dishonesty in Computer Science

It is telling that there exists a plethora of studies dedicated to academic dishonesty specifically within the fields of computer science and information technology (e.g., Roberts (2002), Sheard *et al.* (2003), Barrett *et al.* (2004), Dennis (2004), and Culwin (2008)). There are many reasons for this focus, but a principal one is that computer science accounts for more than its share of incidents of academic dishonesty. For example, Roberts (2002) reviewed incidents of dishonesty at Stanford University over a decade, and found that 37% of all incidents were attributed to computer science courses, while their students represented less than 7% of the student population at the university. MIT, widely regarded one of the top computer science schools in the world, made headlines when an introductory computer science course was found to have rampant plagiarism which resulted in disciplinary action against 73 students (Butterfield, 1991).

Whether the high rates of collusion and plagiarism among computer science students are due to higher rates of dishonesty in computer science or are a result of the use of better detection tools is open to debate. Barrett and Cox (2005) hypothesized that mathematics and computer science are subject to higher rates of collusion because of the nature of the work in these fields. While arts and humanities assignments typically involve personal reflection on the parts of the students, assignments in mathematics and computer science typically have an ideal solution which the students are seeking. For the same reason, it is difficult to detect dishonesty among correct solutions; similar incorrect solutions or strange approaches are those that usually ring alarm bells. Striving to find the ideal correct solution can be an exhausting and frustrating experience, and the temptation to be relieved can be overwhelming. The nature of computer science assignments was encapsulated beautifully by Roberts (2002): “The computer is a relentlessly unforgiving arbiter of correctness.” Because of unique character of computer science assignments, instructors of such courses must develop policies that are specific to their needs (Riedesel *et al.*, 2012; Simon *et al.*, 2013).

It is difficult to objectively establish a distinction between group work that is allowable or not. For the remainder of this section, we discuss the concepts of collaboration, collusion and plagiarism in turn. We devote greater attention to collusion as it occupies a sort of ill-defined grey area between collaboration and plagiarism.

2.1. *Collaboration*

Collaboration is distinguished from collusion in this discussion by defining both activities as a group of students working together on an assignment, but collusion is group activity that is unpermitted (collusion and plagiarism are distinguished shortly). As an example, many courses permit collaboration in the sense that students may meet to discuss ideas for an assignment, but students are prohibited from writing the assignment solutions together, which would be considered collusion.

Collaboration can be beneficial for student learning (Sabin and Sabin, 1994). As educators, our primary objective is for the students to truly understand the material that we are presenting. This goal may be realized by making use of active learning techniques. Fink (2003, p.108) promotes active learning by a variety of means, but a key aspect is through student discussions of the course material. Assignments provide perhaps the ideal setting for such discussion, and the challenge is to encourage discussion to the point that it is acceptable or most beneficial. Allowing group discussion facilitates these active learning opportunities, while the requirement that students individually complete assignments is intended to ensure that they have an understanding of the material. Excessive collaboration will certainly undermine a weaker student's ability to learn from the assignment, and so one of the aims of this paper is to elucidate issues affecting these potentially conflicting objectives.

2.2. *Collusion*

In general, the boundary between collaboration and collusion is ill defined and highly variable, and students have a poor understanding of the boundary (Joy *et al.*, 2011). Although students and educators are able to clearly identify plagiarism when it occurs, the same cannot be said for collusion (Barrett and Cox, 2005). Collusion may be regarded as the middle ground in a spectrum of practices ranging from collaboration to outright plagiarism, and it is best defined as unpermitted collaboration. The boundary between collaboration and collusion is flexible, and varies depending on the desires of the instructor for a given course. This lack of uniformity is unavoidable, but it is problematic. Suppose the instructions for completing an assignment include something of the form "You may discuss your ideas as a group, but the assignments must be written individually." This may lead to confusion among students, for the boundary between what constitutes discussion and writing varies. Barrett and Cox (2005) provided a set of scenarios to students and instructors, and asked whether the students in the scenarios had acted acceptably, or whether they are guilty of plagiarism, collusion, or both. The following scenario (Scenario 3 in their study) elicited the highly personal divide between what defines acceptable practice or collusion:

> *Student A doesn't know how to start the assignment and so he asks student B who helps him by showing him his own work. Student A writes up the assignment in his own words but there are some similarities with student B's work.*

For this scenario, 51% of instructors thought that this would be fine, while 39% defined the activity as collusion. Interestingly, among students, only 38% thought that it would be acceptable, 33% felt that it is collusion, and 24% considered it to be outright plagiarism (Barrett and Cox, 2005, Figure 3). These results underscore the responsibility incumbent upon an instructor to establish a clear definition of what constitutes collusion in the course being taught. It is also worth emphasizing that these rules differ by course, and that it is the responsibility of the student to act within the accepted boundaries of each course.

