The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42976

Assessing the Impact of Open-Resource Access on Student Performance in Computer-Based Examinations

Dr. Zulal Sevkli, Miami University

Dr. Zulal Sevkli has been serving as an Associate Teaching Professor in Department of Computer Science and Software Engineering at Miami University since 2021. Dr. Sevkli's professional focus lies in evidence-based computer science education, as well as the application of bio-inspired metaheuristics and machine learning algorithms to develop decision support systems. Dr. Sevkli earned her Ph.D. in Computer Engineering from Gebze Institute of Technology in 2010. She has taught a wide range of courses across the computer science curriculum and supervised undergraduate and graduate research.

Assessing the Impact of Open-Resource Access on Student Performance in Computer-Based Examinations

Zulal Sevkli
Computer Science and Software Engineering
Miami University
Oxford, OH
sevkliaz@miamioh.edu

Abstract

This study explored the effects of permitting digital resource access during computer-based exams in the context of System Programming course. Two exam types were introduced to students in Fall 2022: closed digital resources and open-resource. Results showed no significant score increase with open-resource exams. However, a negative correlation emerged between scores and digital resource access, especially in the first exam. This correlation weakened in subsequent exams, suggesting changing content influenced resource usage. The findings suggest that open-resource exams, mirroring real-world scenarios, can alleviate student anxiety and foster practical learning. The findings encourage a reevaluation of digital examination practices.

Introduction

Exams are important instrument of student assessment, providing educators with valuable insights into students' comprehension, retention, and application of course material. Today, although educators use many forms of exams in engineering education, we can categorize them based on their format: paper-based and computer-based exams. With the rapid advancement of technology, computer-based examinations have gained popularity due to their efficiency and convenience. They eliminate the need for printing, distributing and collecting physical exam papers. Computer-based exams enable the integration of multimedia elements such as videos, animation, real time feedback and provide greater accessibility and accommodation options for students with diverse learning needs.

Accelerated by factors such as the Covid pandemic and advancements in supported software, digitalization has propelled many educational institutions towards the adoption of computer-based exams [1, 2, 5, 6]. Studies like those by Lappalainen et al. [1], who found improved outcomes by beginning with paper-based exams and continue with computer-based exams, and Grissom et al. [4], who reported higher success in writing recursive solutions through computer-based exams, underscore this trend. Deloatch et al. [15] further highlighted a preference for computer-based exams, citing perceived improvements in quality, speed, and anxiety reduction.

Computer-based exams present both opportunities and challenges, particularly in terms of technical stability and academic integrity.. For example, Rajala et al. [2] developed an examination platform for Java programming, integrating multiple-choice questions and restricting internet access prevent academic integrity violations. Other implementations have emphasized more security, employing methods like Linux USB sticks, special student accounts, or Safe Exam Browser software [10,11,12], dedicated computer labs, or bring-your-own-device (BYOD) strategies coupled with web-based IDEs and well-structured policies [8,13,14]. Various techniques and characteristics of computer-based exams, including regular programming exams in the lab , have been also explored. [3,9].

Another category for exams can be exam methodology such as closed-note exam versus open-note exam. Closed-note exams test students' ability to memorize and recall information without relying on external resources. This can be valuable in subjects where foundational knowledge and memorization are essential. Also, students are often prompted to engage in critical thinking and problem-solving skills to apply their understanding of concepts to novel situations. On the other hand, the restrictive nature of closed-note exams can contribute to increased pressure and anxiety among students, especially those who struggle with memorization or performance under timed conditions.[17] This can hinder their ability to demonstrate their true understanding and proficiency. Closed-note exams may promote a "cramming" culture where students focus on memorizing information temporarily to pass the exam, rather than engaging in long-term learning and retention of concepts. In many real-world situations, individuals have access to resources and tools to aid decision-making and problem-solving. Closed-note exams may not accurately reflect the conditions and demands of these environments, limiting their relevance in preparing students for future academic and professional endeavors.

