Student Created Cheat-Sheets in Examinations: Impact on Student Outcomes

Michael de Raadt
Moodle
michaeld@moodle.com

Abstract

Examinations have traditionally been classified as "open-book" or "closed-book" in relation to the freedom for students to bring resources into examinations. Open-book examinations can have benefits, such as reduced anxiety, de-emphasis of memorisation and reduced cheating. But open-book examinations can also have disadvantages such as reduced preparation and the need for time during examinations to look up facts. An emerging alternative allows students to bring a 'cheat-sheet' of hand-written notes. This form of examination has the potential to offer many of the benefits of an open-book examination while overcoming some of its failings. There has been little evidence showing that cheat-sheets can have an impact, and what exists is contradictory. This study reveals that students who create and use cheat-sheets performed better, on average, in an introductory programming examination. Certain features of cheat-sheets were found to be related to superior performance, which may relate to student understanding.

Keywords: cheat-sheet, examination, open-book, computing education, introductory programming

1 Introduction

Resources available to students can be controlled in order to allow students access to information during an examination. In a closed-book examination, students are not permitted to bring any information into the examination that may assist them; they are required to rely on their memory to recall the information they need. In the 1950s educationalists began to explore new forms of examination and one form suggested was the open-book examination. "In such an examination the student is allowed to make use of any materials at his disposal, including textbooks, lecture notes and dictionaries, but does not obtain answers directly or indirectly from other students" (Kalish, 1958, p. 200). Between these extremes, examiners can constrain students' access to materials to varying degrees; this form of examination can be referred to as a restricted examination. One form of restricted examination allows students to access a teacher created set of notes that summarise facts needed during the examination. Another form of restricted examination allows students to bring their own prepared notes, or 'cheat-sheets', into the examination setting. The purpose of this study is to explore the potential benefits of student created cheat-sheets within the context of an introductory programming examination.

This paper will first present an overview of research in the area of examinations relative to students' freedom to bring materials. This will be followed by the description of a study of the use of cheat-sheets in an introductory programming course and the results of that study. Finally, conclusions and recommendations will be made.

1.1 Open Book Examinations

The use of textbooks and teacher-prepared notes has been a topic of study for some time, with published discussions dating back over 60 years. Tussing (1951) suggested that the advantages of open-book examinations can include:

- reduced anxiety in the form of "fear and emotional blocks";
- a shift in emphasis from memorisation to reasoning; and
- reduction of cheating.

Experimental studies have reported that open-book examinations can have benefits. Schumacher et al. (1978) ran a controlled experiment to compare performance on open-book and closed-book examinations. They found significantly higher average scores when students had access to a textbook during their examination. Theophilides and Koutselini (2000) found evidence that student attitudes towards a course improve where there is an open-book examination.

There are also counter-arguments to the use of open-book examinations. These include a degradation of the seriousness of examinations that can lead to superficial learning (Kalish, 1958). Rather than benefitting students, allowing students to have access to textbooks or teacher-prepared notes can be a hindrance. Boniface (1985) compared the time spent referring to open-book materials and their results. He found that students who spend more time than others referring to such materials tend to end up with poorer marks. It tends to be students who have performed poorly on prior assessment items that rely more on these materials during an examination.

In an experimental evaluation of open book examinations, Kalish (1958) found that average scores were not affected when comparing open-book and closed-book examinations, and concluded that open-book examinations may benefit some students more than others. These findings were echoed by Bacon (1969). According to Feldhusen (1961), students prepare less for an open-book examination, which may ultimately decrease their overall learning.

1.2 Student Created Cheat-Sheets

A student created cheat-sheet is simply a sheet of notes produced while preparing for an examination. The size of the sheet can be specified and a common size is double-sided A4. While students are generally free to add whatever information they believe is relevant, the production of the sheet may be constrained; for example a teacher may specify that the sheet must be hand-written. Forcing students to hand-write their cheat-sheet is a mechanism to ensure that students make some effort to produce the sheet, rather than simply printing course notes or photocopying another student's sheet.

The effect of student created cheat-sheets has been less well explored than open-book examinations, but work in this area is more recent. Dickson and Miller (2005) explored students' use of cheat-sheets in psychology examinations, finding evidence that suggested cheat-sheets did not improve performance and did not reduce student anxiety. Dickson and Miller later revisited student created cheat-sheets using a different experimental approach and focused on the suggestion that preparing such sheets may encourage learning. They allowed students to prepare cheat-sheets, and then at the examination Dickson and Miller removed the cheat-sheets from students and asked them to take the examination. After this, they returned the cheat-sheets and allowed the students to take the examination again. Results showed that students performed better when they had access to their cheat-sheets and from this they concluded that cheat-sheets did not encourage greater learning, but did assist students during an examination (Dickson & Miller, 2008). Dickson and Miller failed to take into account that cheat-sheets are intended to relieve students of the burden of memorisation, yet memorisation seems to be what their experiment was measuring.

