Source Code Plagiarism—A Student Perspective

Mike Joy, Georgina Cosma, Jane Yin-Kim Yau, and Jane Sinclair

***Abstract*—This paper considers the problem of source code plagiarism by students within the computing disciplines and reports the results of a survey of students in Computing departments in 18 institutions in the U.K. This survey was designed to investigate how well students understand the concept of source code plagiarism and to discover what, if any, specific aspects might cause particular confusion. An analysis of the results was carried out to assess understanding by topic and to discover whether various demographic factors may have an influence on that understanding. Within the survey sample, it appeared that the demographic factors tested did not generally affect students' understanding of source code plagiarism. However, analysis of the data for specific topics revealed that there are several areas of activity where the boundary between acceptable and unacceptable behavior is not clearly understood. These findings have implications for plagiarism education programs.**

***Index Terms*—Computing disciplines, plagiarism, source code, student survey, university education.**

I. Introduction

THE problem of students plagiarizing is ongoing within educational institutions and is not confined to the submission of essays and other text-based assignments; it is also an issue within the computing disciplines, where students must write program code that is assessed for correctness and quality. The issue of plagiarism within essays has been addressed at a national level in the U.K.; the JISC Plagiarism Advisory Service [1], which contains both practical advice together with the Turnitin tool, is widely regarded as having made a significant impact on the problem, but the service does not relate to the discipline specific issue of code reuse.

Much work has been done on identifying similarities between code fragments to find instances of plagiarism. Algorithms and tools have been developed for this purpose, including JPlag [2], MOSS [3], and Sherlock [4], which are all effective at accurately identifying similarities in program code. It is generally assumed (albeit with little empirical supporting evidence) that their use serves as a deterrent to students. However, the following question is not addressed in the literature: "Do students understand what source code plagiarism means?"

University courses normally contain material that advises students about plagiarism for essay-based assignments, the correct ways to reference other people's work, and the consequences for a student discovered behaving in an academically "inappropriate" manner. Computing courses may also contain similar guidance relating to reuse of code within programming assignments. There is, however, a "gray area"—there are instances of code similarity that may or may not be considered as plagiarism, depending on the context. A recent survey of computing academics by Cosma and Joy [5] suggests that this confusion is particularly evident in the following circumstances.

- The student is required to write "object-oriented" code. There is a conflict here between acknowledging reuse and "reinventing the wheel" since the templates that students use may be nontrivial.
- Reuse of code that has in part already been submitted for previous assignments.

That survey was based on teachers' perceptions of plagiarism and did not consider how students view the problem. What may be seen as academic misconduct by a teacher may be viewed as legitimate code reuse by a student (for reasons that may be clearly articulated), and vice versa.

In order to examine the student perspective, students were surveyed from computing departments in institutions throughout the U.K. In this paper, the responses to that survey are analyzed in order to find out what students consider to be plagiaristic activities. These results identify activities that are indeed plagiarism, but which are generally not perceived as such by students.

II. Literature Review

Students can obtain source code from various sources including the Internet, source code banks, and textbooks. A survey by Nadelson gathered the perceptions of 72 academics and reported 570 incidents of suspected misconduct by undergraduate and graduate students [6]. The majority of those incidents were "unintentional plagiarism," 134 of which involved undergraduate students and 39 involved graduate students. In addition, many incidents were reported where academics suspected that students had submitted papers copied from the Internet, or which concerned "purposeful plagiarism." Other forms of academic misconduct reported were "class test cheating" and "take-home test cheating."

Studies show an increase in plagiarism, although there is some debate as to how much this is due to a greater amount of plagiaristic activity rather than better detection by modern plagiarism detection tools [7], [8]. One factor that makes plagiarism easier for students, and is thus thought to influence this increase, is the wealth of online resources that exists. For example, students can hire expert coders working for online businesses to implement their programming assignments [9]. A number of studies have expressed concerns about the ease with which students can obtain material from online sources and use the material in their student work [10], [11].

Manuscript received November 17, 2008; revised June 18, 2009 and November 27, 2009; accepted February 05, 2010. Date of publication April 19, 2010; date of current version February 02, 2011.

M. Joy, J. Y.-K. Yau, and J. Sinclair are with the Computer Science Department, University of Warwick, Coventry CV4 7AL, U.K. (e-mail: M.S.Joy@warwick.ac.uk).

G. Cosma is with P.A. College, Larnaca, Cyprus.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2010.2046664

Various techniques used by students to cheat, and reasons why they cheat, have been described by Dick *et al.* [12] and Sheard *et al.* [13]. A recent survey by Culwin *et al.* of academics in U.K. university Computing departments found that half of the 293 academics who responded considered that incidences of plagiarism are increasing [14]. Furthermore, in that survey, estimates of the proportion of students who plagiarize source code in introductory programming courses ranged from 20% to over 50%. There have also been investigations into the impact of plagiarism on the learning and teaching process and into techniques for preventing and for dealing with plagiarism [15], [16].

