Source Code Plagiarism in Computer Science Courses: Facts and Impressions

Bruna Bazaluk[1], **Flavio S Correa da Silva**[1], **Maristela Holanda**[2,3], **Dilma Da Silva**[3]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo
São Paulo – SP, Brazil.

[2]Departamento de Ciência da Computação – Universidade de Brasília
Brasília – DF, Brazil.

[3]Department of Computer Science and Engineering – Texas A&M University
College Station – TX, U.S.A.

bazaluk@ime.usp.br, fcs@usp.br, mholanda@unb.br, dilma@cse.tamu.edu

Abstract. *Plagiarism has been a problem in programming courses at different universities around the world. Understanding why students are cheating on their assignments is an important point in this context. This paper analyses the answers of an anonymous questionnaire applied at a Computer Science undergraduate course of a highly reputed Brazilian University concerning the opinion of the students about cheating and plagiarism on programming assignments. When asked if they had already cheated on an assignment, 23.6% of the respondents said they had, and most of them (85.7%) affirmed the reason was the high level of difficulty. Here we discuss some ethic aspects of these situations.*

1. Introduction

Unethical behavior of students at the university level, particularly related to cheating and plagiarism, is a matter of increasing concern, and recent advancements in web-related technologies and the availability of information online have turned these issues even more pressing, as the growing number of publications devoted to these issues indicate (see e.g. [Cole and Mccabe 1996, Foltỳnek et al. 2019, Aniceto et al. 2021, Sindre and Haugset 2022]).

A broad literature survey on academic plagiarism detection [Foltỳnek et al. 2019] has identified 239 research articles involving different kinds of plagiarism. Specifically for programming courses in universities, source code similarity detection algorithms have been created and empirically tested as tools to detect and monitor plagiarism [Sindre and Haugset 2022, Alzahrani 2022, Karnalim and Chivers 2023].

An important point to consider is why students plagiarize in programming assignments and their perception of plagiarism. Regional studies have been developed e.g. in the US [Park 2003, Klein 2011, Pierce and Zilles 2017, Brown and Rosen 2020], UK [Joy et al. 2011, Joy et al. 2013], Australia [Sheard and Dick 2011, Dick et al. 2008, Sheard and Dick 2012, Simon and Sheard 2015, Simon et al. 2014], Asia [Yu et al. 2020, Hu and Lei 2015] and South Africa [Nwosu and Chukwuere 2020], but there is a lack of information on the perceptions and views of Brazilian students about these issues. In fact, in Brazil it has been more common to focus on how to detect cheating than on why it happens or on the students' opinions.

In order to better understand the problem, most of the initiatives have aimed at faculty opinions and views [Dick et al. 2003]. In rare instances, students' perceptions have been addressed as a primary topic [Joy et al. 2011, Lori G. Power 2009]. This lack of focus on the students' perceptions and opinions seems to be even more pronounced in the case of Brazilian universities.

In this context, the goal of this paper is to show the students' perception about cheating and plagiarism in programming assignments, with a particular focus on students in Brazil [Joy et al. 2011, Lori G. Power 2009] and from the students' point of view. We analyse the answers to a questionnaire asking Computer Science undergraduate students at the University of São Paulo about their opinions on plagiarism and cheating and asking if they have already cheated or plagiarized a programming assignment. Based on this analysis, we present some reflections about the role of academic institutions to nudge the behavior of students towards decisions and attitudes that can prove effective in the long term to strengthen the fabric of a society based on virtuous values.

This paper starts with Section 2, a small discussion about plagiarism and cheating in programming courses around the world and specifically in Brazil. Section 3 presents a framework for ethics. In Section 4 the methodology employed in this work is described. After that, Section 5 presents the results of the students' opinions which were gathered during the study. Section 6 provides a discussion of the results and the roles of educational institutions. Finally, Section 7 presents the limitations of this work, and Section 8 has the conclusions and future work.