This study explores the design, implementation, and evaluation of computer-based examination methods within the context of the System Programming course. We implement a BYOD approach similar to those found in the literature, combining web-based IDEs with Learning Management Systems (LMS) and web-based proctoring software. From the literature, we understand that open-note and close-note are somehow different from each other and there are no enough studies on comparing of these two types of exams’ impact on student performance. The subsequent sections delve into the specific course content and structure, detailing the examination process tailored to this course. Section 2 outlines the system programming course content and its assessments. Section 3 explains exam design methods for System Programming and lists the research questions. The study findings are detailed in Section 4, while Section 5 concludes the paper.

Course Content and Structure

The System Programming course is designed for students who have already completed Introduction to Programming and Object-Oriented Programming courses. As the foundational course for system-based courses, its goal is to acquaint students with the underlying processes of computer systems.

The main topics include command prompt Linux commands, modular programming with C++ on the Linux platform, dynamic memory management, memory hierarchy, number systems and data representation, basic computer architecture, CPU structure and instruction sets and network programming and database programming through SQL. While students are expected to have Java programming skills from prerequisite courses, the introduction of the Linux platform and the C++ language, combined with numerous topics, can present a steep learning curve.

As a core sophomore-level course in Computer Science Department, it enrolls 150-200 students each semester. In the Fall semesters, there's a higher concentration of sophomore students from the Computer Science Department. In contrast, the Spring semesters see a more varied mix of sophomore, junior, and senior students from both the Computer Science and Electrical and

Computer Engineering majors. Student performance has been evaluated through various assessments, each contributing to the final grade:

- Reading assignments (20%)
- Lab assignments (20%)
- Project assignments (20%)
- In-class exams (20% for in-semester, 20% for the final exam)

Students utilize an interactive online textbook zybook [16] which contains many small programs and short formative reading assessments. Fourteen reading assignments are distributed throughout the semester.

Students attend mandatory 2-hour lab sessions each week, working on practical assignments in pairs under the supervision of the professor and teaching assistants. They must submit their work the next day, providing ample time for review without undue time pressure.

The whole course topic is divided into four modules, with students completing three or four reading and lab assignments before embarking on an individual project related to that module. Projects are assigned with a 10–14-day completion timeframe.

Each module concludes with an in-class exam, amounting to four in-semester exams in total. Up to the in-class exam, students evaluate their learning through diverse assessments in a supportive environment, benefiting from time flexibility, professor assistance, and peer support. The final assessment, the in-class exam, is conducted individually within a set timeframe.

The next section will delve into the specific design and structure of the exams for this course.

Method

Computer-based exams were introduced to the course in the Spring of 2022. The feedback from students on the end-of-semester course evaluation survey led to the formulation of two research questions :

1. Is resource restriction necessary during computer-based exams?
2. If we allow students to access online resources during the exam, is there a link between students' scores and their frequency of resource access during exams?

To find the answer to these questions, two computer-based exam types were introduced in Fall 2022 (next semester):

- *First Type (Closed Digital Resources):* Exam questions were created into the Canvas LMS platform. This was achieved using two parts; the first consisted of multiple-choice and true-false questions, while the second required short answers and the designing of partial solutions using C++ on Canvas. Note that Canvas does not provide a special editor for coding but a simple textbox. Students could navigate only forward for the first part and forward and backward through the questions in the second part and were allowed to use handwritten notes during the test. No other restrictive software or options were used other than Honor system that students may be trusted to act in accordance with an honor system and to formally or informally attest that they have not violated academic integrity.[7]
- *Second Type (Open Digital Resources):* It had also two sections: The first section’s questions of the first type exam were copied on Canvas. The second part utilized an interactive zybook platform for designing C++ solutions with live coding, compiling, and debugging features. Students could test their solutions using predefined test cases and were able to see their errors or scores of this part during the exam. The list of acceptable online resources - “whitelisted” web resources [7]- were also provided, including the online zybook textbook,

lecture notes on Canvas, code from lab assignments, and two C++ reference websites (https://cplusplus.com/reference/ and https://en.cppreference.com/w/). To ensure compliance with allowed resources, a proctoring software embedded on Canvas recorded students' network traffic and screens without restricting their navigation during the exam but generating post-exam reports for violations. Prior each exam, practice question sets were also provided for the live coding portion, along with an exam guide detailing study topic.