Much work has been done on why students plagiarize, and a number of motivating factors have been identified. Such factors include: inadequate time management, workload, laziness, not understanding what constitutes plagiarism, fear of failure, high achievement expectations, cryptomnesia, thrill of breaking the rules and risk of getting caught, work ethics, competitive achievement, low self-esteem, time pressure, and a desire to increase their marks [17]–[20].

Lack of understanding is one specific area that is useful to explore since this may help institutions improve and target the antiplagiarism guidance they provide. Marshall and Garry [21] conducted a survey in which 181 students were presented with small scenarios that described issues of copyright and plagiarism. Participants were required to rate the seriousness of the behavior and how that behavior would be regarded by other students, the University, and the general public. They were also asked to indicate whether they had engaged in similar activity. The study found that students generally have a poor understanding of what plagiarism is and of the actions that constitute plagiarism. Most students (94%) correctly identified scenarios describing obvious plagiarism, but did less well with scenarios addressing plagiarism of secondary sources, plagiarism of the form of a source, and paraphrasing, where 27%, 58%, and 62% of students correctly identified the scenario as plagiarism, respectively [21].

Marshall and Garry's findings address commonly misunderstood aspects of textual plagiarism, relating mainly to copying words and failing to attribute text correctly. In computing courses, students must understand plagiarism in the context of programming assignments. Where source code plagiarism is addressed in the literature, it is often considered with respect to tools and detection [2]–[4], [22]. Culwin *et al.* [14] explore the topic from the perspective of academic staff, investigating how actively (and with what degree of automation) departments check for source code plagiarism. Cosma and Joy [5] consider potential misunderstandings concerning source code plagiarism from the viewpoint of staff. It appears that there is currently no work examining how well students themselves understand source code plagiarism. The current study seeks to explore this issue.

III. Methodology

To investigate students' understanding of source code plagiarism, a Web-based survey was set up, consisting of six demographic questions and 15 questions on source code plagiarism. From the existing literature, six topics were identified, five of which represented general areas of particular relevance to source code plagiarism. A final category was added to include other academic cheating that may be confused with plagiarism. The six topics were the following:

1: Self-plagiarism and source code reuse;
2: Copying text from books and online sources;
3: Stealing or paying other people to produce work;
4: Inappropriate collaboration;
5: Converting code to another programming language;
6: Falsification as opposed to plagiarism.

Each of the 15 plagiarism-related questions was categorized according to the topic it addressed. Each question presented a small scenario describing ways students have obtained, used, and referenced source code. For each scenario, the test subject was asked to identify whether the behavior described was plagiaristic, with the possible responses being the following:

a. Yes, definitely;
b. I think it is, but I am not completely sure;
c. I don't know;
d. I think it isn't, but I am not completely sure;
e. No, definitely not.

Each scenario was scrutinized by at least four academics with substantial experience in the detection and management of plagiarism in universities and was deemed to be either "definitely" plagiarism (response a) or "definitely not" (response e). The questions were presented to the students in a randomized order. The questionnaire was advertised to students in Computing departments across the U.K., and although respondents were not required to identify their institution, 77% did so, and responses were received from at least 18 institutions in the U.K. and three elsewhere in Europe. In addition to the questions directly relating to plagiarism, basic demographic information was sought to establish the subject and type of course being studied, the student's area of origin, and whether they had been informed about plagiarism and felt they understood the issues. The questionnaire was conducted anonymously in order to minimize the risk of false responses. A facility was included for students to provide further information if they wished, either anonymously or otherwise, but in order to maximize the response rate, completion of this supplementary information was not a requirement.

The number of responses received (770) was sufficient for detailed statistical analyses on the data to be performed. In order to analyze the data, the correct answer to each of the scenarios was coded as 1 mark, and the opposite answer as -1. Responses b and d were assigned a grade of 0.5 or -0.5 based on the correct answer. For example, if the correct answer is a, then response b would be coded 0.5 marks and response d -0.5 marks. Response c was worth 0 in all cases.

IV. Overall Summary

A. Student Backgrounds

More than half of the respondents, 53.2% (410), were undergraduate students undertaking a B.Sc. degree in a computing subject, 20.6% (159) of the respondents were M.Sc. students undertaking a taught M.Sc. degree in a computing subject, 16% (123) were studying joint B.Sc. degrees in computing with another subject, 4.7% (36) were doctoral students in a computing

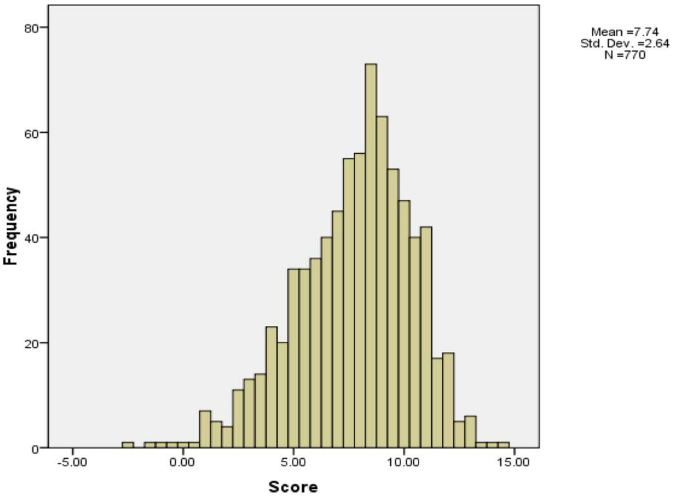

Fig. 1. Frequencies of overall scores.

discipline, and the remaining 42 respondents were studying another subject.

