

Writing Original Code

Here is a compilation of best practices from FSE faculty members (added to and edited by the Academic Integrity Office) for writing original code that does not violate our academic integrity policy. As always, consult your instructor for specific class details and guidelines.

Tips from the Inner Circle AIO Academic Integrity Tips article are:

- You can help classmates understand the theory and concepts but never show or send your code to another student.
- Instead of Google or other online resources, use the textbook, assigned readings and instructor office hours when/if you get stuck.
- Changing the name of the variables, colors, font styles, output formatting in a code is NOT enough to claim that it is your original code.
- If you consult other allowable resources, read the code carefully, understand and try to extract the logic/behavior behind it; this way you can use it as inspiration and then type your own code for your own purposes without copying and pasting.

Additional tips and suggestions are:

- Start the assignment early so when/if you get stuck, you have time to access the ASU resources needed to learn the material.
- If working in a team project, make sure the contribution from each team member is clearly indicated.
- Add your name as the author of your code.
- Use an Integrated Development Environment (IDE). IDEs provide enough help, such as syntax check, code completion, and debugging for you to create bug-free code on your own.
- If your instructor allows you to use other code but requires attribution, do not delete the headers of the original code and make sure that you are using the proper reference for the work.
- Remember that even when using open-source code, check the copyright of it to avoid breaking any rule.
- Always try to use course sponsored discussion boards and Discord servers only. Participating in student-run Discord servers can often result in code sharing and distribution of inaccurate information. When in doubt about a discussion board, ask your instructor.
- The most powerful and useful feature of modern high-level programming languages (Like C++, Java and Python) is the ability to name things. Use variables to name values and to keep your expressions simple. Use functions (methods) to name instructions and to keep your logic simple. Break complex expressions down into a sequence of simple expressions, storing intermediate values in well-named variables. Break complex functions (methods) down into two or more simple methods that each do what their name says they do.