Updated September 29, 2017

NYCDA Intensive Programs

Plagiarism Guide

What is Plagiarism?

General Definition

The act of using the ideas or work of another person or persons as if they were one's own without giving proper credit to the source. Includes the submission of a work, either in part or in whole completed by another; failure to give credit for work that belongs to another, use of another's project or part thereof without giving credit.

In the Context of Programming in the Classroom

Plagiarism is the act of copying code that was written by someone else and passing it off as your own without citing the original source material.

NYCDA Policy

Plagiarism is not tolerated at NYCDA and is a violation of academic integrity as outlined by the Academic Policies and Standards. Students who are found to be violating the academic integrity policy around plagiarism will be approached by their instructor and asked to resolve the issue immediately. If a student violates this policy a second time, they will have to make an appointment with their Success Coach and given an Academic Warning. If this behavior continues, the student's case will be moved to the NYCDA Academic Committee who will determine if the discretions warrant immediate dismissal.

Rationale

Plagiarism hurts your learning process

One of the major skillsets that you will need to nurture as a programmer is the ability to read someone else's code, interpret it so that you can learn from their code to build upon yours. Oftentimes, during job interviews and in the workplace you will be tasked to modify someone else's code or build a feature that adds to it. When plagiarizing code, you are eliminating a learning opportunity for yourself that is essential to your career and will likely hurt your chances of success in the future.

It will probably break your project

When working on a larger project that involves many features, code that is taken from an outside source verbatim will more than likely not fit the style nor the logic in which your project is written in. Therefore, you may find yourself spending more time adapting that piece of code into your project than you would have if you would have written it yourself.

How to Write “Honest” Code

1. Read through any relevant documentation regarding what the possibilities are for what you are trying to build. (e.g. doing research on what kind of animations are available in css before coding so you know what you can do)
2. Try your best to code a solution to the best of your ability using the knowledge you already have.
3. If you find yourself at a loss of how to code something, search on Google specific keywords that narrow down how to solve individual roadblocks as opposed to searching for an entire solution or answer. (e.g if you are unsure how to save something to the database using Ruby then perhaps you can search “storing information in a database using Ruby”)
4. When searching the internet if you find articles that identify a solution to your current roadblock, make sure to read through the content so you can understand why the solution works as well as what the solution is.
5. If you find sample solutions in code to particular problem you are having, try your best to understand the logic. This is challenging at first but becomes easier the more you do it. It is a necessary skill to learn as a programmer.
6. If you do choose to use code from an outside source, rewrite the code in your own style, cite the original source, and write a brief description explaining what it does in a comment.

Note: If you are working with a group, avoid copying a group member’s original source code. Ask them to walk through what their code does step by step so you can interpret and come up with a solution in your own style.