Almost all of the respondents were from either the U.K. or the European Union (702), with the next largest grouping, Asia, containing just 37. The origins of the remaining 31 respondents included Africa (7), North America (5), the Middle East (4), and Australasia (1).

A substantial number of respondents (23%) chose not to provide their institution name. Those respondents who did name their institution were from 18 different U.K. universities and three overseas universities. Of the 18 U.K. universities, 13 were "pre-1992" universities (that is, established universities incorporated before 1992) and accounted for 68% of the respondents, while five were "post-1992" (polytechnics or other similar higher education institutions awarded university status in 1992 or later) and represented 9%. Overseas responses accounted for less than 1%.

B. *Overall Scores*

As described, the mark awarded for each question was in the range $\{1, 0.5, 0, -0.5. -1\}$. The coding allowed the calculation of a "total mark" for each respondent on a scale from -15 to $+15$, where a score of $+15$ indicates that they answered each question correctly and consequently appear to understand all the plagiarism issues addressed. The vast majority of students (502) received a score between 5.5 and 10 marks, 131 respondents scored 10.5 and above, and 137 scored 5.5 or less, as illustrated in Fig. 1. The fact that only 17% of students scored above 10.5 suggests that many students have an incomplete understanding of what constitutes plagiarism in programming assignments. Section V describes the results gathered after applying various statistical tests to the responses with the aim of revealing patterns in the data. More specifically, the responses of students were analyzed to uncover patterns between scores and differences in area of origin, differences in degree programs, and differences in institutions. Such information would reveal whether knowledge of what constitutes plagiarism is influenced by the students' background factors. For example, do postgraduate students have a better understanding than undergraduates?

TABLE I
SUMMARY OF SCORES BY STUDENT ORIGIN

Area of Origin	Mean	N	SD
Unspecified	11.7500	2	.35355
Africa	4.0000	7	3.73050
Asia	6.0676	37	2.70843
Australasia	11.0000	1	n/a
EU	7.3563	87	2.75966
Middle East	8.1250	4	1.03078
North America	9.9000	5	1.63554
South America	10.2500	2	.35355
UK	7.9106	615	2.54682
Other Country	6.6000	10	1.96921
Total	7.7409	770	2.63974

V. Demographic Analyses

A. *Different Areas of Origin*

Some authors have suggested that students from different countries may perceive plagiarism in different ways. Each respondent was asked to identify their region of origin, not as a specific country (so as to ensure that the respondents would be confident the survey was genuinely anonymous), but by their status as a student (that is: home, EU, or other) and, for the third group, the area from which they come. The average scores by area are presented in Table I. The majority, 615, were U.K. students, with a further 87 from the EU. The only other substantial group was Asia, from which 37 students responded. Because the remaining groups consisted of 10 respondents or fewer, it was not appropriate to perform statistical tests to investigate the differences in student performance (scores) between those groups.

As the sample was derived from an open and anonymous survey, it was not possible to guarantee an even spread in any of the demographic categories. This is particularly true for area of origin since the survey was conducted in the U.K., and it is therefore unsurprising that the participants were mostly from the U.K. Four major groups were considered—U.K., EU, Asian, and all other countries. The choice of geographical groups was pragmatic, providing groups of sufficient size to allow comparison of mean scores using a *t*-test. No significant differences were found between the mean scores of the U.K. and EU groups ($t = 1.77$, df $= 107.75$, $p = 0.08$) and the U.K. and all other countries ($t = 0.77$, df $= 31.87$, $p = 0.45$).

However, there was a statistically significant difference between the U.K. and Asian groups ($t = 4.03$, df $= 39.93$, $p < 0.001$). When the two groups' responses were compared for each individual question, the differences in means for those questions were found not to be significant. However, for 14 of the 15 questions, the mean score for the U.K. group was higher than that for the Asian group, suggesting that there may be a general perceptual difference between the two groupings, but that the difference is not focused on a single aspect of plagiarism. The basic data gathered in this survey cannot support any

TABLE II
SUMMARY OF SCORES BY DEGREE COURSE

Degree	Mean	N	SD
Computing with a science	7.1429	63	2.79173
Computing with Business	7.1000	60	3.11339
Computing B.Sc.	7.7524	410	2.59354
Computing M.Sc.	8.0629	159	2.46799
Computing Ph.D.	7.7778	36	2.63613
Other	8.1905	42	2.58738
Total	7.7409	770	2.63974

strong conclusions on the relationship between results and demographic factors. However, it may provide an indication of useful areas to explore in the future.