Almost in complete opposition to Dickson and Miller is the work of Erbe (2007), who suggests that student created cheat-sheets can reduce examination anxiety while increasing learning, particularly in courses that assess on the first three levels of Bloom's taxonomy (Bloom, 1956). Erbe emphasises that examinations do more than assess learning; the way examinations are structured and implemented can cause student learning, a thought also shared by Yu, Tsiknis and Allen (2010). Erbe suggests that open-book examinations can cause students to be lulled "into too much of a sense of security and, if they had not prepared adequately, the book was not very useful anyway" (p. 97). Erbe quotes Boniface (1985) in relation to this. Erbe noticed variety in the content and composition of cheat-sheets constructed by her students and now awards a prize for the best cheat-sheet; doing this also helps to reduce the tension around the examination. Erbe states that while students spend a lot of time preparing their cheat-sheets, they do not actually refer to them often in the examination: "Preparing the cheat sheets proved to be sufficient for learning what was on the test. This was the major difference between handing out information composed by me and having the students find their own. Students tailored the information to their own needs and wrote down information they still needed to learn. The act of writing and organizing the information for the cheat sheet allowed most students to fill in the holes in their knowledge" (p. 97). A number of other educationalists share the same view (Janick, 1990; Weimer, 1989), however there does not appear to be objective, empirical evidence to support this view.

There can be diversity in the quality and composition of student created cheat-sheets. Visco et al. (2007) analysed the cheat-sheets that students created for a chemical engineering examination. They found great variety among students' cheat-sheets and suggested that the "goodness" of a cheat-sheet does not necessarily map to examination performance.

1.3 Examinations in Computing Education

Since 1988 when Bagert asked the question: *Should computer science examinations contain "programming" problems?* (Bagert Jr., 1988), instructors of programming have been considering how students should be assessed, particularly in examinations. The nature and content of examinations are currently a topical issue in Computing Education research (Lister et al., 2004; Lister et al., 2006; Sheard et al., 2008; Sheard et al., 2011; Whalley et al., 2006). Related to the make-up of examinations are the conditions under which examinations are conducted, such as the length of examinations and what resources students have access to during examinations.

In a review of introductory programming assessment, Daly and Waldron (2004) suggest allowing "students to bring in a handwritten A4 'cheat-sheet' which can contain whatever they want. The process of creating the 'cheat-sheet' may also be educational" (p. 211). Daly and Waldron do not mention how they believe cheat-sheets are educational. No studies have reported on the use of cheat-sheets in computing education examinations.

1.4 Research Questions

In real-world circumstances, programmers rely on resources for specific information, such as syntax specifications and examples of solutions to problems. While expert programmers possess a wealth of tacit solutions to problems (Soloway, 1986), they are not expected to memorise specific information, so it is unrealistic to expect students to do so for an examination.

Student created cheat-sheets may overcome the need for memorisation and bring about other benefits, but this idea needs to be explored and analysed. To achieve this, the following research questions are proposed.

- Do students who create and use a cheat-sheet perform better than students who do not?
- Does a student created cheat-sheet lift a student's performance compared to earlier course assessment?
- What features can be identified on student created cheat-sheets?
- Do these identifiable features relate to examination performance?

2 Methodology

In order to answer the above research questions, an analysis is to be performed on the cheat-sheets created by students and used in an examination.

2.1 Setting

The examination was conducted at the end of an introductory programming course run at the University of Southern Queensland. There were 89 students who sat the examination including a mix of on-campus (21%) and external (79%). Students sat the examination at numerous examination centres around the world.

Leading up to the examination, students were provided with a sample examination that mirrored the style of the final examination but included different questions. The sample examination included a reference to language specific functions, similar to what students had used during the course, but customised for the examination.

Students were informed about the topics covered by questions in the examination. Both the sample examination and final examination included a mix of code writing and short answer questions. All questions were new and could not be answered by simply copying from a cheat-sheet.

Students were informed about the conditions of the examination, including the ability for them to bring a student created cheat-sheet. The cheat-sheet requirements were specified as:

- hand-written;
- A4, double-sided; and
- containing any information they saw as relevant.

