

Coding and Academic Integrity

Author: Dr David Cooper, School of Elec Eng, Comp and Math Sci (EECMS), Curtin University.

Modified: 13-01-2018, Mark Upston, School of Elec Eng, Comp and Math Sci (EECMS), Curtin University.

Approved: 14-01-2018 by Prof Kevin Fynn, HoS, School of Elec Eng, Comp and Math Sci (EECMS), Curtin University.

“This is not about you getting the job done, it is about you **learning how** to get the job done”

Key Guidelines for Curtin University Coding Assignments

1. Do not copy or adapt other people’s code, use third-party libraries, or re-use your own pre-existing code without official permission.

If you do, the unit coordinator may determine that you did not actually complete the assignment properly. Hence, you could receive a reduced or zero mark in the normal course of the marking process. You cannot receive marks for other peoples’ work, or for your previously-submitted work.

Note 1: A third-party library usually refers to any library that is not part of the standard API of the language/platform being used.

Note 2: Official permission can be, but is not limited to, both legal authorisation and written permission from the Unit Coordinator or their delegate.

2. Use in-code comments to give precise references to all sources.

This applies to any external code you copy or adapt, to any third-party libraries you use, to any pre-existing code you wrote yourself, AND to any other reference material on which your code is based, such as algorithm listings, file format descriptions, etc.

You must indicate precisely *which* code (from among the code you submit) was obtained from or based on these external sources. Refer to the **How to Reference Code** section

Missing or inadequate referencing constitutes plagiarism, and you could receive a penalty for Academic Misconduct.

3. Do not share code, or work in a group, without official permission.

Working together on the same code, or sharing your code prior to submission, constitutes collusion (except where the assessment explicitly allows or requires groupwork). You could receive a penalty for Academic Misconduct.

4. Do not make your assignment work publicly-accessible (until after the semester is over).

If you use online services like Github, BitBucket, Google Drive, etc. in preparing your assignment, you must ensure that nobody else can access your work (other than relevant teaching staff and anyone in your group, in the case of legitimate groupwork). Use only “private” repositories.

If your assignment work is publicly-accessible prior to the due date (or even for several weeks *after* the due date, allowing for possible extensions), this may be considered collusion, and you could receive a penalty for Academic Misconduct. If you are in any doubt when you can make it public, seek permission from the Unit Coordinator.

Principles

There are two key principles at stake. The first is Academic Integrity, and you can find more information on this at academicintegrity.curtin.edu.au. In part, this means avoiding plagiarism and collusion, and thus ensuring you precisely and accurately reference all your sources. Coding takes creativity, just like essay and report writing, painting, music composition, etc., and hence if you use other people's code, you must respect and acknowledge *their* creativity.

The second principle is simply the need to *do your own work*, and to have others do *their* own work too.

In an industrial setting, where the goal is simply to “get the job done” efficiently and effectively, teamwork and code reuse is essential practice. You must simply ensure you have the legal right to reuse a given piece of code, and follow any conditions stipulated by the licence agreement (which almost certainly implies referencing the original author(s) as well).

In contrast, a university assignment is designed to be a learning opportunity, linked to the Unit Learning Outcomes, where you build up your own understanding and experience in applying certain concepts. It is also a means to assess (mark) your progress, to ensure that you get recognition for your efforts. If parts of the assignment task are instead solved by other people, then you haven't had the proper learning experience. Moreover, you should not expect to be awarded marks for work that is not actually yours, even if you reference it correctly.

For coding assignments, these principles are especially important to remember, because code can often be more “copy-able” and “groupwork-able” than essays or reports. (Code is usually broken down into many loosely-connected, re-usable parts, which can be worked on separately, and copied as-is, or with very minor changes, into a different piece of software to help solve a different problem.)

In some university assignments and projects, you will be given some pre-existing code as a starting point, and/or requested to work as part of a team. In such cases, the lecturer or unit coordinator has determined that you can achieve the required amount of learning without doing everything yourself. However, this is the prerogative of the lecturer/UC, and you should not make any assumptions. If in doubt, reference it!