Collusion is generally regarded in a softer light than plagiarism. In fact, many educators are willing to tolerate collusion, since students are actually thinking and learning in the process (Barrett and Cox, 2005). Tolerance also varies with respect to the value of the task; high levels of collusion on homework with low value is easier to accept than the same for major assignments, as students may be getting a significant amount of credit for work that is not their own. Franklyn-Stokes and Newstead (1995) conducted a study which asked students and educators to rank different types of cheating behaviour by seriousness, and also by their perceived frequency. They identified an inverse relationship between these rankings; behaviours that are considered more serious violations occur less frequently. One observation based upon these results is that cheating on exams is considered more serious and occurs less frequently than cheating on coursework. Another observation is that plagiarism is generally considered more serious than collusion (as one would expect).

2.3. *Plagiarism*

Plagiarism is "the action or practice of taking someone else's work, idea, etc., and passing it off as one's own; literary theft" (OED, 2011). For the purposes of this discussion we expand on this definition; in particular, we consider plagiarism to be the act of copying someone else's work whether or not the original author is aware of the act. Therefore, under this definition, it is considered an act of plagiarism if someone gives their work to another for copying (a "literary gift", so to speak).

Plagiarism is particularly a problem in computer science courses, where the ideal solution to a problem may be obtained with a simple cut-and-paste operation. Within the category of plagiarism, there again exists a spectrum of severity of offences. For example, while a student is guilty of plagiarism if code is obtained from an uncredited source online, it is possible that the (credited) use of such code is permitted by the instructor and the omission of due credit was an oversight. This type of plagiarism may be handled with some leniency, particularly for a first offence. At the other end of the spectrum is the student who obtains a solution from a peer and then adapts the code for the sole purpose of obfuscating their actions. This is particularly an issue if some amount of collaboration is permitted (and encouraged).

Plagiarism is regarded as a serious academic offense, and institutions go to great pains to impress upon the students the severity of the penalties for students who are caught (Roberts, 2002). East (2010) provides an anecdote where an instructor took a controversial stand against plagiarism that resulted in an entire class being charged with academic misconduct. The issue becomes thorny when it is questionable whether the students were knowingly dishonest, and consequently it is incumbent upon a course instructor to clearly and explicitly state what the policies for the course are. This can mitigate some incidents of plagiarism, particularly those of an unintentional nature. In general, students have a poor perception of what constitutes plagiarism and will often err too heavily on the side of caution (Ashworth *et al.*, 1997). This point has been reinforced by observations that the amount of plagiarism actually taking place has been declining, at least amongst computer science students (Culwin, 2008; Sheard and Dick, 2011), although the general perception is the opposite (Dick *et al.*, 2003). Certainly, evidence presented by Dick *et al.* (2003) suggests that students tend to cheat when given the opportunity to do so.

It is worth considering appropriate policy with respect to self-plagiarism (also known as auto-plagiarism). Roig (2009, p.16) suggests that the essence of self-plagiarism is that previous work is submitted with the implicit understanding that it is new work. For our discussion, this would likely be most relevant in upper-year undergraduate or graduate courses where students are free to select projects suiting their personal interests. Such projects may have significant overlap with some of the student's previous work as a natural consequence. Students should be made aware of the course policy with respect to self-plagiarism in such circumstances. A suitable policy should reflect the standards in place for academic publishing: if new work bears a strong resemblance to previous work, then this should be disclosed and due care should be taken to ensure that new work is sufficiently original. Failure in either of these regards constitutes an act of plagiarism (Roig, 2009).

3. Mitigating Dishonesty

In this section we consider some of the common reasons for dishonest behaviour amongst computer science students, and we examine some of the practices that are commonly used to discourage or deter such practices, both from the perspectives of students and instructors. We begin by looking at detection techniques and disciplinary action, which is the traditional approach to dealing with academic dishonesty (correspondingly, the threat of disciplinary action is the usual approach to mitigating dishonesty). We follow up with an examination of other approaches that may be used to mitigate specific causes of academic dishonesty (such techniques should typically be applied in conjunction with the threat of disciplinary action). These include ensuring that the students possess or have access to necessary foundational knowledge, dispelling any inflated sense of entitlement, and properly motivating students.

3.1. *Detection & Discipline*

One of the significant challenges for mitigating academic dishonesty is detection. Various studies have shown the rate of cheating in university courses to ranges from 40% to 96% (e.g., McCabe and Trevino (1996), Dick *et al.* (2003), and Yeo (2007), see Park (2003) or Ercegovac and Richardson (2004) for fairly comprehensive reviews), while the rates of detection may be as low as 1.3% (Björklund and Wenestam, 1999). The improvement of detection techniques is being aggressively undertaken, as it is believed that high rates of detection and prosecution will deter cheaters. If students believe that cheating is commonplace, then they are more likely to cheat themselves, as they feel that this creates a level playing field.