B. *Differences in Programs*

Further anecdotal evidence suggested that there might be differences depending on the type of program in which the student was enrolled. For example, a student studying for a joint Computing and Business degree who writes many essays may be more sensitive to plagiarism than a student of Computer Science alone. Furthermore, a graduate student might be expected to be more aware of the issues than an undergraduate.

The respondents were divided according to their stated type of degree course (summarized in Table II), and a pair-wise comparison of means was carried out using a *t*-test. None of the comparisons yielded a significant difference, and a hypothesis that any of the groups is more aware of plagiarism issues than any other cannot therefore be supported.

C. *Differences in Institutions*

Whether respondents from a particular institution (or type of institution) performed differently to the respondents as a whole would suggest possible examples of "good practice" in educating students about plagiarism. Although there is a range of means, the unusually high or low values relate to institutions from which very few students responded, and none of the means is significantly different from the overall mean.

VI. Plagiarism Issues

For the purpose of this analysis, the responses were divided into two groups—those students who scored 10 or above overall (group A) and those who scored less than 10 (group B). Group A thus contains the (178) students who seem to have a good understanding of what constitutes source code plagiarism, while group B consists of those who appear to understand less well. The choice of 10 as a cutoff point selects the "top quartile" of students. The first analysis considered the data relating to student understanding and education, relating this to how students performed overall. The second analysis considered results for each of the six topics.

A. *Student Understanding*

Respondents were asked to state: (a) whether they had been informed by their institution about plagiarism (96.7% agreed); (b) whether they felt they understood the issues (98.3% agreed); and (c) whether they understood the penalties if caught (92% said they did). The 25 who claimed not to have been informed and the 13 who admitted to not understanding originated from several universities. These 36 respondents (of whom only two both did not understand and claimed not to have been informed) represent 4.8% of the sample. From this relatively small percentage, it can be inferred that universities are actively and successfully sensitizing the majority of students to plagiarism and attempting to emphasize to students the academic incorrectness and inappropriateness of plagiarism and the penalties for committing the offense.

It is interesting to note, however, that in the case of one internationally renowned institution, 11 students claimed not to have been informed (although 62 other students from the same department disagreed), but those 11 students all said they understood. Possible reasons why a student may not have been informed of plagiarism may include nonattendance at the lecture on plagiarism (if the lecture was given as part of their course) and consequent failure to have read the relevant documentation (e.g., section on plagiarism in a handbook) or even the student forgetting their having received that information.

For each of questions (a) and (b), the differences between the means of group A (students with a generally good understanding) and group B (students with weaker understanding) were not significant, but this was not the case for (c), suggesting that awareness of the consequences of being caught may be a significant factor affecting a student's motivation to understand what constitutes plagiarism.

B. *Students' Understanding of Source Code Plagiarism Topics*

The scenarios that formed the basis of the survey's plagiarism questions were grouped according to the six topics presented in Section III. The first five topics relate to aspects of source code plagiarism, and each was explored by several questions. Understanding of a topic was thus checked across two or more scenarios. One question was included to address topic 6 (distinguishing plagiarism from other cheating). The responses were analyzed by topic as described below. In addition, for each topic the general performance of group-A students was compared to that of group-B students to see whether the degree to which any given topic was understood was in line with students' overall understanding of source code plagiarism. For each question, a *t*-test was performed on the mean scores for the two groups to test the hypothesis that there is a significant statistical difference between them. For a given question, a significant difference between the two means suggests that the issue addressed is a significant factor in student understanding of plagiarism in general. With one exception (discussed below), the hypotheses were supported.

1) Topic 1: Self-Plagiarism and Source Code Reuse: Two questions (referred to here as 1a and 1b) addressed this topic, describing two scenarios in which a student copies source code from an assignment that was previously submitted for academic credit and incorporates that code into a new submission. The hypothesis is that self-plagiarism is an activity that many students would not recognize as being plagiarism. The essential difference between the two scenarios was that in 1a the student "acknowledges (reuse of his previous work) in the program as a comment," whereas in 1b the student simply "incorporates (the code) into his project and submits it." For question 1a, 89.8% of respondents answered correctly that plagiarism had definitely not taken place; however for question 1b, a clear instance of self-plagiarism, only 6.8% thought either that plagiarism had definitely taken place or were unsure.

There is also a significant negative correlation ($t = -0.36$, $p < 0.001$, $n = 755$) between the answers to the two questions, so that most students who answered 1a correctly got 1b incorrect. In other words, the respondents consistently thought that neither scenario represented plagiarism.

The responses to question 1b suggest that students perceive that copying work that they have previously submitted for academic credit—with or without referencing—is acceptable behavior and so support the hypothesis.