2. Plagiarism and Cheating in Programming Courses

Throughout the years, many academics have tried to come up with a definition of plagiarism in general, but only a few focused on programming courses. A broadly accepted characterization of plagiarism in this specific context is as follows:

"Source-code plagiarism in programming assignments can occur when a student reuses [...] source-code authored by someone else and, intentionally or unintentionally, fails to acknowledge it adequately [...], thus submitting it as his/her own work. This involves obtaining [...] the source-code, either with or without the permission of the original author, and reusing [...] source-code produced as part of another assessment (in which academic credit was gained) without adequate acknowledgement [...]." [Cosma and Joy 2008]. In addition to that, as a source-code's essence is its structure, codes that differ only by the variable names or functions orders may also be considered plagiarism [Lancaster and Culwin 2004].

Plagiarism can be contextualised within the broader issue of forgery, which is defined in the Cambridge Dictionary as "*an illegal copy of a document, painting, etc. or the crime of making such illegal copies*". Forgery is usually managed considering three complementary strategies:

1. *Prevention:* how to design a system in such way that authorship cannot be defrauded, e.g. by requesting secure digital signatures to be included in source code that is delivered by students for assignment.
2. *Detection:* how to monitor the behavior of individuals in such way that attempted forgery can be detected, e.g. by using code similarity measures as indicators of potential malicious behavior.

3. *Dissolution:* how to establish value scales in such way that forgeries are no longer socially accepted and valued, at least for the majority of individuals.

Academic institutions have a fundamental role in the development and implementation of these strategies. As a fundamental first step towards the assessment of which strategy can be more pressing and how to implement them, it is required that the perceptions of all stakeholders in academic institutions is well understood. In the next sections we discuss how this understanding can be achieved.

3. Ethics

As hinted in the previous section, source-code plagiarism demands *prevention, detection* and *dissolution*, and specialized technologies for *prevention* and *detection* have been successfully developed.

Students' perceptions and opinions about this issue, as gathered from a sample of students at the University of Sao Paulo, will be shown in the following sections. They indicate that further analysis of the issue is appropriate, possibly based on the framework of Ethics and focusing specifically on *dissolution* of plagiarism.

The generally accepted framework for Ethics starts with the concept of *good life*, meaning a "life worth living" from individual as well as collective viewpoints. Historically, three approaches have been proposed to build a good life [Bartneck et al. 2021]:

1. **Ethics of virtues:** Pursuit of *virtues* such as honesty, courage, compassion, generosity, fidelity, integrity, fairness, self-control, and prudence, based on cultivation of virtuous actions and attitudes.
2. **Ethics of duties:** Fulfillment of *duties* and obedience to laws.
3. **Ethics of consequences:** Consideration of *consequences* of potential actions and attitudes and pursuit of positive balance between positive and negative consequences of actions.

Even though these three approaches are usually considered as complementary, the *Ethics of virtues* stands out as the single approach based on which individual actions and attitudes are guided by principles – the *virtues* – instead of social relations – obedience to laws is typically considered when it can be monitored by third party, and consequences of actions and attitudes are typically considered with respect to effects on third party.

Social relations – including technology mediated relations – are permanently and rapidly evolving, hence stable ethical values to weave the fabric of longstanding cultural and social values must specifically account for an *Ethics of virtues* [Vallor 2016]. From this perspective, educational institutions should provide students with explicit and consistent messages about the cultivation of virtues as a practice at least as important as scholarly and technical development.

As observed in existing research initiatives, universities seem to have failed in conveying such messages, given that students are under permanent peer pressure to pursue high and fail-proof performance [Klein 2011, Dick et al. 2008], and frequently exposed to institutional values indicating consequentialist attitudes as of higher relevance than other ethical stances [Hutton 2006] or ambiguity with respect to the relevance of Ethics in general [Klein 2011, Lathrop and Foss 2000].