There are a number of commercial software applications which have been created to aid in the detection of plagiarism (see Maurer *et al.,* (2006) for a thorough review). A recent study found that over a third of instructors are using such tools (Dick *et al.*, 2003), and universities encourage this practice. Turnitin is one such application, which may be made freely available to instructors by their institution. Turnitin may even be used formatively, so that students improve their working habits and receive reinforcement for best practices with respect to course policies on plagiarism (Barrett and Malcolm, 2006; Rolfe, 2011), which may in turn reduce or even remove the need for the tool for the purposes of plagiarism detection (Chew *et al.*, 2013). A number of tools have been developed specifically for computer science courses by addressing the unique challenges of examining code, see e.g. Lancaster and Culwin (2004), Cosma and Joy (2012), or Ðurić and Gašević (2013). Occasionally students raise concerns regarding the privacy and security of their intellectual property with these kinds of tools, or with the efficacy of the tool itself (Jones and Moore, 2010). Some institutions give students the right to refuse the screening of their assignments by automated tools (Mitchell, 2011), and in such cases the student and instructor may agree upon an alternative form of additional screening.

The penalties that may be applied if students are caught cheating range in severity from receiving a zero on the offending assignment or test to expulsion from the university. While severe penalties are a deterrent, a survey of students by Sheard *et al.* (2003) found that it ranks fifth on a list of deterrents, below factors such as knowing the value of their work and pride in their work. However, simply appealing to the moral code of the students does not reduce cheating rates, while the credible threat of detection and punishment has been demonstrated to reduce the amount of cheating in a course (Hollinger and Lanza-Kaduce, 2009). At the other end of the spectrum, employing a strict and severe disciplinary strategy, exemplified by an aggressive zero-tolerance approach to prosecuting academic dishonesty, may deter even honest students from enrolling in a course (Levy and Rakovski, 2006).

Professors are busy and some regard dealing with cases of academic dishonesty as a waste of their time. In a survey of nearly 500 university professors conducted in 1996, it was found that 20% of respondents had ignored cases of blatant cheating (Björklund and Wenestam, 1999). Surveys of faculty members have revealed that large percentages had ignored cases of suspected cheating (e.g. 40% (Coren, 2011) and 51% (Barrett and Cox, 2005)). However, official university policy typically forbids personal judgment when an instructor is considering whether to prosecute suspected cheaters (see Mitchell (2011), for example). It is critically important that students have a good understanding of what the course policies are with respect to plagiarism, as students may have views differing from those of their instructors (O'Regan, 2006; East, 2010). We show how to address this issue in the *Mitigating Confusion* section on page 12.

3.2. *Foundational Knowledge*

Students who are more comfortable with course material are less likely to cheat (Ashworth *et al.*, 1997). In particular, in computer science, students entering their first year of study are assumed to have a certain level of background knowledge. Students who fall short of the prerequisites are more likely to cheat than their better prepared peers (Dennis, 2004). Also, the level of maturity and personal motivation of students is inversely related to their tendency to cheat (Sheard *et al.*, 2003).

Some students will elect to cheat even in classes where all students begin the course with roughly equivalent skill sets (Palazzo *et al.*, 2010). This leads to poorer learning amongst the cheaters, and subsequently higher rates of attrition. Roberts (2002) echoes this sentiment, in that cheaters may be students who have fallen behind in the material during the progression of the course and are completely lost when attempting assignments later in the term. Therefore, one approach to mitigating dishonesty is to ensure that students possess the necessary foundational knowledge for the course (Davis, 1994). Of course, this is not always possible for all students, as it may be like leading the proverbial horse to water[1]. The use of teaching strategies which require students to keep abreast of the course material will mitigate the latter category of offences. Caution is necessary, since simply loading students with exercises in order to address the deficiencies of those lacking foundational knowledge may in fact lead to poorer performance in the course (Kontur and Terry, 2013). We discuss how student response systems may be used for addressing common knowledge gaps on page 12.

3.3. *Entitlement & Apathy*

Students may be inclined to cheat because they feel a sense of entitlement to their grades (Naude and Hörne, 2006). Such students may regard paying tuition as a transaction to purchase their grades and degree (Auer and Krupar, 2001). In the cheating incident at MIT, many students attested that they deserved a good grade because they had invested a significant amount of time in the assignment (Butterfield, 1991). This association of reward with effort rather than product is distressing, to say the least. How can an instructor address such perceptions?

One plausible approach is to emphasize that those students who conduct themselves honestly tend to perform better in the course. Palazzo *et al.* (2010) observed the performance of students with regard to their rates of cheating in a course as the term progressed. Students who were the worst offenders on the assignments performed the poorest on their exams. On average, students who copied more than 50% of their assignment work earned nearly two letter grades lower on the final exam than students who copied less than 10% of the work on assignments. Further, while only 20% of students cheated on over 30% of the material, they represented 47% of the students who fail the course. These results are intuitive and hardly surprising. However, it is rather surprising and even disheartening to discover that dishonest students are comfortable with the fact that

[1] For the unfamiliar, the proverb states that you can lead a horse to water, but you cannot force it to drink.

cheating hurts them (Palazzo *et al.*, 2010). In their study, students were shown a graph at the beginning of the term clearly demonstrating a decline in performance with increased cheating behaviour, but no decline in the prevalence of cheating in the course was observed when compared with previous sessions of the course. Furthermore, some students who were caught cheating rationalized their actions in precisely this context, demonstrating that they understood and accepted the consequences of their actions, e.g.: "cheating isn't bad because it only hurts you at test time" (Palazzo *et al.*, 2010, p.8).