2) Topic 2: Copying Code From Books and Other Sources: This topic was covered by five scenarios (2a–e) in which a student (or group of students) copies code from a book into their assignment that they then submit. In 2a, a student submits the assignment "without noting that he obtained the code somewhere else." In 2b, the student acknowledges that their code is sourced from a book, but the student has forgotten which one and their reference is likely to be incorrect. In 2c, a group project is described that is submitted "with all references complete." The scenario in 2d refers to a student who consults two textbooks just to "read for ideas" and "gain inspiration," but completes their program alone. Finally, the scenario in 2e is a group project in which the students "note in the program in the form of comments the reuse of code, and in their documentation the reuse of classes." It was hypothesized that 2a would be understood as plagiarism, and that 2c, 2d, and 2e would be understood as correct behavior. Scenario 2b is technically plagiarism, but one for which it *might* be argued there are mitigating factors.

Almost all (93.9%) identified 2a as plagiarism (with 81.3% being definite and a further 12.6% saying they were not completely sure), 98.1% of respondents to 2c correctly identified the scenario as representing good behavior, as did 91.9% of respondents to 2e, supporting the hypothesis that the three examples would be clearly understood.

It was also hypothesized that students may be unclear that references must be provided precisely and correctly, and that not doing so is both plagiarism (since the material is used without its true source being identified) and falsification (since there is an attempt to deceive). The responses to 2b show that only 31.3% understood (possibly not clearly) that including false references, even if the inclusion was unintentional, is plagiarism, and over half (53.5%) believed that the conduct of the student in the scenario was not plagiarism, and the responses therefore support the hypothesis.

A further hypothesis was that the distinction between "copying" and "consulting" may be unclear. Question 2d addresses the distinction, and only 76.6% agreed that the student was acting appropriately. Furthermore, the difference between the means for groups A and B for this question was not significant ($t = 1.65$, $\mathrm{df} = 312.2$, $p = 0.101$), suggesting that the boundary between "using code" and "getting ideas" is not well understood and is independent of a student's more general understanding of plagiarism issues.

3) Topic 3: Copying From Another Student: Three scenarios (3a, 3b, 3c) were used for this topic. In 3a, a student copies and modifies code from an unattended terminal. In 3b, the student pays a student in the year above him to author part of his program, and in 3c the student copies the contents of an uncollected printout found near a printer.

For questions 3c (96.9%) and 3a (92.4%), the responses clearly indicated that students correctly identified plagiarism as having taken place. For 3b, however, the positive response rate of 85.6% is not so clear cut, with over 10% of the respondents considering it not to be a case of plagiarism. A possible explanation for the discrepancy may be a perception that, albeit cheating, the scenario might not fit exactly the description of plagiarism as presented in the survey.

4) Topic 4: Inappropriate Collaboration: For some assignments, the assessment may be on an "individual" basis, in which case working together with friends to author the submission is cheating. If such collaboration is not acknowledged, submission of a jointly authored assignment counts as plagiarism. Two scenarios (4a, 4b) were used. In the first (4a), two students on the same module collude on an individual assignment and submit very similar pieces of work. In 4b, in a group assignment, two students assigned to different groups exchange parts of their work and include them in their own groups' submissions.

The responses to the two questions were unexpected. For 4a, 62.4% identified the activity as plagiarism (28.4% were unsure), and for 4b, 61.6% made a positive identification (30.7% unsure). These figures suggest that students misunderstand the boundaries between formative discussion and inappropriate collusion between students and strongly suggest that students believe that working together with fellow students is acceptable even when informed otherwise.

5) Topic 5: Converting to Another Programming Language: This topic concerned translating code found elsewhere written in a different programming language to the language required in an assignment and submitted as the student's own work. Two examples (5a, 5b) were used. The first (5a) relates to a student writing a C++ program who finds equivalent code in a Java textbook, converts this to C++, and submits the program without reference to the textbook. In the second scenario (5b), the student converts part of a Visual Basic program written for a high school project to their first-year Java program, but notes this fact in the Java program "in the form of a comment."

Less than half the respondents (48.6%) recognized 5a as plagiarism, whereas 96.9% understood that 5b was not. The first figure suggests that converting code to a different language is another gray area. The high number of correct responses to 5b is not necessarily out of line with this since the answers may be influenced by the explicit mention of a reference in the program's comments and may reflect an attitude of "if it's referenced, it can't be plagiarism" rather than an understanding of the specific topic. This suggests that an approach based on "if in doubt, reference!" is one that students will appreciate and may be an effective way of encouraging students to adopt good referencing habits.

6) Topic 6: Falsification as Opposed to Plagiarism: One question was included that relates to a cheating scenario that does not fit any likely definition of plagiarism. In this question, the student fails to get a program to compile and run and just "modifies the program output to make it produce the wanted output, even though the rest of the program was not functioning as the assignment requires it to" and then submits this work. This scenario was correctly identified as not plagiarism by 89.2% of the respondents, clearly indicating a good awareness of the difference between cheating in general and the specific offense of plagiarism.

C. Supplementary Information

Few students chose to provide further information, although one student highlighted the issue of code licensing and commented that "some (generous) people make their code freely available even in books, in which case, I wouldn't see any problem in reusing modules/functions." This view suggests that the distinction between plagiarism and copyright infringement (between acknowledgment and reuse) is unclear.

