



# Plagiarism of computer code

---

**Target audience:** Subject coordinators, academic integrity decision makers

**Key issue being addressed:** Issues surrounding the 're-use' of unacknowledged sources of computer code

**Purpose of the case:** To illustrate some of the contexts of plagiarism beyond the copying of text

**Materials and preparation needed to answer case:**

- Remind participants to find and access appropriate policy and academic integrity resources at own institution prior to coming to session.
- Copy of university academic integrity policy and procedures.
- Separate PowerPoint for facilitator based on 1 or 2 hour session.
- Arwin, C, & Tahaghoghi, SMM 2006, 'Plagiarism detection across programming languages', Proceedings of the 29th Australasian Computer Science Conference, vol. 48, pp. 277 -286, Australian Computer Society, accessed 17 June 2012, <[crpit.com/confpapers/CRPITV48Arwin.pdf](http://crpit.com/confpapers/CRPITV48Arwin.pdf)>

## The case

### *Abstract*

*A subject coordinator is dismayed to find that many of his applied programming students have submitted assignments containing copied source code.*

Jonathan<sup>1</sup> is the subject coordinator in an undergraduate subject on applied programming. The assessment tasks in this subject require students to design, implement, test and debug simple programs using a specific programming language. The assignments require students to submit program source code. Jonathan knows that students can be tempted to copy code from friends, the Internet, or a 'private tutor' rather than create their own source code. Jonathan tries to design the subject and the assessment tasks to reduce the opportunities to do this because he recognizes that if students do not acquire the skills to manipulate the programming language they will not really have met the intended learning outcomes of the subject.

When one of Jonathon's colleagues, Amy, asks if she can trial the program she has developed to test for shared code on some of the assignments submitted by Jonathan's students Jonathan agrees, not thinking that any plagiarism of code would be detected. He is astonished to find over 15% of the students in his subject have submitted assignments containing code which Amy's program indicates contains shared code. In Jonathan's mind this means that students have copied.

---

<sup>1</sup> All characters in the case are pseudonyms

### Questions for discussion

1. What does your university policy say about designing the assessment of student learning to support student learning?
2. What does your university policy say about the copying and translation of computer code?
3. What does your university policy say about the copying and translation of computer code from one computer language into another (also called 'porting' or 'inter-lingual plagiarism'?)
4. What other information would Jonathan need in order to make sound interpretations of the meaning of Amy's findings?
5. What other strategies could Jonathan implement to reduce the incentive to, and risk of, students copying code?
6. It is acceptable for Jonathan to investigate the matters further given that students did not receive any notice that their assignments would be analysed using Amy's program?
7. What information should Jonathan obtain about the quality of Amy's program before using it to analyse student assignments

### Suggested readings

Collberg, C & Kobourov, S 2005, 'Self-plagiarism in computer science', *Communications of the ACM*, vol. 48, no. 4, pp. 88-94, accessed 17 June 2012  
<<http://cacm.acm.org/magazines/2005/4>>

Devlin, M 2002, 'Plagiarism detection software: How effective is it?', Excerpt from R James, C McInnis and M Devlin, *Assessing Learning in Australian Universities - Ideas, strategies and resources for quality in student assessment*, Centre for the Study of Higher Education, University of Melbourne, accessed 17 June 2012,  
<[cshe.unimelb.edu.au/assessinglearning/docs/PlagSoftware.pdf](http://cshe.unimelb.edu.au/assessinglearning/docs/PlagSoftware.pdf)>

Joyce, D 2008. 'Academic Integrity and Plagiarism: Australasian perspectives', *Computer Science Education*, vol. 17, no. 3, pp. 187-200.

Mann, S., & Frew, Z 2006, 'Similarity and originality in code: Plagiarism and normal variation in student assignments', in D Tolhurst & S Mann (Eds.), *Proceedings of the Eighth Australasian Computing Education Conference* (pp.143-150), Darlinghurst, Australia: Australian Computer Society, accessed 17 June 2012 <[crpit.com/Vol52.html](http://crpit.com/Vol52.html)>

*Support for this project/activity has been provided by the Australian Government Office for Learning and Teaching. The views in this project do not necessarily reflect the views of the Australian Government Office for Learning and Teaching.*



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Australia License](https://creativecommons.org/licenses/by-nc-sa/3.0/au/).