Battling a sense of apathy is challenging. Some students confess that they cheat because they are simply lazy (Sheard *et al.*, 2003; Wilkinson, 2009). Dennis (2004) found that a related factor, that students ran out of time because they started the assignment too late, was the top reason that students plagiarized. However, it is conjectured that this is more attributable to poor time management than outright laziness. The utilization of strategies which assist with time management, particularly for first year students, would mitigate dishonesty among students who find themselves in these circumstances. We discuss how tutorials may be used for this purpose shortly.

3.4. *Motivation*

Some students who cheated on assignments rationalize their actions by claiming that the assignment is a waste of their time or that they are not motivated to complete the task (Howard, 2001; Palazzo *et al.*, 2010). To mitigate this, Palazzo *et al.* (2010) suggested that a course design which involves more teacher interaction results in students believing that the instructor is more concerned with their learning (rather than simply assigning a grade based on their performance). They assert that such modifications to the physics courses at MIT is a primary factor in the roughly 75% reduction in cheating that they have observed since implementing the changes.

This leads us to the hypothesis that another technique for addressing this issue is to ensure that the students are aware of the Intended Learning Outcomes (ILOs) for a given assignment.[2] Dick *et al.* (2003) found that when students possessed a clear understanding of the reasons that the work has been assigned to them (i.e. the ILOs), they are less likely to cheat. This may be done informally by mentioning the ILOs during class, but a stronger approach would be to explicitly describe them on the assignment itself. We discuss this further in the following section.

4. Establishing Boundaries & Improving Learning Opportunities

Our interest in this study has focussed on choosing the boundaries for collaboration and how to improve course design to enhance student learning opportunities in this context. The status quo is that the professor is free to set the boundary between collaboration

[2] The rigid use of intended learning outcomes has faced criticism, but this is primarily directed at settings such as lectures where class interaction may lead to the worthwhile exploration of ideas potentially differing from the ILOs of the lecture (Hussey and Smith, 2003, 2008). This criticism is well founded, but it is not relevant to our application.

and collusion where they see fit, but there is little guidance on what constitutes an appropriate choice. We begin by discussing the drawbacks of the current model, and we reinforce the need for boundaries to be in place. Finally, we propose a variety of strategies that may be used to establish boundaries and encourage acceptable collaboration in a computer science course. Note that there is no single optimal approach with respect to course design, as effective policies for one course may not suit another. Choices should be based on the course material and structure and the intended learning outcomes of the assignment. Our goal is to aid course development by highlighting some options available to computer science instructors.

4.1. *The Status Quo*

In theory, the current model that many computer science courses use is ideal. As mentioned previously, many courses encourage students to discuss the assignment questions, but the writing must done individually. There are a number of problems with the usual implementations of this model, however. Dishonest students ignore the restrictions, or students may not understand the boundaries. Many honest students err on the side of caution, and some will avoid group work altogether to be safe. The result is that some of the best students (or at least some of the honest ones) are missing out on the active learning opportunities associated with the assignment because they feel that it is the most honest course of action.

Course instructors may emphasize the acceptable limits of collaboration, but often do little to facilitate or encourage students to work near the boundary of acceptable behaviour. The time constraints on a course are a significant reason for this.

4.2. *Unlimited Collaboration*

One approach to reducing collusion would be to allow any amount of collaboration among the students, with the result that there is no need to check for collusion. In the extreme, this would permit outright copying of another student's solutions if the student so desires. However, as discussed previously, Palazzo *et al.* (2010) demonstrated that providing students with the knowledge that cheating hurts them and appealing to their maturity in this regard has no effect on reducing the level of cheating that occurs. It is evident that simply permitting any level of collaboration would increase the amount of copying that would take place in the course, and would thus damage the average student's learning experience, and so this policy is not recommended.

4.3. *Improving Tutorials*

A first suggestion for improving course assignments is to encourage collaboration on the assignment questions during tutorial sessions where instructors and/or teaching assistants (TAs) are available to facilitate group discussion. Further, these sessions could be used for discussion of assignments that have been returned so that students may compare the solutions that they developed to the ideal solutions that were used for grading. If managed properly, these tutorial sessions would create active learning opportunities, so that students who are comfortable with the material may assist other students or engage in high level discussions with their peers. Furthermore, by having instructors and/or TAs present to observe student interaction, best practices for collaboration may be actively reinforced. There are several challenges that have to be addressed with such a proposition.