Students were told that the final examination would include a language reference, similar to the sample examination, so they need not include such information in their cheat-sheets.

Students were required to submit their cheat-sheet with their examination papers. The cheat-sheets were collected by examination invigilators, who had specific instructions to do so.

From an experimental perspective, there was no mechanism for ensuring that there would be groups with and without cheat-sheets and there was no attempt to control the membership of these groups when they emerged.

2.2 Method of Analysis

The analysis of cheat-sheets was conducted by:

1. identifying each sheet with a code number,
2. separating them from their accompanying examination answers, and then
3. identifying features in each (see Coding Scheme section below), which were recorded against the code numbers.

All feature analysis was conducted before comparing the use of cheat-sheets, and the contained features, against student performance.

2.2.1 Coding Scheme

Before attempting to identify features in all cheat-sheets, a subset of the cheat-sheets was examined and a number of common features were identified. No pre-existing schema was used. These features are described in Table 1. There were two categories of features: those that related to layout (how information was organised on cheat-sheets) and content (what information was found on cheat-sheets).

The features were then checked in the entire collection of cheat-sheets. Each of the features was identified in a simple binary fashion, being either present or absent.

3 Results

From the 89 students who sat the examination 72 cheat-sheets were collected, which indicates that 81% of students chose to create a cheat-sheet and 19% of students either chose not to create a cheat-sheet or missed the fact that they could.

3.1 Relative Performance

Relative performance of students with and without cheat sheets was measured. There was no control over how students fell into these groups, however the range in both groups' marks was roughly equal, as indicated by the minima and maxima in Table 2.

Layout Features	Dense	A cheat-sheet was *dense* when both sides of the paper were covered, leaving little vacant space. This is a measure of the amount of information on the paper and possibly the effort invested in creating the sheet.
	Organised	A cheat-sheet was *organised* when the space on the paper was compartmentalised to make fuller use of space, usually with boundaries and identifiers for compartments.
	Order matches course content	If the ordering of the content on the student's cheat-sheet followed the ordering of content presented in the course, this feature was seen as present.
Content Features	Code examples	Relevant to a programming examination, the presence of program code examples in the cheat-sheet was measured.
	Abstract representations	Marked as present when concepts were represented in a general way, using text or diagrams rather than program code for specific examples.
	Sample answers	Where students included answers to sample examination questions, this feature was considered present.
	Language reference duplication	This feature was marked as present when students included information that duplicated information provided in the language reference in the paper.

Table 1: Features measured in cheat-sheets

	Min	Max	Mean	StdDev
With cheat-sheet	3	47	32.5	11.9
Overall	0	47	29.0	13.2
Without cheat-sheet	0	47	23.9	15.7

Table 2: Performance with and without cheat-sheets

In a t-test for equality of means, with roughly equal variance, there was a significant difference between the two groups (t=2.5, p=0.016). This indicates that students who produced and used a cheat-sheet performed significantly better than those who did not.

The overall average performance in the examination was 29.0 out of 50 possible marks. Students with cheat-sheets performed, on average, higher than this mean and students without cheat-sheets performed worse, as shown in Table 2.

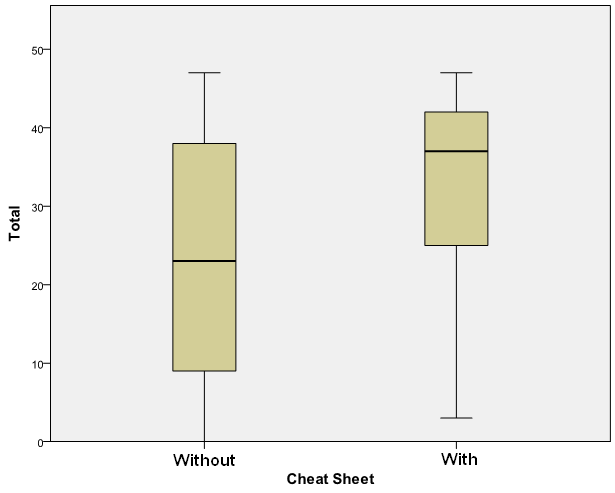

Figure 1: Comparison of performance by students with and without cheat-sheets

The distribution of marks by each group (with and without cheat-sheets) is represented in Figure 1. The box-plot on the left relates to students without cheat-sheets. This group had a lower mean at 23.9. The box-plot on the right shows the performance of students who created and used a cheat-sheet. This group had a higher mean at 32.5 and generally performed above a "passing" mark of 25, with a narrower standard deviation (see Table 2).

