

## **Plagiarism detection in Programming coursework.**

**Prof. Constantine Mukasa, Northeastern University**

Constantine Mukasa received a B.S. degree in Computer Engineering from Bethune-Cookman University, Daytona Beach, Florida, USA in 2007, and his M.Sc. and Ph.D. degrees in Electrical Engineering from Florida Atlantic University, Boca Raton, Florida, USA, in 2013 and 2017, respectively. He is currently an Assistant Professor at Northeastern University in Boston, MA. His research interests include Engineering Education pedagogies, Experiential learning and Teaching Technology, Team-based Learning, Summer P12 Engineering Activities, Wireless network connectivity, and interference modeling.

## **Work in Progress: Plagiarism detection in First-year Programming coursework.**

### Abstract:

Students are taught to avoid plagiarism at all educational levels, from K-12 to graduate schools, but teachers still deal with plagiarized work regularly. All forms of plagiarism are unacceptable, and tools such as Turnitin have been developed to detect plagiarism on writing assignments. However, most First-year engineering coursework involves computer programming in languages such as C++ and MATLAB. These languages are specialized, making it difficult for traditional tools such as Turnitin to detect plagiarism. In this work-in-progress, we present Gradescope as an essential tool in addressing plagiarism concerns in programming coursework. This tool detects copied work and provides statistical similarity values such that the instructor can take adequate measures to address this issue promptly. In future work, we plan to investigate students' perspectives about the tool and its efficacy.

### Introduction:

Even though students are taught to avoid plagiarism at all levels of education, from k-12 to graduate school, higher education instructors still observe and deal with academic misconduct such as cheating and copying or using another person's work without permission or credit. It may be intentional or non-intention, and its effects are unwanted in the academic world. Regardless of the intention, higher education institutions have implemented ethical codes that address plagiarism. Those caught are punished based on the established ethical standards to curb and eliminate this behavior [1]. Additionally, there has been an array of technological tools that have been integrated into courses by instructors to detect and address plagiarism. Therefore, we focus on a tool used to detect plagiarism in a First-year programming course.

Typically, plagiarism is considered in writing assignments and research papers, where tools such as Turnitin are used to highlight plagiarised work. Turnitin is a web-based software that checks and compares submitted work to several online databases and highlights areas of concern. Unfortunately, Turnitin does not work well for engineering programming courses with files generated from programming software such as C++ or MATLAB. In this study, we focus on first-year courses that have about 30-150 students. In such typical large classes, the expectation is that each student's submission will be different as it represents how the student understands the concepts and their diverse prior programming experience. As a result, the submitted code will show a wide variation in the choices of variable names, syntax use, structure selections, and errors experienced. But students who intend to plagiarize hide it by making a few superficial changes that are hard for an instructor or teaching assistant (TA) to efficiently detect or when more than one grader is assigned to an assignment. It is such scenarios that can benefit from the use of similarity detection tools such as Gradescope.

### Gradescope Setup

Gradescope is a product of Turnitin company that launched in 2014. Gradescope prides itself as an efficient grading tool that easily integrates into several Learning Management Systems (LMS) such as Canvas. The company lists more than 1000 universities and 50000 instructors as users of this tool [2]. During the spring semester of 2021, we incorporated Gradescope into a First-year engineering course on Canvas that seamlessly synced assignment grades. Students upload the

actual source code files (e.g., .cpp, .m, etc.) on Gradescope, and the instructor/TA checked the submissions for similarities above 40% by clicking on a single button. Figure 1 shows one scenario of flagged source files and sections of concern submitted by two students.