The first problem is simply a matter of time. It may be difficult to introduce tutorial sessions into courses which lack them at present. However, to begin with, this format could be implemented in a first or second year course which already has tutorial sessions. An added benefit of this proposal is that it may improve tutorial attendance. Tutorial sessions in these courses are often sparsely attended because many undergraduates do not see the value of tutorials (Baderin, 2004). A correlation between the tutorial sessions and assignments may help improve attendance and they would provide an opportunity to reinforce the intended learning outcomes for the assignment.

Another challenge is to minimize collusion and plagiarism, while these tutorial discussions may actually facilitate such actions. However, the prohibitions that are enforced presently would be maintained, and the same plagiarism detection techniques would be used. The tutorial sessions, if used correctly, may actually reduce the tendencies toward plagiarism. One of the major reasons identified earlier as to why students cheat was that they had poor time management or had procrastinated, and so they did not have enough time to finish the assignment on their own. By timing the assignment discussions in tutorials so that they occur on the order of several days before the assignment is due, the students will be primed into actively thinking about the assignment material with adequate time to complete the problem set properly.

On a related note, the use of tutorials can help students keep up to date on the course material. A lack of foundational knowledge and students falling behind in the course were several more of the primary reasons for dishonest behaviour identified earlier.

4.4. *Intended Learning Outcomes*

As educators, we often feel that the lessons to be learned by course work are self-evident. A course is usually designed so that the sections fit together well, and each section has a set of lessons and values that contribute to the skill set of the student. However, it is often the case that students lose sight of these values, particularly as they become pressed for time while juggling the workload resulting from taking a number of courses.

With each lecture and assignment, it is worthwhile to emphasize the intended learning outcomes (ILOs).[3] With respect to our discussion, the students should be provided with a short section at the beginning of the assignment which explicitly states what they should learn by completing the questions (Dick *et al.*, 2003). This makes it clear that the students should master these concepts if they wish to do well in the course, and that this assignment is a vehicle for that purpose rather than just some busy work.

[3] Fuller *et al.* (2007) discuss developing ILOs specifically in the context of a computer science course.

Underscoring the ILOs may also reduce the sense of entitlement among students. Students who feel entitled to a grade may regard the assignment as a transaction with the instructor, in the sense that the instructor wants a good set of solutions from the student and will provide a good grade in return. By emphasizing the ILOs as an objective for the work, the students may be more likely to see the assignment as a means to understanding the material and hence to obtaining a better grade in the course.

With these goals in mind, the instructor must consider what level of collaboration (and the associated active learning opportunities) will best help the students achieve the objectives. This may even vary from question to question within an assignment which is fine in limited circumstances provided that the boundaries are clearly defined. For example, given a challenging problem on an assignment in an introductory computer science course, one may wish to allow the students to develop an algorithm which addresses the problem in groups, while the subsequent implementation of the algorithm must be done individually.[4] In this case, it would be worthwhile to provide an example of an algorithm to unambiguously demonstrate exactly how much collaboration is permitted for each aspect of the solution.

For another example, consider an upper-year course in theoretical computer science. When students are asked to perform their first NP-completeness reductions, it may be a good active learning opportunity to allow students to discuss possible problems to use in the reduction. However, it should be emphasized where the limit of acceptable collaboration lies; in this case appropriate boundaries may include forbidding the discussion of gadget details and/or forbidding writing of any kind during the discussion.

4.5. *Quizzes*

To ensure that students are putting thought into their assignments, the instructor may provide a quiz to the class after each assignment is due (ideally in the class immediately following the assignment deadline) to test their knowledge of the assignment material. To be successful, these quizzes should be fairly easy for someone who dutifully completed the assignment. The intent is that weaker students would be compelled to understand the material during collaboration sessions, since they know that they will be tested on it shortly.

Implementing quizzes is particularly easy if the class is using a student response system such as clickers (there is a large body of work dedicated to such systems, see e.g. Caldwell (2007) and Trees and Jackson (2007)). Such a medium minimizes the time requirements for the quiz, and it may be marked instantly so that students are able to compare their level of comprehension to that of their peers. Furthermore, this facilitates immediate review during lecture if it is apparent that there is a common misunderstanding or knowledge gap among the students in the class (Hoanca, 2013). This type of approach is also well received by students, e.g. see Wood (2004).

[4] This assumes that the development of the algorithm is not one of the primary learning objectives of the assignment, but rather that the purpose of the assignment is to study the use of particular data structures or something of the sort.

In our own experience, students have relished the opportunity to discuss problems with their peers given a disparate response in class. When polled regarding their satisfaction with the response system (in this case, students used clickers to respond to 3–5 questions per lecture, upon which they were graded for correctness), only 2% of students indicated that they felt there were too many such questions, while 23% of respondents would have preferred more (the remaining 75% were happy with 3–5 questions).