3.2 Improvement

Improvement was measured by comparing each student's performance in prior assessment (relative to the mean prior assessments) and their performance in the examination (relative to the mean examination mark).

On average, students who used a cheat-sheet improved their performance, with the opposite effect demonstrated by students without cheat-sheets, as shown in Table 3. In other words, preparing and using a cheat-sheet helped students to improve their level of performance between prior assessment and the examination.

	Examination Marks Improvement	Overall Assessment Improvement
With cheat-sheet	+5.3	+2.6%
Without cheat-sheet	-2.2	-1.1%

Table 3: Improvement in performance by students with and without cheat-sheets

3.3 Identified Features

In the 72 cheat-sheets collected, features were identified. A total of seven common features were identified in students' cheat-sheets. The list of features is given in the Coding Scheme section above, with a description of how each feature was identified. The occurrence of each feature is given in Table 4 together with the measured impact of each feature.

		Occurrences	Difference to mean
Layout Features	Dense	43 (60%)	+1%
	Organised	49 (68%)	-2%
	Order matches course content	14 (19%)	+13%
Content Features	Code examples	52 (72%)	-7%
	Abstract representations	34 (47%)	+21%
	Sample answers	15 (21%)	-30%
	Language reference duplication	7 (10%)	+3%

Table 4: Occurrence and impact of identified features

The analysis of features shows that the presence of some features relates to higher or lower examination performance, on average, by the creators of the cheat-sheets that contain them. Other features do not seem to relate to a difference in performance. The impact of each feature was calculated by comparing the average of the group of students whose cheat-sheets exhibited that feature with the overall mean and finding the difference.

In layout features, the density and organisation of information in a cheat-sheet did not seem to relate to any major difference in performance. Students who ordered the content of their cheat-sheets to match the ordering of course content performed, on average, 13% better than the mean.

The analysis of content features showed that students included code examples as well as abstract representations of content presented in the course. While some students included both (22%), most students included only one or the other. Students who included code examples in their cheat-sheets tended to perform slightly lower than average. Students who included abstract representations of concepts tended to perform 21% higher than the average.

Ignoring advice to the contrary, some students included information already provided in a language reference. Despite the fact that this content took up space that could have been dedicated to other content, such students did not seem to be negatively impacted by doing so.

4 Conclusions

The findings of this study indicate that the preparation and use of student created cheat-sheets does have an impact on student performance. This contradicts the findings of Dickson and Miller (2005) and provides evidence that supports the suggestions of Erbe (2007) and Daly and Waldron (2004) that the process of creating a cheat-sheet can improve student outcomes.

This study found variety in features in student's cheat-sheets, which echoes the findings of Visco et al. (2007). Ordering cheat-sheet content to match course content relates to higher examination performance. This may indicate that students who create cheat-sheets in such an ordered fashion are undertaking a more thorough, start-to-finish approach when creating their sheets, and perhaps learning more from this experience. It may also be the case that when content was ordered in a way that was familiar to the student's experience, less effort was required to find information on the cheat-sheet during the examination, which relates to the concern of Boniface (1985) who suggested that time used referring to information in examinations degrades performance.

Students who included abstract representations of content in their cheat-sheets were more successful. This may be due to abstract representations being more adaptable to new problems than specific examples. In order to include and use abstract representations students would need to have reached the higher SOLO relational or extended abstract levels (Biggs & Collis, 1982), while students who relied on coding examples may be working at the lower multistructural level. Answering at a higher SOLO level has been related to understanding in introductory programming (Lister et al., 2006).

The inclusion of sample examination answers in cheat-sheets was related to poorer performance. It seems likely that students who did this hoped that questions in the final examination would be the same as questions in the sample examination, which was not the case. This seems to reveal a poorer understanding of concepts by students. The sample examination was intended to be a test of student learning after revision and not the only instrument for revision itself.

Where students included material that duplicated what was available in the language reference included in the examination paper, there was no major impact on performance, even though the space used by this content could have been used for other, more necessary content. This perhaps indicates that revising such references may have aided student understanding, and having their own descriptions of such references may have made them easier to look up and use.

4.1 Recommendations

From the results of this study, it is recommended that examiners consider allowing students to use cheat-sheets in examinations as this may increase learning, reduce anxiety and, as shown in the results of this study, lead to improved performance.

When creating their cheat-sheets, students should be advised to:

- conduct a thorough review and order the content of their sheet to match the ordering of course content;
- attempt to record generalised, abstract representations of concepts, rather than specific examples, so that ideas can be adapted during examinations; and
- avoid hoping that answers to sample examinations will match final examination questions.