As the pursuit of virtues is considered a personal matter beyond the responsibilities of academic institutions, there is a danger of leaving students without clarification about how to behave. We find, for example, the following statement in the specialized literature:

"Students who have a desire to learn or master a particular body of information are less likely to cheat than are students motivated by extrinsic or performance factors, such as academic standing, grades [...]" [Jordan 2001].

4. Methodology

The triggering event for the development of this work was an incident observed by faculty members at the Department of Computer Science, University of Sao Paulo. The University of Sao Paulo is a highly reputed university in Latin America, and the Computer Science major is ranked within the top 80 worldwide according to the latest QS World University Ranking. Nevertheless, during the second semester of 2022 (from August to November), 5 cases of source code plagiarism were detected on graded activities in 4 different subjects, specifically related to (1) parallel programming, (2) databases, (3) distributed systems and computer networks and (4) general programming practices and techniques.

In order to try to understand what happened, student representatives created an anonymous questionnaire for all students to answer. It was sent to all registered students by e-mail and through an informal group using a message app. When the questions were sent there were 273 students registered, 39 female and 234 male; 55 students responded to the questionnaire, that is, approximately 20% of the students.

The survey presented to the students had 7 main questions and 4 optional ones where they could write their opinions. The questions were:

- Have you ever copied (entirely or partially) a programming assignment?
- If you have, what were the reasons for you to do it?
- Which of the following options do you consider cheating or plagiarism (in programming assignments)?
  - Copying a full assignment from someone else.
  - Copying blocks of code found online.
  - Talking to your friends about possible solutions for the assignment.
  - Look at a solution as a starting point to write yours.
  - Others. (Here they could write their own opinion)
- Do you consider cheating (or plagiarizing) on a programming assignment unethical?
- Do you think someone who cheats (or plagiarizes) on a programming assignment should be punished?
- Who should be punished when a programming assignment is copied from someone else?
  - Only the person that copied.
  - Only the person whom the assignment was copied from.
  - Both.
  - It depends on the situation.
  - Nobody.

- What could be the ideal punishment for cheating (or plagiarizing) on a programming assignment?
  - Give the student a 0 on this assignment.
  - Fail the student in this course.
  - Give the student a warning.
  - Start an inquiry process against the student.
  - Expel the student.
  - Others. (Here they could write their own opinion)

For all the "Yes or No" questions there was an additional option: "Depends". For each of these questions there was a space for the students to explain if they had answered "Depends".

5. Results

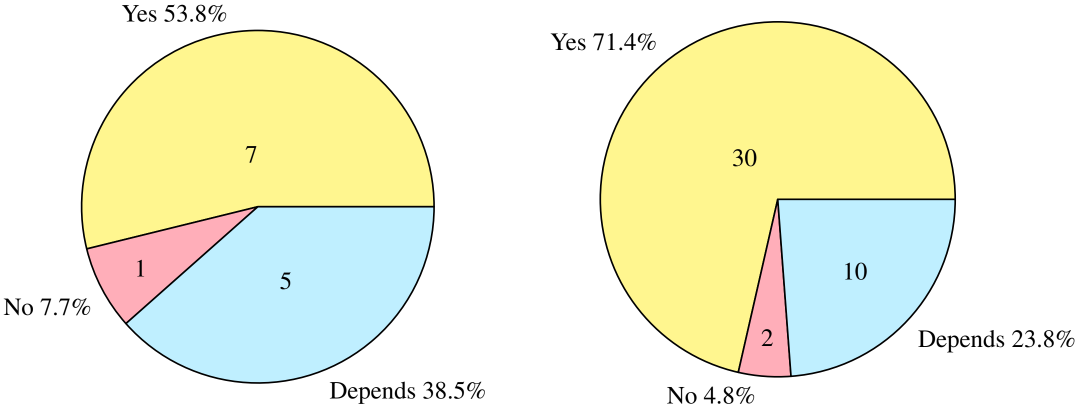
When students were asked if they had already copied a programming assignment, 13 students said they had, that is 23.6% of the respondents; the other 42 students, 76.4%, said they have never done it.