A further issue identified was whether the use of library functions or macros, which are simply invoked by a programmer, require referencing.

VII. Discussion

A survey such as this is a snapshot in time, and an individual's understanding of source code plagiarism is difficult to measure accurately. Most students probably understand the basic issues quite well and the effort put into educating students about academic integrity in many institutions suggests that this understanding should not be a surprise and should be supported by the statistics. Indeed, as was identified in Section VI-A, most respondents considered that the issues had been understood. However, this does not necessarily mean that all aspects are understood equally well. Earlier work suggests that there are "gray areas" in teachers' understanding, and it is reasonable to expect that this is also the case from the students' viewpoint [5]. This expectation is confirmed by the current study.

The first generally misunderstood area is that of reusing one's own code from a previous assignment, and a scenario that clearly represented this self-plagiarism was recognized as such by only 7% of respondents. It may be that some students thought an offense other than plagiarism was involved in this case. However, the finding that reuse of one's own previously submitted work may be thought of as acceptable agrees with the results of Cosma and Joy relating to staff [5] and to those of Marshall and Garry addressing text-based plagiarism [21]. Although there may be debate as to which sort of academic offense self-plagiarism represents, 18.6% of staff thought that it was not an offense at all or did not know if it was [5]. Bretag and Carapiet [23] discovered that 60% of academics in a small random sample had self-plagiarized in their own published work. The fact that staff are unclear about this issue is likely to compound the problem for students since mixed messages may be received. It is hard to capture a succinct, all-encompassing, and understandable definition of plagiarism, and the formal definitions provided by institutions may differ in how they are presented, what they appear to include, and how they might be interpreted [15]. A standardization of definitions between institutions would have the obvious advantage of ensuring that students (and staff) always receive the same message.

The second gray area noted was that students were unclear that providing false references constituted plagiarism. Again, some students may have thought it was a different type of offense. The question of intent may also have influenced the results since providing a misleading reference or forgetting to reference may not be a deliberate attempt to pass off work as one's own, but intention is difficult to establish. Guidance to students on plagiarism could usefully reinforce the importance of accurate and complete referencing.

Nearly 25% of students surveyed thought that reading a book for background information (but completing an assignment without the aid of the book) constituted an offense. Although it is perhaps better to err on the side of caution, these responses reveal that students are not clear where the boundaries lie. It may be felt that seeing plagiarism where there is none is better than failing to spot it when it exists, but the "if in doubt, reference!" approach becomes impractical if students really believe that everything ever seen should be included.

Considerable confusion appears to exist concerning collaboration. Again, some students may have regarded collusion as being an offense other than plagiarism, but it appears that genuine misunderstanding exists over when joint working is appropriate. This is an area that may be more apparent in programming than in text-based assignments. Code development is often an assessed group activity, in which it will be natural for students to discuss work and collaborate.

More than half of the respondents thought it acceptable to take a program from a book and convert it to another language without acknowledgment. This result contrasts with 94% of students recognizing that straight copying of code (in the same language) is plagiarism. The rationale may be that translation is seen as having the student's own "stamp" and own intellectual effort placed on it. The fact that the original intellectual contribution of the source is not referenced appears somehow to be excused by the student's act of translation. This relates to the concept in text-based plagiarism that the form of an argument (not just exact words) should be acknowledged.

The gray areas outlined above show the topics that caused greatest confusion. The results did not indicate general levels of misunderstanding about basic plagiarism issues to be as high as some studies have found. For example, Marshall and Garry found that 17% of students sampled thought that copying words with an acknowledgment was plagiarism [21]. Even so, figures in the current study that have been tacitly passed over as unremarkable (such as 6% of respondents not understanding that copying a program without reference to its source is plagiarism) represent a considerable number of students who may err through ignorance.

The survey was anonymous, so there is no way of validating the data obtained. However, the responses to questions that students were expected to understand clearly (such as 2a and 2c) are consistent with expectations, and responses to questions with similar content (such as 4a and 4b) are also consistent, so there is no reason to believe that respondents deliberately answered incorrectly.

In general, the data relating to student background did not provide evidence for any particular influential factor. Students from different degrees and different institutions exhibited broadly similar knowledge. The only result of note was that the group who classed themselves as home students may perceive source code plagiarism differently than those students within the Asian group. Although some authors [24], [25] have suggested that cultural factors have an influence on students' understanding of plagiarism and on their likelihood of offending, this is a very difficult area to unpick. While some report that non-Western students are often regarded as being more inclined to plagiarize [25], other studies have shown a more complex picture—for example, non-Western students who had just arrived in the U.K. were found to be *less* likely to plagiarize than those who had previously studied in the U.K. [27]. Although it is difficult to isolate cultural influences, studies such as Hayes and Introna [26] have attempted to develop understanding of this area in order to give better support to students from different backgrounds. One issue generally raised [21], [24], [26], [28] is that studying in a language in which one is not fluent may lead to a greater tendency to plagiarize or to the easier detection of plagiarized work. It is possible that this is not such a differentiating factor in source code plagiarism since the programming language is new to all students and the challenges of writing good programs are equal whatever a student's native language. More work is needed to explore this.