4.6. *Mitigating Confusion*

Our purpose to this point has been to demonstrate that the boundaries between collaboration, collusion, and plagiarism are idiosyncratic, in that individual instructors will have their own policies which may also vary depending on the task. Furthermore, this is actually appropriate, since proper course design should consider what level of collaboration has the most benefit for the learning experience of the students. The final point that we would like to address is that of ensuring that students clearly understand the boundaries defined for each course, since a lack of understanding with respect to such boundaries is also a leading cause of plagiarism (Yeo, 2007; Wilkinson, 2009; Owunwanne *et al.*, 2010). To this end, we recommend presenting some of the scenarios of Barrett and Cox (2005) with the students (as discussed previously), to concretely demonstrate what kinds of activities are permitted in the course (this is similar to the approach advocated by Dawson and Overfield (2006)). At minimum, this will assist conscientious students in their efforts to work responsibly with their peers.

In order to implement this recommendation for our own courses, we provide the students with a set of slides describing 4 scenarios during the first lecture of the term. The scenarios are chosen to clearly illustrate the limits of acceptable collaboration for the course. At the end of the course, the students were anonymously polled regarding the efficacy of this exercise. When asked "How useful were these slides for you for understanding what constitutes acceptable behaviour for [this course]?" the responses were as follows:

63% – Useful, they helped clarify what's allowed.
25% – Useful, although they only reinforced what I already knew.
0% – I learned nothing from them.
0% – They were a waste of time.
12% – They left me more confused.

The results show that students generally appreciated the exercise, and that even more discussion on the topic may be warranted.

5. Conclusions

After distinguishing between collaboration, collusion, and plagiarism, we found that the differences are subjective in nature. A computer science course should have limits on collaboration that are defined with the learning objectives of the coursework in mind, and a course instructor should strive to ensure that the students enrolled in the course are aware of where the boundaries lie. We identified many strategies for mitigating academic dishonesty by studying some of the prevalent causes in computer science. One of the most promising avenues for improvement is to reinforce the intended learning outcomes of assignment material. We presented concrete suggestions for improving a computer science course so that the amount of cheating is reduced, while improving the learning environment for the students. These include using tutorial sessions wisely, reinforcing the intended learning outcomes of their assigned work, providing regular small quizzes, and working to reduce confusion toward the course policies.

Acknowledgements

The helpful guidance and corrections provided by a number of anonymous reviewers is gratefully acknowledged; the discussion was significantly improved due to their careful reading and insightful suggestions. This work began as a part of the Certificate in University Teaching Program at the University of Waterloo, and in particular we are indebted to Svitlana Taraban-Gordon for providing invaluable guidance for the development of this study.

References

Ashworth, P., Bannister, P., Thorne, P. (1997). Guilty in whose eyes? University students' perceptions of cheating and plagiarism in academic work and assessment. *Studies in Higher Education*, 22 (3), 187–203.

Auer, N., Krupar, E. (2001). Mouse click plagiarism: the role of technology in plagiarism and the librarians role in combating it. *Library Trends*, 15–32.

Baderin, M.A. (2004). Towards improving students' attendance and quality of undergraduate tutorials: a case study on law. *Teaching in Higher Education*, 9(4), 491–509.

Barrett, R., Cox, A., Malcolm, J., Lyon, C. (2004). Plagiarism prevention is discipline specific: a view from computer science. In: *Plagiarism: Prevention, practice and policies conference*.

Barrett, R., Cox, A.L. (2005). At least they are learning something: the hazy line between collaboration and collusion. *Assessment & Evaluation in Higher Education*, 30(2), 107–122.

Barrett, R., Malcolm, J. (2006). Embedding plagiarism education in the assessment process. *International Journal for Educational Integrity*, 2(1), 38–45.

Björklund, M., Wenestam, C.-G. (1999). Academic cheating: frequency, methods, and causes. In: *European conference on educational research*.

Brimble, M., Stevenson-Clarke, P. (2005). Perceptions of the prevalence and seriousness of academic dishonesty in Australian universities. *The Australian Educational Researcher*, 32(3), 19–44.

Butterfield, F. (1991). Scandal over cheating at M.I.T. stirs debate on limits of teamwork. *The New York Times*, May 22, A23.

Caldwell, J.E. (2007). Clickers in the large classroom: current research and best-practice tips. *CBE-Life Sciences Education*, 6(1), 9–20.

Chew, E., Ding, S., Rowell, G. (2013). Changing attitudes in learning and assessment: cast-off plagiarism detection and cast-on self-service assessment for learning. *Innovations in Education and Teaching International* (ahead-of-print), 1–10.

Coren, A. (2011). Turning a blind eye: faculty who ignore student cheating. *Journal of Academic Ethics*, 9(4), 291–305.

Cosma, G., Joy, M. (2012). An approach to source-code plagiarism detection and investigation using latent semantic analysis. *Computers, IEEE Transactions on*, 61(3), 379–394.

Culwin, F. (2008). A longitudinal study of nonoriginal content in final-year computing undergraduate projects. *Education, IEEE Transactions on*, 51(2), 189–194.

Davis, S. (1994). Teaching practices that encourage or eliminate student plagiarism. *Middle School Journal*, 25(3), 55–58.