When the students who have cheated or plagiarized were asked why they have not followed the rules, 12 students (85.7%) said it was because of the assignment's high level of difficulty; 10 (71.4%) claimed they lacked time to do it; 2 (14.3%) said they were just lazy; 1 (7.1%) admitted it was easier to copy; 1 (7.1%) said they didn't like the course and were unmotivated to do the assignment (in this question the respondents were able to select more than one answer and add other options if they wanted).

When students were asked about what they considered cheating or plagiarism in programming assignments, the answers were:

- Copying a full assignment from someone else. 51 students (92.7%)
- Copying blocks of code found online. 31 (56.4%)
- Talking to your friends about possible solutions for the assignment. 0
- Look at a solution as a starting point to write yours. 11 (20%)
- Others. 5 (9%)

Figure 1 and Figure 2 show the answers given by the two different groups (the students who have already cheated and the ones who have not) for the two main questions: "Do you consider it unethical to cheat or plagiarize a source-code assignment?" and "Do you think cheating or plagiarizing a source-code assignment should be punished?".



(a) Answers from students who HAVE already cheated.

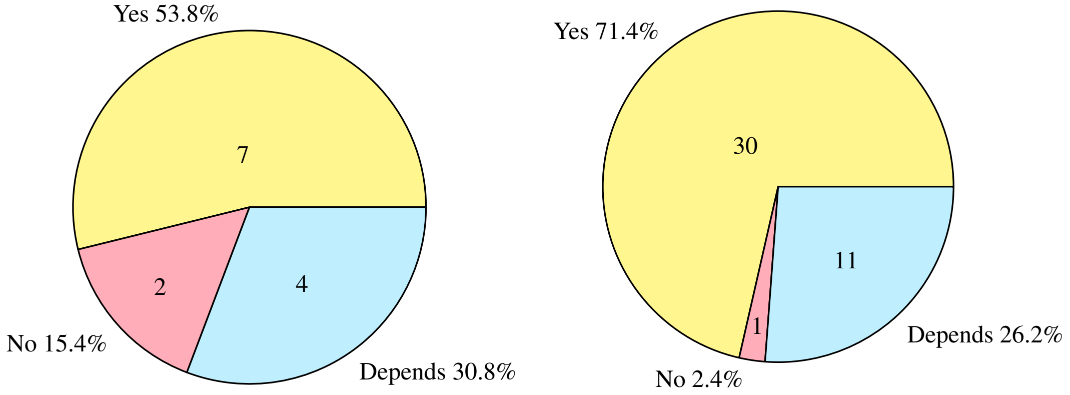
(b) Answers from students who HAVE NOT cheated.

Figure 1. "Do you consider cheating or plagiarizing a source-code assignment unethical?"

As shown in the graphs, a small majority of students who have already cheated believes that it is unethical and should be punished, even though they have done it themselves. Another significant amount said that it may be unethical and punished depending on the situation.

Among the students who said cheating may be unethical, depending on the situation, were some interesting comments. Most said they think copying the assignment entirely is unethical, but talking to friends or looking at another code to base their work on is not. Two students said during the major they have too many assignments to do and it is impossible to finish all of them without resorting to cheating. One student even said "The assessment also evaluates the Professor. If the student plagiarizes, the Professor is as guilty as them."

Regarding the question of whether cheating should be punished, 4 students who have already done it said it depends on the situation. Many said if it was the first time the student cheated on an assignment they should only be warned, and if they had done it more than once, the Professor could give this assignment a 0. Two students also commented that sometimes the assignment is "[...] impossible to do without the help of the internet or discussing with friends".



(a) Answers from students who HAVE already cheated.

(b) Answers from students who HAVE NOT cheated.