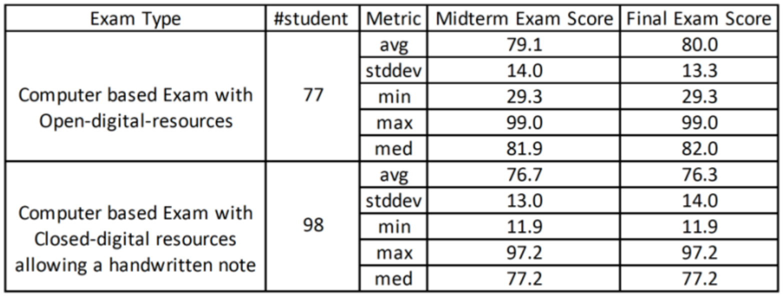
Both type of computer-based exams conducted in the classroom. Students used their own devices. They were responsible to install or keep up-to-date necessary software (Chrome browser, proctoring plug-in to browser, authentication for Zybook and Canvas and wireless connection) and bring hardware utilities such as charge adaptor. To practicing their device if it is ready to exam, mock exams were delivered on Canvas prior the real exam day.

Quasi-Experimental Result

In Fall 2022, seven sections of system programming courses utilized Canvas for exams. These sections were taught by two faculty members, with one faculty member teaching four sections (considered control condition) and the other teaching the remaining three (considered experimental condition). The faculty member who taught four sections administered closed-digital resource exams allowing handwritten notes, while the faculty member who taught three sections administered open-whitelisted digital resource exams. For the latter, proctoring software tracked website access. The coordination of these seven sections was a collaborative effort between the two faculty members, who worked together to prepare the exam questions.

All students took the exams in the same week, with a total of four computer-based in-class exams conducted roughly every three and a half weeks. Once students submitted, first part of the both types of computer-based exams were graded automatically while second part were graded manually by the instructors. Although live-coding part on the open digital resources type exam were graded automatically, instructors overviewed the errors and may give partial points depends on the error types on the students' programs.

Table 1: Descriptive statistical data for two types of Computer-based exams for the system programming course

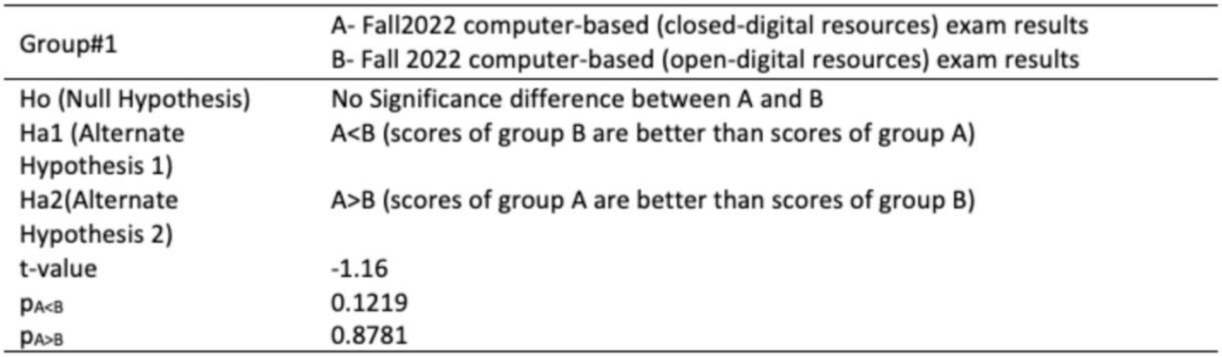


Exam Type	#student	Metric	Midterm Exam Score	Final Exam Score
Computer based Exam with Open-digital-resources	77	avg	79.1	80.0
		stddev	14.0	13.3
		min	29.3	29.3
		max	99.0	99.0
		med	81.9	82.0
Computer based Exam with Closed-digital resources allowing a handwritten note	98	avg	76.7	76.3
		stddev	13.0	14.0
		min	11.9	11.9
		max	97.2	97.2
		med	77.2	77.2