Dawson, M., Overfield, J. (2006). Plagiarism: do students know what it is? *Bioscience Education,* 8. `http://journals.heacademy.ac.uk/doi/pdf/10.3108/beej.8.1`

de Bruin, G., Rudnick, H. (2007). Examining the cheats: the role of conscientiousness and excitement seeking in academic dishonesty. *South African Journal of Psychology*, 37(1), 153–164.

Dennis, L. (2004). Student attitudes to plagiarism and collusion within computer science. In: *Plagiarism: Prevention, Practice and Policies Conference*.

Dick, M., Sheard, J., Bareiss, C., Carter, J., Joyce, D., Harding, T., Laxer, C. (2003). Addressing student cheating: definitions and solutions. *SIGCSE Bulletin*, 35(2), 172–184.

Đurić, Z., Gašević, D. (2013). A source code similarity system for plagiarism detection. *The Computer Journal*, 56(1), 70–86.

East, J. (2010). Judging plagiarism: a problem of morality and convention. *Higher Education*, 59(1), 69–83.

Ercegovac, Z., Richardson, J. (2004). Academic dishonesty, plagiarism included, in the digital age: a literature review. *College & Research Libraries*, 65(4), 301–318.

Fink, L. (2003). Creating significant learning experiences: an integrated approach to designing college courses. Jossey-Bass, San Francisco.

Franklyn-Stokes, A., Newstead, S. (1995). Undergraduate cheating: who does what and why? *Studies in Higher Education*, 20(2), 159–172.

Fuller, U., Johnson, C., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Lahtinen, E., Lewis, T., Thompson, D., Riedesel, C., Thompson, E. (2007). Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4), 152–170.

Hoanca, B. (2013). How well do clicker scores correlate with course performance? A case study in two MIS courses. In: *Proceedings of the information systems educators conference*, 30, 1–8.

Hollinger, R.C., Lanza-Kaduce, L. (2009). Academic dishonesty and the perceived effectiveness of countermeasures: an empirical survey of cheating at a major public university. *NASPA Journal*, 46(4), 587–602.

Howard, R. (2001). Forget about policing plagiarism: Just teach. *Chronicle of Higher Education, Section: The Chronicle Review*, B24.

Hussey, T., Smith, P. (2003). The uses of learning outcomes. *Teaching in Higher Education*, 8(3), 357–368.

Hussey, T., Smith, P. (2008). Learning outcomes: a conceptual analysis. *Teaching in Higher Education*, 13(1), 107–115.

Jones, K., Moore, T. (2010). Turnitin is not the primary weapon in the campaign against plagiarism. In: *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*. New York, NY, USA: ACM, 425–429.

Joy, M., Cosma, G., Yau, J., Sinclair, J. (2011). Source code plagiarism – a student perspective. *Education, IEEE Transactions on*, 54(1), 125–132.

Kontur, F., Terry, N. (2013). The benefits of completing homework for students with different aptitudes in an introductory physics course. *ArXiv e-prints*. `http://arxiv.org/ftp/arxiv/papers/1305/1305.2213.pdf`

Lancaster, T., Culwin, F. (2004). A comparison of source code plagiarism detection engines. *Computer Science Education*, 14(2), 101–112.

Levy, E.S., Rakovski, C.C. (2006). Academic dishonesty: a zero tolerance professor and student registration choices. *Research in Higher Education*, 47 , 735–754.

Maurer, H., Kappe, F., Zaka, B. (2006). Plagiarism – a survey. *Journal of Universal Computer Science*, 12(8), 1050–1084.

McCabe, D., Trevino, L. (1996). What we know about cheating in college: longitudinal trends and recent developments. *Change: The Magazine of Higher Learning*, 28(1), 28–33.

Mitchell, B. (2011). Office of Academic Integrity, University of Waterloo. `http://uwaterloo.ca/academicintegrity/index.html`

Naude, E., Hörne, T. (2006). Cheating or collaborative work: Does it pay? *Issues in Informing Science and Information Technology*, 3(1), 459–466.

OED. (2011). Oxford English dictionary: plagiarism, n. `http://www.oed.com/view/Entry/144939?redirectedFrom=plagiarism`

O'Regan, K. (2006). Policing – or, at least, policying – plagiarism at one Australian university. *Journal of University Teaching & Learning Practice*, 3(2), 113–122.

Owunwanne, D., Rustagi, N., Dada, R. (2010). Students perceptions of cheating and plagiarism in higher institutions. *Journal of College Teaching & Learning* (TLC), 7(11), 59–68.

Palazzo, D.J., Lee, Y.-J., Warnakulasooriya, R., Pritchard, D. E. (2010). Patterns, correlates, and reduction of homework copying. *Physical Review Special Topics: Physical Education Research*, 6(1), 010104-1–010104-11.

Park, C. (2003). In other (people's) words: plagiarism by university students – literature and lessons. *Assessment & Evaluation in Higher Education*, 28(5), 471–488.

Riedesel, C., Clear, A., Cross, G., Hughes, J., Simon, Walker, H. (2012). Academic integrity policies in a computing education context. In: *Proceedings of the final reports on innovation and technology in computer science education 2012 working groups*. New York, NY, USA: ACM, 1–15.