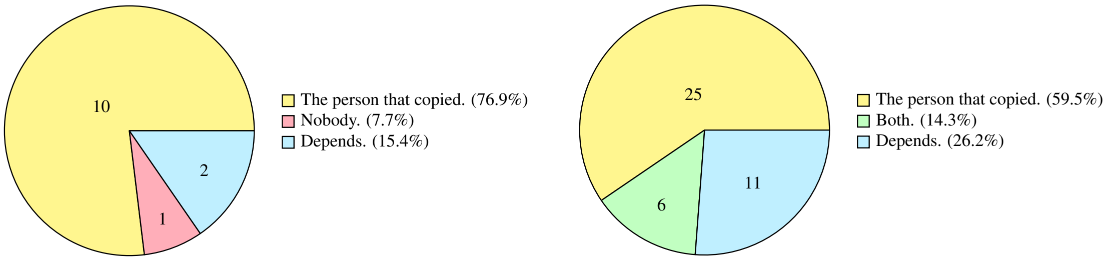
Figure 2. "Do you think cheating or plagiarizing a source-code assignment should be punished?"

As shown in the graphs, a large majority of respondents who have never cheated believe it is unethical to cheat or plagiarize programming assignments and it should be punished.

The smaller part, which thinks it may be unethical depending on the situation, has similar opinions to the students who have already cheated. Some said if the content and difficulty level of the assignment matches the classes given by the Professor, it is unethical to cheat, otherwise, it may be ethical. Others commented that copying is unethical, but discussing ideas is reasonable.

Many of the students that have not cheated and said that, depending on the situation, plagiarizing or cheating on an assignment should be punished, they also stated that each case should be judged individually, analyzing not only how the cheating happened but also why the student did it.

Moving on to the next question, "Who should be punished when a programming assignment is copied from someone else?", the following answers were received from the groups (Figure 3).



(a) Answers from students who HAVE already cheated.

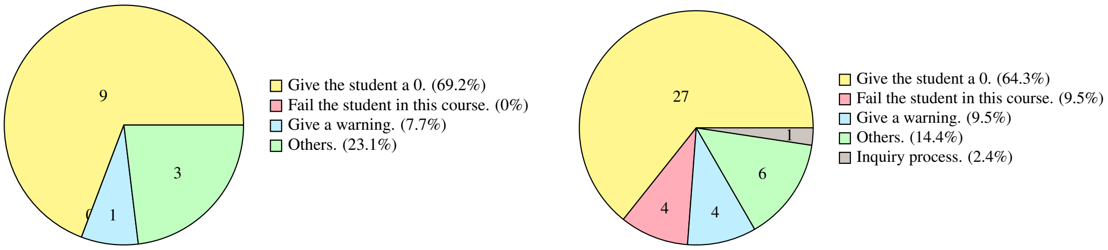
(b) Answers from students who HAVE NOT cheated.

Figure 3. "Who should be punished when a programming assignment is copied from someone else?"

Among the people who have already cheated, two said the punishment should depend on the situation, one of them added that sometimes it is impossible to determine the offending party and the other commented that maybe the person who was copied "[...] was just trying to help."

About the people who have not cheated, most of them said if the person whom the assignment was copied from knew about it and freely provided a copy of their assignment, they should be punished, together with the one who copied. One student commented that if the source-code is available online, the person should be punished.

The second to last question, "What could be the ideal punishment for cheating (or plagiarizing) on a programming assignment?", gave the following answers (Figure 4).



(a) Answers from students who HAVE already cheated.

(b) Answers from students who HAVE NOT cheated.

Figure 4. "What could be the ideal punishment for cheating (or plagiarizing) on a programming assignment?"

Most who chose "Others" suggested a combination between the other options, many said if it was the first time the student cheated, they should only receive a warning, otherwise, they should receive a harsher punishment.