4.2 Future Work

The findings of this study would benefit from wider analysis in other introductory programming courses with more control over experimental group membership. It would be interesting to see if the findings of this study continued longitudinally. For more generally applicable results, experimentation in other knowledge domains would be needed.

5 References

Bacon, F. (1969): Open book Examinations. *Education and Training,* **11**(9):363.

Bagert Jr., D. J. (1988): Should computer science examinations contain "programming" problems? *Proceedings of the nineteenth SIGCSE technical symposium on Computer science education (SIGCSE'88),* Atlanta, Georgia, USA, February 25-26, 1988. 288 - 292.

Biggs, J. B., & Collis, K. F. (1982): *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. New York, Academic Press.

Bloom, B. S. (1956): *Taxonomy of Educational Objectives*, Edwards Bros., Ann Arbor, Michigan.

Boniface, D. (1985): Candidates' use of notes and textbooks during an open-book examination. *Educational Research,* **27**(3):201 - 209.

Daly, C., & Waldron, J. (2004): Assessing the Assessment of Programming Ability. *Proceedings of the 35th SIGCSE technical symposium on Computer science education (SIGCSE '04),* Norfolk, Virginia, USA, March 3 - 7, 2004 210 - 213.

Dickson, K. L., & Miller, M. D. (2005): Authorized Crib Cards Do Not Improve Exam Performance. *Teaching of Psychology,* **32**(4):230 - 233.

Dickson, K. L., & Miller, M. D. (2008): Do Students Learn Course Material During Crib Sheet Construction? *Teaching of Psychology,* **35**(2):117 - 120.

Erbe, B. (2007): Reducing Text Anxiety while Increasing Learning - The Cheat Sheet. *College Teaching,* **55**(3):96 - 98.

Feldhusen, J. F. (1961): An Evaluation of College Students' Reactions to Open Book Examinations. *Educational and Psychological Measurement,* **21**(3):637 - 646.

Janick, J. (1990): Crib Sheets. *Teaching Professor*, **4**(6):2.

Kalish, R. A. (1958): An Experimental Evaluation of the Open Book Examination. *Journal of Educational Psychology*, **49**(4):200 - 204.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., et al. (2004): A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, **36**(4):119 - 150.

Lister, R., Simon, B., Thompson, E., Whalley, J. L., & Prasad, C. (2006): Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. *ACM SIGCSE Bulletin*, **38**(3):118 - 122.

Schumacher, C. F., Buztin, D. W., Finberg, L., & Burg, F. D. (1978): The Effect of Open- vs. Closed-Book Testing on Performance on a Multiple-Choice Examination in Pediatrics. *Pediatrics*, **61**(2):256 - 261.

Sheard, J., Carbone, A., Lister, R., & Simon, B. (2008): Going SOLO to assess novice programmers. *SIGCSE Bulletin*, **40**(3):209 - 213.

Sheard, J., Simon, Carbone, A., Chinn, D., Laakso, M.-J., Clear, T., et al. (2011): Exploring Programming Assessment Instruments: a Classification Scheme for Examination Questions. *Proc. Proceedings of the International Conference on Educational Research (ICER) 2011*.

Soloway, E. (1986): Learning to program = learning to construct mechanisms and explanations. *Communications of the ACM*, **29**(9):850 - 858.

Theophilides, C., & Koutselini, M. (2000): Study Behavior in the Closed-Book and the Open-Book Examination: A Comparative Analysis. *Educational Research and Evaluation*, **6**(4):379 - 393.

Tussing, L. (1951): A Consideration of the Open Book Examination. *Educational and Psychological Measurement*, **11**(4):597 - 602.

Visco, D., Swaminathan, S., Zagumny, L., & Anthony, H. (2007): Interpreting Student-Constructed Study Guides. *Proceedings of the 114th Annual ASEE Conference & Exposition*, Honolulu, Hawaii, USA, June 24 - 27, 2007. 1 - 9.

Weimer, M. (1989): Exams: Alternate ideas and approaches. *Teaching Professor*, **3**(8):3 - 4.

Whalley, J. L., Lister, R., Thompson, E., Clear, T., Robins, P., Kumar, P. K. A., et al. (2006): An Australasian Study of Reading and Comprehension Skills in Novice Programmers, using the Bloom and SOLO Taxonomies. *Proceedings of the Eighth Australasian Computing Education Conference (ACE2006)*, Hobart, Australia, January 2006. 243 - 252.

Yu, B., Tsiknis, G., & Allen, M. (2010): Turning Exams Into A Learning Experience. *Proc. Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE2010)*.