VIII. Conclusion

Approaches to plagiarism detection and prevention outside of the computing disciplines have sometimes divided plagiarizing students into three groups: those who plagiarize deliberately, those who do so negligently, and those who genuinely do not understand what plagiarism means [8]. This study examines the third category with respect to source code plagiarism. Several commonly misunderstood issues are identified, including the following:

- reuse of source code from previous assignments;
- the accuracy of references cited;
- the limits of collaboration;
- the need to reference when code has been taken and converted to a different language.

The findings agree with other studies (generally of text-based plagiarism) that, as well as detection and enforcement of appropriate penalties, it is also necessary to look beyond to understand more fully the reasons why plagiarism is so widespread. Most, if not all, U.K. Higher Education institutions inform students about plagiarism. This study indicates that for students writing program code, the training is not fully effective in raising awareness of what constitutes plagiarism.

It is also interesting to compare the student view with the staff perception as reported in [5], and to note that the only significant common issue is that of reuse of code—students did not highlight object-oriented code as a potential source of confusion.

Barrett and Malcolm [27] stress that telling students is not enough: Plagiarism education must become an active part of the student experience, preferably by embedding it within the work that students routinely carry out for assessment. This has implications for how assessment is conducted, and while it may be a more effective solution, it is not simple or low-cost.

It is interesting to note that some areas of confusion in students' minds have generated much debate, such as self-plagiarism and differing definitions of plagiarism between institutions. Staff may not be clear about the boundaries and may, in their own practice, not live up to the ideals presented to students. The potential for mixed messages is enormous.

Plagiarism is a complex topic, and the boundary between useful behavior that can help a student's educational development (such as collaboration with other students or "patch writing" from trusted sources as a stepping stone to developing independent skills) and behavior that will be punished as "cheating" can cause misunderstanding. There is certainly room for more research into understanding motivations for plagiarism and how best to support students from different backgrounds. In the short term, rules and policies exist, and students who offend will be punished. The very least a student might expect would be a clear, consistent, and simple statement of the rules and a consistency of practice and message around them. However, given the diversity that currently exists in and between institutions, even this basic requirement is not as easy to meet as it sounds. The findings reported here identify a number of issues related to source code plagiarism that students find confusing and will therefore help teachers provide effective support for students to understand (and avoid) this type of plagiarism.

Acknowledgment

The authors are grateful to the representatives of the Higher Education Academy Subject Network for Information and Computer Sciences for circulating the questionnaire and to the students who took the time and trouble to respond.

References

[1] JISC Plagiarism Advisory Service, JISC, 2008 [Online]. Available: http://jiscpas.ac.uk/

[2] L. Prechelt, G. Malpohl, and M. Philippsen, "Finding plagiarisms among a set of programs with JPlag," *J. Universal Comput. Sci.*, vol. 8, no. 11, pp. 1016–1038, 2002.

[3] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, San Diego, CA, 2003, pp. 76–85.

[4] M. Joy and M. Luck, "Plagiarism in programming assignments," *IEEE Trans. Educ.*, vol. 42, no. 1, pp. 129–133, Feb. 1999.

[5] G. Cosma and M. Joy, "Towards a definition of source code plagiarism," *IEEE Trans. Educ.*, vol. 51, no. 2, pp. 195–200, May 2008.

[6] S. Nadelson, "Academic misconduct by university students: Faculty perceptions and responses," *Plagiary*, vol. 2, no. 2, pp. 1–10, 2007.

[7] P. Larkham and S. Manns, "Plagiarism and its treatment in higher education," *J. Further Higher Educ.*, vol. 26, no. 4, pp. 339–349, 2002.

[8] F. Culwin, "I think my students are less naughty, but maybe the tools are more effective?," in *Proc. 2nd Int. Plagiarism Conf.*, Gateshead, U.K., Mar. 2009, pp. 10–13 [Online]. Available: http://www.plagiarismadvice.org/media/2006papers/FintanCulwin.pdf

[9] T. Jenkins and S. Helmore, "Coursework for cash: The threat from online plagiarism," in *Proc. 7th Annu. Conf. HEA Netw. Inf. Comput. Sciences*, Dublin, U.K., 2006, pp. 121–126.

[10] J. Kasprzak and M. Nixon, "Cheating in cyberspace: Maintaining quality in online education," *J. Assoc. Adv. Comput. Educ.*, vol. 12, no. 1, pp. 85–99, 2004.

[11] P. Scanlon and D. Neumann, "Internet plagiarism among college students," *J. College Student Dev.*, vol. 43, no. 3, pp. 374–85, 2002.

[12] M. Dick, J. Sheard, C. Bareiss, J. Carter, T. Harding, D. Joyce, and C. Laxer, "Addressing student cheating: Definitions and solution," *SIGCSE Bull.*, vol. 35, no. 2, pp. 172–184, 2003.

