



Blog · 5 min read · Wednesday 8 November 2017

Plagiarism and Programming: How to Code Without Plagiarizing



Jonathan Bailey

Plagiarism Consultant --
Plagiarism Today



In honor of National STEM Day, we are investigating plagiarism in the STEM subjects. In this post, guest blogger Jonathan Bailey examines the rise of plagiarism in computer science classes.

Programming classes are rapidly becoming among the most popular classes at universities. But, as more and more students are learning to code, an increasing number are finding themselves accused of plagiarism.

According to a recent [New York Times article](#), at Brown University, more than half of the violations of the academic code involved cheating in computer science classes. Similarly, at Stanford, 20% of one computer science class were flagged for cheating.

Part of this is that programming, in many ways, is still a new form of creativity. While we have centuries of discussion surrounding plagiarism and the written word, programming has far less history as a means of communication and it doesn't have the citation standards that we do with research papers.

This can make a lot of the boundaries when it comes to plagiarism and programming seem ambiguous. This is an issue not just in the classroom, but also in the courtroom. This has been highlighted in [the seven-plus year litigation between Google and Oracle over Google's use of Java APIs](#) to make Java applications run on Android.

In that case, Google admitted to copying the "declaring code" so the apps would be compatible but did not copy the "implementation code" which determines how Android processes the instructions from those apps. Though representing only a very small percent of the code in Java, the copied APIs were enough to kick off both a huge lawsuit and a major legal debate.

However, in the classroom, there's only one judge to be concerned with and that is the instructor.

As such, when it comes to avoiding plagiarism in programming classes, the best first step is the same as in any other class: Listen to your instructor and ask questions when you are unsure of how to proceed.

Instructors are there to guide you and help you learn, part of that is with plagiarism and citation issues.

When working with your instructors, there are several issues to focus one:

- What are the rules on using outside code?** When, if ever, is it acceptable to bring in outside code? What sources can it be from? Many classes will allow outside code but only from approved sources, such as a classroom library or open source projects. Knowing when and where it's acceptable to use outside code is crucial to avoiding plagiarism.
- What are the rules on collusion?** A large number of programming plagiarism cases deal with collusion, with students in a class copying from one another. Understanding if, when and how students can help one another in the class is invaluable to avoiding cheating allegations.
- How do you cite unoriginal code?** When you do use code, either from another student or an outside source, how do you cite it?

Beyond that there are several other steps you can and should do to ensure that you do not face accusations of coding plagiarism.

- Code in a Cleanroom:** [Writing in a cleanroom](#) is a process through which you physically separate outside text from you own. You can and should take similar precautions with your code, never mingling outside code with your work unless you've properly cited it.
- Comment Your Code Thoroughly:** Every programming language has a means of inserting comments into code and thoroughly commenting code is a good practice every programmer should follow. However, it is doubly important when incorporating outside code into your work, as comments are a simple way to illustrate what is and is not original. It's also useful if your instructors have any question about whether you wrote the code as good comments can illustrate a thorough understanding of what was written.
- Use Original Variable Names:** When possible, ensure that your variable, class and other names are original. To be clear, simply changing variable names on copied code does NOT make it original, [as pointed out by the University of Pennsylvania](#), but having unique and meaningful variable names not only makes coding easier for you, but also highlights that the work is yours.

With coding, the most important thing is to always do your work for yourself, unless specifically allowed and/or instructed to do otherwise. While learning to work with outside code is an important skill, programming assignments are meant to test your coding skill. That is impossible if you simply copy and paste the work of others.

If you do your work, cite what you do copy and don't work with other students without permission, you should be fine. In that regard, programming is very similar to any other assignment in college.

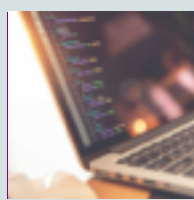
This post was contributed by Jonathan Bailey, foremost expert in plagiarism. He has spent over 16 years fighting plagiarism professionally and currently blogs on [Plagiarism Today](#), where he raises awareness about the societal effects of plagiarism.

Related Articles



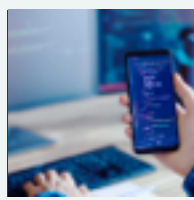
Blog
Three Unique Academic Integrity Challenges in Computer...

When it comes to academic integrity, every field of study faces unique challenges, both in terms of teaching...



Blog
Originality in Coding

Coding may be a unique and creative field, but the importance of giving credit to others' work is no less...



Blog
What is Programming Plagiarism? Why Is It on the Rise?

What is programming plagiarism? It is, simply put, using another person's source code and claiming it as your own....

Subscribe!

Turnitin blog posts, delivered straight to your inbox.

First Name *

Email *

By completing this form, you agree to Turnitin's Privacy Policy. Turnitin uses the information you provide to contact you with relevant information. You may unsubscribe from these communications at any time. *

☐ I Agree

[Submit](#)

- Products
- Feedback Studio
 - Gradescope
 - Turnitin Originality
 - Turnitin Similarity
 - iThenticate
 - Revision Assistant

- Solutions
- Higher Education
 - Secondary Education
 - Research & Publication
 - Partners
 - Customer Stories

- Support
- Help
 - System Status
 - Support Center

- Company
- About
 - Careers
 - Press
 - Blog
 - Accessibility
 - Product Privacy

- Connect
- Turnitin Educator Network
 - Twitter
 - LinkedIn
 - Facebook
 - Pinterest