Roberts, E. (2002). Strategies for promoting academic integrity in CS courses. *Frontiers in Education*, 3, F3G14–19.

Roig, M. (2009). Avoiding plagiarism, self-plagiarism, and other questionable writing practices: a guide to ethical writing. United States Department of Health & Human Services. Office of Research Integrity. `http://ori.hhs.gov/sites/default/files/plagiarism.pdf`

Rolfe, V. (2011). Can Turnitin be used to provide instant formative feedback? *British Journal of Educational Technology*, 42(4), 701–710.

Sabin, R., Sabin, E. (1994, March). Collaborative learning in an introductory computer science course. *SIGCSE Bulletin*, 26(1), 304–308.

Sheard, J., Carbone, A., Dick, M. (2003). Determination of factors which impact on IT students' propensity to cheat. In: *ACE '03: Proceedings of the fifth Australasian Conference on Computing Education*. Australian Computer Society, Inc., 119–126.

Sheard, J., Dick, M. (2011). Computing student practices of cheating and plagiarism: a decade of change. In: *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*. New York, NY, USA: ACM, 233–237.

Sheard, J., Markham, S., Dick, M. (2003). Investigating differences in cheating behaviours of IT undergraduate and graduate students: the maturity and motivation factors. *Higher Education Research & Development*, 22(1), 91–108.

Simon, Cook, B., Sheard, J., Carbone, A., Johnson, C. (2013). Academic integrity: differences between computing assessments and essays. In: *Proceedings of the 13th Koli Calling international conference on computing education research*. New York, NY, USA: ACM, 23–32.

Trees, A., Jackson, M. (2007). The learning environment in clicker classrooms: student processes of learning and involvement in large university-level courses using student response systems. *Learning, Media and Technology*, 32(1), 21–40.

Waite, W.M., Jackson, M.H., Diwan, A., Leonardi, P.M. (2004). Student culture vs group work in computer science. *SIGCSE Bulletin*, 36(1), 12–16.

Weber, L., McBee, J., Krebs, J. (1983). Take home tests: an experimental study. *Research in Higher Education*, 18(4), 473–483.

Wilkinson, J. (2009). Staff and student perceptions of plagiarism and cheating. *International Journal of Teaching and Learning in Higher Education*, 20(2), 98–105.

Wood, W.B. (2004). Clickers: a teaching gimmick that works. *Developmental Cell*, 7(6), 796–798.

Yeo, S. (2007). First-year university science and engineering students' understanding of plagiarism. *Higher Education Research & Development*, 26(2), 199–216.

R. Fraser obtained a B.Sc. Honours in Environmental Science from Carleton University and a B.Sc. in Cognitive Systems, Computational Intelligence and Design from the University of British Columbia. He completed a M.Sc. in Computing at Queen's University at Kingston and the Ph.D. in Computer Science from the University of Waterloo. While at Waterloo, he completed the Certificate in University Teaching program. He is currently a postdoctoral fellow in the Computer Science department at the University of Manitoba and primarily conducts research in the area of algorithms and complexity, with an emphasis on computational geometry and approximation algorithms.

Neteisėtas bendradarbiavimas ir plagijavimas informatikos kursiniuose darbuose

Robert FRASER

Straipsnyje apžvelgiama akademinio nesąžiningumo pobūdis informatikos kursiniuose darbuose. Nagrinėjamas įvairių bendradarbiavimo priemonių veiksmingumas ugdyme, taip pat aprašomos pagrindinės priežastys ir pristatomos skirtingos strategijos studentų nesąžiningam elgesiui mažinti rašant informatikos kursinius darbus. Informatikos kursinis darbas yra unikalus tuo, kad dažnai egzistuoja standartiniai uždavinių sprendimai ir atliktas darbas gali būti labai lengvai paviešintas ar nukopijuotas. Todėl straipsnyje apibūdinama išskirtinė neteisėto bendradarbiavimo ir plagijavimo prigimtis, kaip ji suvokiama studentų, dėstytojų ir administracijos. Išnagrinėjus keletą pagrindinių nesąžiningo studentų elgesio priežasčių, pristatoma keletas metodų, kaip sumažinti netinkamą elgesį. Taip pat pateikiama keletas idėjų kompiuterių mokslo kursui tobulinti. Siūloma pabrėžti kiekvienos užduoties mokymosi rezultatus, organizuoti mokymo sesijas ir taip skatinti tinkamą bendradarbiavimą, po kiekvienos atliktos užduoties siųsti apklausas, susijusias su užduočių turiniu, apibrėžti aiškias ribas tarp tinkamo ir netinkamo bendradarbiavimo kiekviename kurse. Nors ši problema yra nagrinėjama informatikos bendruomenėje, dauguma dalykų gali būti pritaikyta ir kitoms disciplinoms – inžinerijai, matematikai ir pan.

CEEOL copyright 2024