How to Reference Code

Here are some approaches to referencing other people's work in a university coding assignment, listed in order of preference. You *must* have written official permission to use other people's work in the first place, of course.

1. Using libraries. By their nature, libraries are pre-packaged collections of reusable code (e.g., GTK, Apache Commons, etc.). In a limited sense, they are actually self-referencing, in that in order to use a library you have to specify what it is, or it simply won't work.

Standard libraries do not require referencing (beyond what is required to make them work), as they are the code equivalent of "common knowledge":

```
import java.util.Scanner;
...
#include <stdio.h>
...
```

Third-party libraries should have more explicit referencing, in a comment at the top of any relevant source code files:

```
// Uses "Apache Commons Lang 3.6", Apache Foundation,
// https://commons.apache.org/proper/commons-lang/
import org.apache.commons.lang3.text;
...

/* Uses "GTK+ 3.0", The GTK Team, https://www.gtk.org */
#include <gtk/gtk.h>
...
```

2. Copying external code into a set of separate (isolated) files. If the external code you wish to use has not been pre-packaged into a library, the next best approach is to copy the relevant files or snippets of code whole into a separate directory of your project. If you copy whole files, keep them *unmodified* except to add referencing information to the top as a comment (if not already there).

```
// Obtained from X. Programmer, http://example.com/thecode.java
// (accessed 22 March 2017).

public class ExampleCode
{
    ...
}
```

If you have several code snippets from one or more sources, you might bundle them together and give referencing information for each one:

```
/* This file comprises externally-obtained code. */

#include "example.h"

/* Obtained from X. Programmer,
 * http://stackoverflow.com/questions/12345/abc-xyz
 * (Accessed on 31 May 2017)
 */
int theFunction(double x, double y)
{
    ...
}

/* Obtained from S. Dev,
 * http://github.com/dev/example/tree/master/src/thecode.c
 * (Accessed on 2 June 2017)
```

```
*/  
void anotherFunction(void)  
{  
    ...  
}
```

3. Copying externally-obtained code, or adapting algorithmic logic, into your own files. Sometimes you might need to integrate externally-obtained code into files that also contain your own code. You must clearly show where each externally-obtained snippet starts and stops, and where each one comes from. (A comment at the top is not enough.)

```
/* Parts of this file comprise externally-obtained code. */  
public class MyCode  
{  
    public static void main(String[] args)  
    {  
        ...  
        // Obtained from X. Programmer,  
        // http://stackoverflow.com/questions/12345/abc-xyz  
        // (Accessed on 31 May 2017).  
        for(int i = 0; i < 10; i++)  
        {  
            doSomething();  
        }  
        // End of code obtained from X. Programmer  
        ...  
    }  
}
```

If you have not copied anything *per se*, but rather implemented an algorithm based on an external source, reference this too:

```
/* Algorithm based on Author and Author, "Amazing Algorithms" (2008),  
page 451. */  
...
```

4. Constructing your own code by means of modifying an existing template. DO NOT DO THIS for university assessment purposes, UNLESS the lecturer has explicitly provided certain code for this purpose, or explicitly allowed you to find some third-party code. There is no easy way to indicate where your work stops and someone else's work starts. *If* you do have permission to do this, reference the original source in a comment at the top of the file (if such a comment is not already there):

```
// Based on code from X. Programmer, http://example.com/thecode.java  
// (accessed 22 March 2017).  
public class ExampleCode  
{  
    ...  
}
```

Plagiarism Detection Frequently Asked Questions

There are many questions surrounding detecting, investigation and penalising of plagiarism, particularly in regard to code, that need to be explained:

Do staff who detect and investigate potential plagiarism also decide on penalties?

By design, these things are handled by different people. Potential Academic Misconduct is reported by markers and teaching staff, investigated by an “inquiry officer”, the Student Discipline Panel then determines whether or not misconduct has occurred, and what penalty is to apply.