Finally, the respondents were able to leave comments or other opinions they had about the subject. Many respondents said if the student had not merely copied but had also understood what they had copied and given credit to the source, it would not be a problem. Some people said simple assignments, as writing the code for a specific and well-known algorithm, are not good assignments and the Professors should give more creative and engaging tasks for the students.

6. Discussion

We have observed empirical evidence supporting the views described in this work. Specifically, as reported in previous sections, we have identified a variety of characterizations of plagiarism provided by students through the questionnaires.

In this work we also showed that many students considered cheating because they thought the assignment was too difficult. Considering that during their professional lives the students will certainly face many difficulties, shouldn't the undergraduate course be the place to deal with challenges? In this scenario, how should the Professor explore this theme during the classes so that the students realize its importance?

As it can be seen here, the biggest part of the students, both those who have already cheated and those who have not, considered cheating (or plagiarizing) unethical and punishable. However, more than 30% of the students who have cheated answered "depends" for both questions. Should the University, the Computer Science Department, the Professors or the students themselves create a campaign to discuss ethics on the University environment?

Another interesting observation is that one student said explicitly that the when a student cheats it is not their fault but the fault of the “failed evaluation system”. Is this probably a biased point of view? Or is this because they have a “wrong” definition of ethics?

The convenient accessibility of information is also a challenge for the Professors, specially after the Covid pandemic, because students are able to access information from different sources, such as videos, blogs or even social networks. The emergence of artificial intelligence tools that generate code automatically is also being observed. How can educators avoid plagiarism in these cases? This is already being discussed in many Universities, should the students be more involved in these discussions?

7. Limitations

As only 20% of the registered students answered the questionnaire, it is not possible to have precise conclusions.

Another limitation is the lack of demographics in this study, only the proportion of men and women registered on the major as a whole is known, but not how many people of each gender answered to the survey. We also do not know whether the respondents are freshmen or senior students.

In order to have a clearer and more precise view of Brazilian Computer Science students’ opinions on cheating and plagiarism a broader study is required, involving surveys from many different universities.

8. Conclusion

Empirical evidence suggests that universities are ambiguous with respect to Ethics, particularly Ethics of virtue. As a consequence, students do not have clarity with respect to what plagiarism is and to whether plagiarism could be considered ethically acceptable in specific cases. If the university does not embrace virtue-based ethical principles as fundamental to their activities, these concepts may never be fully clarified and, as a consequence, may continue to be a relevant issue to be solved.

In the future, it is planned to apply the same questionnaire, adding information on the demographic of the respondents, at two other highly reputed Universities, a Brazilian and a North American one. Besides, we are going to suggest to the Computer Science Department at the University of Sao Paulo that the Professors, or even the student representatives, discuss the questions we raised at the discussion session with the students, using lectures, round tables or informal talks.

References

Alzahrani, N. (2022). *Detecting Students of Concern in Introductory Programming Classes: Techniques to Indicate Potential Struggle or Cheating*. University of California, Riverside.

Aniceto, R. C., Holanda, M., Castanho, C., and Da Silva, D. (2021). Source Code Plagiarism Detection in an Educational Context: A Literature Mapping. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Lincoln, NE, USA. IEEE.

Bartneck, C., Lütge, C., Wagner, A., and Welsh, S. (2021). *An introduction to ethics in robotics and AI*. Springer Nature.

Brown, P. and Rosen, I. (2020). Misunderstandings, Mistakes, and Dishonesty: A Post-hoc Analysis of a Large-scale Plagiarism Case in a First-year Computer Programming Course. In *2020 ASEE Virtual Annual Conference Content Access Proceedings*, page 34977, Virtual On line. ASEE Conferences.

Cole, S. and Mccabe, D. (1996). Issues in academic integrity. *New Directions for Student Services*, pages 67 – 77.

Cosma, G. and Joy, M. (2008). Towards a definition of source-code plagiarism. *IEEE Transactions on Education*, 51(2):195–200.