[13] J. Sheard, A. Carbone, and M. Dick, "Determination of factors which impact on it students' propensity to cheat," in *Proc. 5th Australasian Comput. Educ. Conf.*, Adelaide, Australia, 2003, pp. 119–126.

[14] F. Culwin, A. MacLeod, and T. Lancaster, "Source code plagiarism in U.K. H.E. computing schools, issues, attitudes and tools," South Bank University, London, U.K., Tech. Rep. SBU-CISM-01-02, 2001.

[15] S. Dey and A. Sobhan, "Impact of unethical practices of plagiarism on learning, teaching and research in higher education: Some combating strategies," in *7th Int. Conf. Inf. Technol. Based Higher Educ. Train.*, pp. 388–393.

[16] J. Carroll and J. Appleton, "Plagiarism: A good practice guide," 2001 [Online]. Available: http://www.jisc.ac.uk/

[17] J. Beasley, "The impact of technology on plagiarism prevention and detection: Research process automation, a new approach for prevention," in *Proc. Plagiarism: Prevention, Practice Policy Conf.*, Newcastle upon Tyne, U.K., 2004, pp. 23–29.

[18] D. McCabe, L. Trevino, and K. Butterfield, "Cheating in academic institutions: A decade of research," *Ethics Behav.*, vol. 11, no. 3, pp. 219–232, 2001.

[19] M. Bjorklund and C. Wenestam, "Academic cheating: Frequency, methods, and causes," in *Proc. Eur. Educ. Res.*, Lahti, Finland, 1999.

[20] R. Bennett, "Factors associated with student plagiarism in a post-1992 University," *J. Assess. Eval. Higher Educ.*, vol. 3, no. 2, pp. 137–162, 2005.

[21] S. Marshall and M. Garry, "How well do students really understand plagiarism?," in *Proc. ASCILITE Conf.*, Brisbane, Australia, 2005, pp. 457–467.

[22] A. Ahtiainen, S. Surakka, and M. Rahikainen, "Plaggie: GNU-Licensed source code plagiarism detection engine for Java exercises," in *Proc. 6th Baltic Sea Conf. Comput. Educ. Res.*, Koli, Finland, 2007, pp. 141–142.

[23] T. Bretag and S. Carapiet, "A preliminary study to identify the extent of self-plagiarism in Australian academic research," *Plagiary*, vol. 2, no. 5, pp. 1–12, 2007.

[24] G. D. Deckert, "Perspectives on plagiarism from ESL students in hong kong," *J. Second Lang. Writing*, vol. 2, no. 2, pp. 131–148, 1993.

[25] C. Juwah, D. Lal, and A. Beloucif, "Plagiarism: Overcoming the cultural issues associated with plagiarism for international students," 2008 [Online]. Available: http://www.rgu.ac.uk/files/ACF52AC.doc

[26] N. Hayes and L. D. Introna, "Cultural values, plagiarism, and fairness: When plagiarism gets in the way of learning," *Ethics Behav.*, vol. 15, no. 3, pp. 213–231, 2005.

[27] R. Barrett and J. Malcolm, "Embedding plagiarism education in the assessment process," *Int. J. Educ. Integrity*, vol. 2, no. 1, pp. 38–45, 2006.

[28] C. Park, "In other (people's) words: Plagiarism by university students—Literature and lessons," *Assess. Eval. Higher Educ.*, vol. 28, no. 5, pp. 471–488, 2003.

Mike Joy received the B.A. and M.A. degrees in mathematics from Cambridge University, Cambridge, U.K., the M.A. degree in post-compulsory education from the University of Warwick, Coventry, U.K., and the Ph.D. degree in computer science from the University of East Anglia, Norwich, U.K.

He is currently an Associate Professor with the University of Warwick. His research interests focus on educational technology and computer science education.

Dr. Joy is a Chartered Fellow of the British Computer Society and is a Chartered Engineer. He is a Member of the Association for Computing Machinery (ACM) and a Fellow of the Higher Education Academy.

Georgina Cosma received the B.Sc. degree in computer science from Coventry University, Coventry, U.K., and the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K.

She is currently a Lecturer at P.A. College, Larnaca, Cyprus. Her fields of interest include information retrieval and educational technology.

Jane Yin-Kim Yau received the B.Sc. degree in computer science from the University of Leicester, Leicester, U.K., in 2002, and the M.Sc. degree in computer science from the University of Warwick, Coventry, U.K., in 2004. She is currently a doctoral candidate in the Intelligent and Adaptive Systems research group in the Department of Computer Science at the University of Warwick. Her research interests focus on a mobile context-aware framework for managing learning schedules.

Jane Sinclair received the B.A. and M.A. degrees in mathematics and philosophy from the University of Oxford, Oxford, U.K., in 1981 and 1985, respectively, and the Ph.D. degree in computer science from the Open University, Milton Keynes, U.K., in 1997.

She is currently an Associate Professor with the Department of Computer Science, University of Warwick, Coventry, U.K. Her research interests include computer science education, formal methods, and computer security.

Dr. Sinclair is a Fellow of the Higher Education Academy.