To analyze the exam results, we computed the average, standard deviation, and median values for two distinct types of exams and demonstrated in Table1. The results of exams where open resources were allowed indicate that the median score of three sections is higher than the average, showing that most students scored above the average. In contrast, the closed-resource exam exhibited nearly identical average and median values of four sections. Still, both were lower than those of the open-resource exam, suggesting better performance in the open-resource setting. Table 1 also shows the min scores achieved in each exam type. It’s important to mention that the lowest scores were higher in exams with open resources. This aspect can potentially motivate students and increase retention rates

We also conduct independent samples t-test at the 0.05 significance to compare the means of the computer-based exams with closed versus open-digital resources to see if they are statistically different or similar. The result of the t-test in Table 2 shows that both p-values (0.1219 and 0.8781) are greater than the significance level of 0.05. We can conclude that there is no significant difference in scores between the two exam types. In other words, computer-based exams permitting open-digital resources don't necessarily lead to higher scores for students, contrary to what instructors might have anticipated. This suggests that while access to digital resources mirrors real-life scenarios, it doesn't significantly impact students' grades but providing decreased pressure and anxiety among students in the context of these exams.

Table 2- t-test results with the 0.05 significance for the two different exam types



Group#1	A- Fall2022 computer-based (closed-digital resources) exam results B- Fall 2022 computer-based (open-digital resources) exam results
Ho (Null Hypothesis)	No Significance difference between A and B
Ha1 (Alternate Hypothesis 1)	A<B (scores of group B are better than scores of group A)
Ha2(Alternate Hypothesis 2)	A>B (scores of group A are better than scores of group B)
t-value	-1.16
$p_{A<B}$	0.1219
$p_{A>B}$	0.8781

In response to the second question, Figure 1 illustrates the scatter plots of the number of accesses of whitelisted open resources over exam score for four exams.

To determine the type of correlation between exam scores between 0 and 50 and number of times students accessed the digital resources, we compute the Pearson correlation coefficient (r can range from -1 to 1) which measures the linear relationship between two datasets. The correlation coefficient for Exam 1 data is -0.38. This indicates a moderate negative linear relationship between Exam1 scores and the number of times students accessed the digital resources. In other words, as exam scores increase, the number of times students accessed the digital resources tends to decrease, and vice versa. From the scatter plot of Exam 1 in Figure 1, we can visually confirm the negative trend. Students who scored lower on the exam generally accessed the digital resources more frequently than those who scored higher.

When we calculate the correlation coefficient (r) for the subsequent exams (Exam 2, Exam 3, and Exam 4), the results are -0.12, -0.03, and -0.18, respectively. Note that p-values of each correlation analysis are below 0.05. For all three exams, there's a weak negative correlation between the exam score and the number of times students accessed the digital resources. However, the relationship is not strong in any of the cases. The weakest relationship is observed in Exam 3, while Exam 4 shows a slightly stronger negative relationship than Exam 2. This variability in results could stem from the diverse topics covered in the exams. While students had prior experience with the topics covered in Exam 1 from previous programming courses, subsequent exams introduced new system-based subjects such as networking, memory hierarchy, and database programming, which were novel to students. Consequently, students might have felt compelled to access resources during the exam, irrespective of their performance level. It's worth noting that these four exams were spaced four weeks apart throughout the semester. The assimilation of new topics may require additional time over the semester. Therefore, regardless of their success rate in the class, students may have felt more comfortable checking their knowledge with accessible resources during the exam