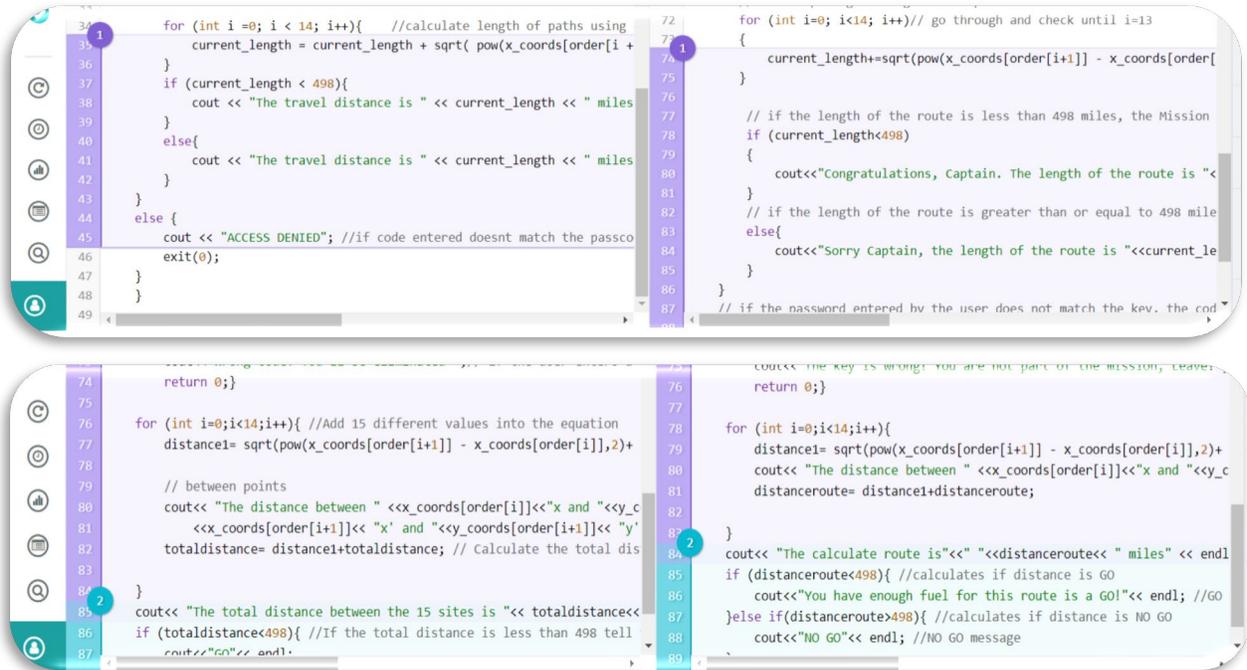


Figure 1: Falsely flagged (Top) and correctly flagged (Bottom) with similar sections side by side.

The instructor/TA should investigate further by clicking on the file to access more details and evaluate the flagged file sections for copying. For instance, a closer look at Figure 1, the top part was flagged, but the highlighted areas are not similar, and no copying was observed. However, a closer look at the bottom part shows that the flagged sections are so similar, and copying may have happened. *Note: This tool is not a fully developed plagiarism detection tool such as Turnitin that searches several databases, but it checks for similarity across submitted files.*

### Initial Study Results:

We were interested in the students' experience with Gradescope integration and whether such a tool can deter copying. So, we asked students to respond to an anonymous survey. About 93 students of 128 completed the study, and the results are shown in Figure 2 and Figure 3. Based on Figure 2, about 75% had a good experience, and about 14% responded with an unfavorable rating. These initial results highlight that Gradescope can successfully be incorporated into a class without significantly affecting students' experience. In Figure 3, about 87% selected a five (5) and above on the scale, indicating that Gradescope can help deter copying. But still, some students felt that the tools would not prevent them.

### Conclusion:

We hope that using such tools can help instructors quickly detect plagiarism and develop a strategy to address it. The idea is not to use the tool to police cheating but rather to detect opportunities to educate and leverage the cheating incident as a teachable moment [3]. Any adopted strategy to deal with cheating should be appropriate with transformative experiences or

activities that include reflecting, thinking, and developing new skills and experiences [4]. Students who cheat are less likely to attain the course learning objectives. Thus, we should provide them help and assistance to make sense of the experience and develop better decision-making skills. Otherwise, they may develop better ways to cheat or hide the act.

How would you describe your overall experience with Gradescope?

93 responses

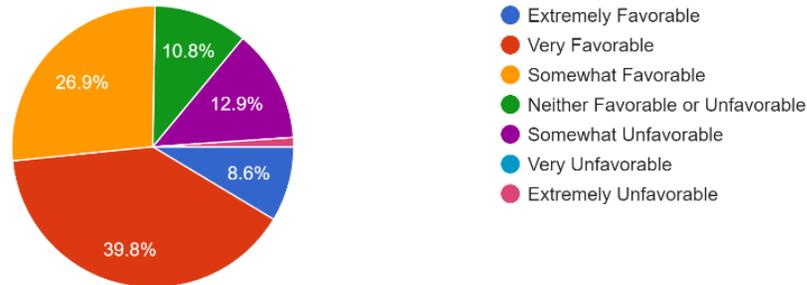


Figure 2: Students' experience with Gradescope.

Overall, how likely did/would the code similarity check tool deter you from copying or plagiarizing someone's code?

93 responses

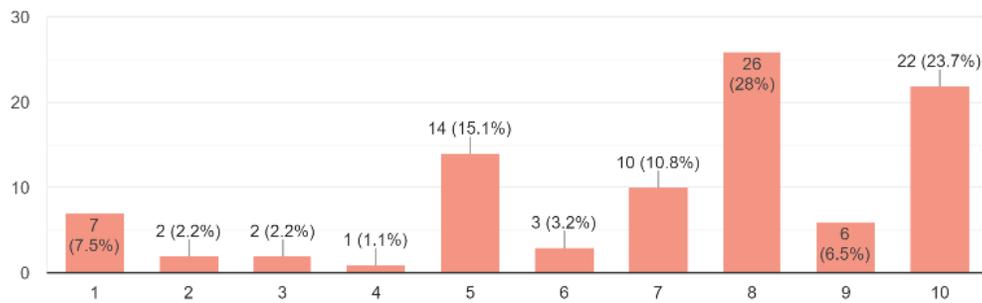


Figure 3: Student perception on similarity check tool.

## References:

- [1] W. Merkel, "Collage of confusion: An analysis of one university's multiple plagiarism policies," *System*, vol. 96, p. 102399, 2021/02/01/ 2021, doi: <https://doi.org/10.1016/j.system.2020.102399>.
- [2] "Gradescope: Deliver and Grade Your Assessments Anywhere." <https://www.gradescope.com/> (accessed 05/20, 2021).
- [3] B. T. Gallant, "Academic Integrity as a Teaching & Learning Issue: From Theory to Practice," *Theory Into Practice*, vol. 56, no. 2, pp. 88-94, 2017/04/03 2017, doi: 10.1080/00405841.2017.1308173.
- [4] A. Y. Kolb and D. A. Kolb, "Learning Styles and Learning Spaces: Enhancing Experiential Learning in Higher Education," *Academy of Management Learning & Education*, vol. 4, no. 2, pp. 193-212, 2005. [Online]. Available: <http://www.jstor.org/stable/40214287>.