What this means is that you cannot ask anyone to “be lenient”, or “just go ahead and give me zero”. This power is not given to anyone involved in the reporting or investigative stages.

When the decision is made by software, could I/will I be penalised if the similarity/originality score is above a certain value?

All decisions are based on human academic judgment, and the final outcome is not based on any auto-generated similarity score. There is no similarity score that you are “allowed to get away with”, nor is there a score that guarantees you will be penalised. Penalties for plagiarism are never automatically applied or dismissed.

The numbers shown by any plagiarism/collusion detection tool serve only to draw the staff’s attention to similarities that might need to be investigated. However, an investigation could still be undertaken, and a penalty imposed, irrespective of the similarity score. Indeed, the process does not inherently require a detection tool at all.

I want to be able to check the similarity score before submitting. Will the software tell me whether I have plagiarised?

This is a confusion of concepts. You *already know* whether you have copied someone else’s work. You do not need software to tell you this. Moreover, no software is able to check your work against the work of your classmates prior to submission (unless you have actually obtained those submissions, which in itself is likely to constitute Academic or General Misconduct).

Turnitin lets you check written work against its database of *pre-existing* material. Even then, however, Turnitin cannot be used for code, and tools like TokenDiff do not necessarily have a database at all, in which case there is nothing to check against prior to submission.

Could I be unfairly penalised due to false-positive similarities: standard code, examples, conventions, etc. made available to all students?

Again, all decisions are based on human academic judgement. Academic staff are well aware of the factors that might legitimately cause two students’ code to appear similar. Such factors are always taken into account. (TokenDiff even has a feature to disregard legitimate common code during the comparison process.)

As part of an inquiry, you have the right to prove you are innocent. Use some form of repository to keep records of everything – your drafts, versions that didn’t work, etc.

What if there is only one way to write this code?

There are *very, very many* ways to write it. This is one of the key lessons in software development. Code would not be copyrightable if there was only one solution to a given problem.

While there are constraints imposed by the nature of the assignment, there are always many different possible solutions within those constraints. If you and another student have made choices leading to the same code, this is cause for suspicion and possibly allegations of plagiarism and/or collusion.

What if I change the code? Will it still be plagiarism?

If the code you submit is in any way *based on* someone else's code, you must cite your source. If you do not, then you *are* committing plagiarism, irrespective of any changes you might make.

If I change the code, will I be *caught* for plagiarism?

Changing the names of variables, functions, methods, etc. will have virtually no effect on the results from TokenDiff. Neither will changing formatting, comments, loop types, or any number of other tricks. Indeed, this may be interpreted as a deliberate attempt to avoid being caught for plagiarism, which implies deliberate intent to gain unfair advantage, which is misconduct.

If I can explain how the code works, will I be penalised for plagiarism?

Being able to explain the code you've copied does not make it your code, and it does not get you off the hook. While it's true that you *must* be able to explain your code or face suspicion of plagiarism, the opposite is not necessarily true.

Explaining code does not require the same level of comprehension as writing the code in the first place. *Writing* code is the skill that we expect you to develop and demonstrate in a coding assignment.

Can I rebut a plagiarism allegation by pointing out that individual words and symbols are common or expected?

When confronted with a plagiarism allegation, you must understand the evidence. Evidence of plagiarism is not just a collection of matching words and symbols, but the way those words and symbols are put together – their sequence and context. Discussing them separately misses the point.

A finding of plagiarism is made by considering the *length* of duplicated sections of work, the uncommonness of *combinations* of words and symbols (notable idiosyncrasies), and any ways in which those duplicated sections could have occurred legitimately. This is what you must address.

My friends can be trusted with my code. They just want to get some ideas. Can I give them my code?

They are not just "getting ideas". They are attempting to commit Academic Misconduct, and so are you if you agree to their request. Whether or not they actually copy your code, you would be helping them gain unfair advantage over other students. A real friend would not put you at risk of a misconduct penalty for their own personal gain.