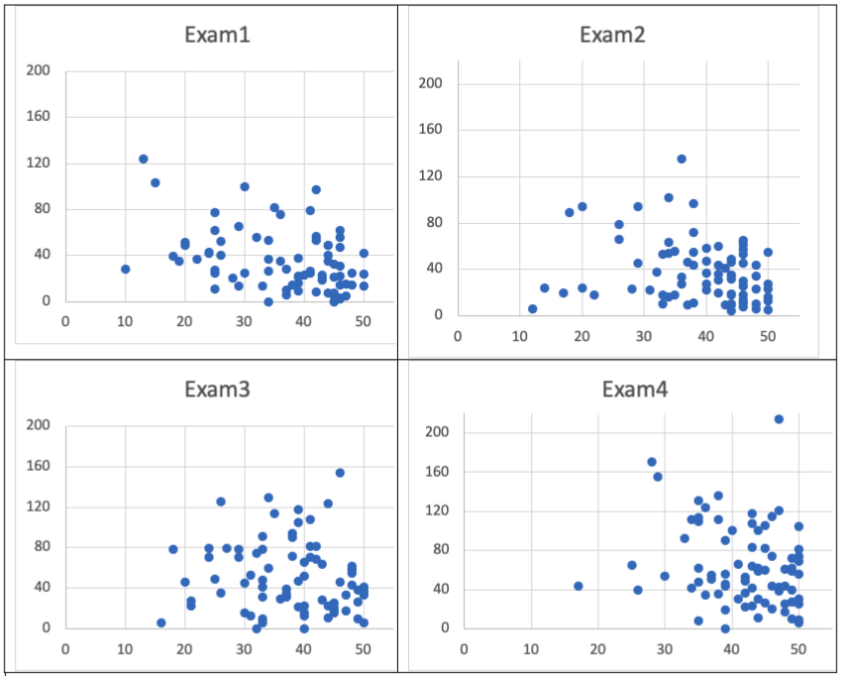


Figure 1 :The scatter plots of the number of accesses of whitelisted open resources(axes-y) over exam score (axes-x) for Four Exams in Fall 2022

Conclusion and Future Work

This study examined the impact of allowing digital resource access for computer-based exams. A well-designed exam should strike a balance between being appropriately challenging—neither too easy nor too difficult—and providing students with the opportunity to demonstrate their knowledge effectively in a warm classroom presence.

From the faculty's perspective, efficient grading processes are essential, especially in large classrooms. The exam platform should mirror everyday working environments to facilitate students' ability to recall and interpret information. Furthermore, it should offer multiple channels for faculty feedback and enable easy analysis of student performance data.

Our study demonstrates how to address both student and faculty expectations in exam design and administration. We opted for a Learning Management System (LMS) platform to develop our computer-based exam and incorporated the Proctorio plugin for low-stakes monitoring, which records student network traffic and screen activity without restricting their navigation. Students were allowed access to whitelisted resources during the exam, which lasted one hour to accommodate laptop battery life. The exam utilized a web-based programming environment and securely stored multiple versions of student submissions on the cloud to mitigate any potential technical issues or internet connectivity disruptions.

Our exam questions were crafted to require a blend of recall and critical thinking skills, as well as executable programming solutions. Contrary to our initial expectations, allowing open access to digital resources did not lead to significantly higher student scores compared to closed-note exams. Analysis of network traffic data revealed a moderate negative correlation between exam scores and the frequency of digital resource access, particularly noticeable in the initial exam. However, this correlation diminished in subsequent exams, underscoring the influence of exam content and students' prior knowledge.

Importantly, the overall class average remained consistent regardless of whether exams were open-resource or closed-resource. This consistency underscores the depth and complexity of the exam questions, which went beyond mere information retrieval and demanded advanced cognitive skills like analysis and synthesis. For students, open-digital resource exams can reduce test anxiety, as they aren't solely reliant on memorization and can access information as they would in real-world situations. This approach fosters a more practical and application-based learning environment. Additionally, it prepares students for real-world challenges, where problem-solving often involves seeking and applying information rather than relying purely on recall.

While this study provides valuable insights into the relationship between exam types and student performance, several limitations should be acknowledged. Firstly, the assignment of students to exam types was not fully random but rather based on the faculty's assignment to the sections. In future studies, more rigorous methods could be employed to ensure random assignment, potentially enhancing the validity of the findings.

Additionally, to gain a comprehensive understanding of the impact of different digital exam types on student learning and behavior, our future studies could incorporate student feedback through surveys. These surveys could assess changes in study habits and anxiety levels in response to different exam formats. Understanding how students adapt their study strategies and cope with exam-related stress could provide valuable insights for designing more effective assessment methods in the future.