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

Dick, M., Sheard, J. I., and Hasen, M. (2008). Prevention is better than cure: Addressing cheating and plagiarism based on the IT student perspective. In *Student plagiarism in an online world: problems and solutions*, pages 160 – 182. Information Science Reference, Hershey, PA. OCLC: ocn145379537.

Foltŷnek, T., Meuschke, N., and Gipp, B. (2019). Academic plagiarism detection: a systematic literature review. *ACM Computing Surveys (CSUR)*, 52(6):1–42.

Hu, G. and Lei, J. (2015). Chinese university students' perceptions of plagiarism. *Ethics & Behavior*, 25(3):233–255.

Hutton, P. A. (2006). Understanding Student Cheating and What Educators can do About it. *College Teaching*, 54(1):171–176.

Jordan, A. (2001). College Student Cheating: The role of motivation, perceived norms, attitudes, and knowledge of institutional policy. *Ethics and Behavior*, 11:233–247.

Joy, M., Cosma, G., Yau, J. Y.-K., and Sinclair, J. (2011). Source Code Plagiarism—A Student Perspective. *IEEE Transactions on Education*, 54(1):125–132.

Joy, M., Sinclair, J., Boyatt, R., Yau, J. Y.-K., and Cosma, G. (2013). Student perspectives on source-code plagiarism. *International Journal for Educational Integrity*, 9:3–19.

Karnalim, O. and Chivers, W. (2023). Reporting less coincidental similarity to educate students about programming plagiarism and collusion. *Computer Science Education*, pages 1–31.

Klein, D. (2011). Why learners choose plagiarism: A review of literature. *Interdisciplinary Journal of e-learning and learning objects*, 7(1):97–110.

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

Lathrop, A. H. and Foss, K. E. (2000). *Student Cheating and Plagiarism in the Internet Era: A Wake-Up Call*. Libraries Unlimited.

Lori G. Power (2009). University Students' Perceptions of Plagiarism. *The Journal of Higher Education*, 80(6):643–662.

Nwosu, L. I. and Chukwuere, J. E. (2020). The attitude of students towards plagiarism in online learning: a narrative literature review. *Gender & Behaviour*, 18(1):14675–14688.

Park, C. (2003). In other (people's) words: Plagiarism by university students–literature and lessons. *Assessment & evaluation in higher education*, 28(5):471–488.

Pierce, J. and Zilles, C. (2017). Investigating Student Plagiarism Patterns and Correlations to Grades. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 471–476, Seattle Washington USA. ACM.

Sheard, J. and 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*, pages 233–237, Darmstadt Germany. ACM.

Sheard, J. and Dick, M. (2012). Directions and dimensions in managing cheating and plagiarism of IT students. In *Proceedings of the Fourteenth Australasian Computing Education Conference*, volume 123 of *ACE '12*, pages 177–186. Australian Computer Society, Inc., AUS.

Simon, Cook, B., Sheard, J., Carbone, A., and Johnson, C. (2014). Student perceptions of the acceptability of various code-writing practices. In *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*, pages 105–110, Uppsala, Sweden. ACM Press.

Simon and Sheard, J. (2015). Academic Integrity and Professional Integrity in Computing Education. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, pages 237–241, Vilnius Lithuania. ACM.

Sindre, G. and Haugset, B. (2022). Techniques for detecting and deterring cheating in home exams in programming. In *2022 IEEE Frontiers in Education Conference (FIE)*, pages 1–8.

Vallor, S. (2016). *Technology and the virtues: A philosophical guide to a future worth wanting*. Oxford University Press.

Yu, L., Jiang, H., Zhu, H., Zhao, Q., and Chen, J. (2020). Investigating the Understanding of Plagiarism: A Case Study of Code Plagiarism in China. In *2020 15th International Conference on Computer Science & Education (ICCSE)*, pages 176–181, Delft, Netherlands. IEEE.