In summation, this study enriches the growing discourse on digital examination techniques. It affirms that innovative exam methods, including the likes of open-resource exams, can be woven into the educational assessment without compromising academic integrity or student performance. By presenting detailed analyses, statistical comparisons, we aim to contribute to the broader discussion on adapting computer-based examination methods in computer science education. We hope this study gives instructors confidence to apply open-resources online exam and prompts further discussion and development of best practices.

References

[1] Vesa Lappalainen, Antti-Jussi Lakanen, and Harri Högmander. 2016. *Paper-based vs computer-based exams in CS1*. In Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16). Association for Computing Machinery, New York, NY, USA, 172–173.

[2] Teemu Rajala, Erkki Kaila, Rolf Lindén, Einari Kurvinen, Erno Lokkila, Mikko-Jussi Laakso, and Tapio Salakoski. 2016. Automatically assessed electronic exams in programming courses. In Proceedings of the Australasian Computer Science Week Multiconference (ACSW '16). Association for Computing Machinery, New York, NY, USA, Article 11, 1–8.

[3] Gabriela Silva-Maceda, P. David Arjona-Villicana, and F. Edgar Castillo-Barrera, *More time or better tools? a large-scale retrospective comparison of pedagogical approaches to teach programming*, IEEE Trans. on Educ., vol. 59, no. 4, pp. 274–281, 2016

[4] Scott Grissom, Laurie Murphy, Renee McCauley, and Sue Fitzgerald, *Paper vs. computer-based exams: A study of errors in recursive binary tree algorithms*, in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education, SIGCSE '16*, pp. 6–11, ACM, New York, NY, USA, 2016.

[5] Ben Stephenson, *An Experience Using On-Computer Programming Questions During Exams*, in *Proc. Western Canadian Conference on Computing Education, WCCCE 2018*, ACM, 2018.

[6] Craig Zilles, R. Timothy Deloatch, Jacob Bailey, Buhuwan B. Khattar, Wade A. Fagen-Ulmschneider, and Cinda Heeren, *Computerized testing: A vision and initial experiences*, in *Proc. ASEE Annual Conference and Exposition: Making Value for Society*, ASEE 2015, ASEE, 2015.

[7] Chris Bourke, Yael Erez, and Orit Hazzan. 2023. *Executable Exams: Taxonomy, Implementation and Prospects*. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 381–387.

[8] Charlie Daly and John Waldron. 2004. *Assessing the assessment of programming ability*. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education (SIGCSE '04).* Association for Computing Machinery, New York, NY, USA, 210–213.

[9] João Paulo Barros, Luís Estevens, Rui Dias, Rui Pais, and Elisabete Soeiro. 2003. *Using lab exams to ensure programming practice in an introductory programming course*. In *Proceedings of the 8th annual conference on Innovation and technology in computer science education (ITiCSE '03).* Association for Computing Machinery, New York, NY, USA, 16–20

[10] Mathew Hillier and Andrew Fluck, *Transforming exams - how IT works for BYOD e-Exams*, in *Proc. International Conference on Innovation, Practice and Research in the Use of Educational Technologies in Tertiary Education, ASCILITE 2017,* ASCILITE, pp. 100–105, 2017.

[11] Hugo Ribeiro and Margarida Amaral, *High Stake Exams with BYOD at the University of Porto: Bypassing Constraints*, in *Proc. International Conference of Education, Research and Innovation*, ICERI 2018, IATED, pp. 437–442, 2018.

[12] *Safe Exam Browser*, 2023, [Online]. Available: https://www.safeexambrowser.org/. Accessed: August 2023

[13] Teck Keong Seow and Swee Kit Alan Soong, *Students' perceptions of BYOD open-book examinations in a large class: a pilot study*, in *Proc. International Conference on Innovation, Practice and Research in the Use of Educational Technologies in Tertiary Education*, ASCILITE 2014, ASCILITE, pp. 604–608, 2014.

[14] Oka Kurniawan, Norman Tiong Seng Lee, and Christopher M. Poskitt, *Securing Bring-Your-Own-Device (BYOD) Programming Exams*, in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, ACM, New York, NY, USA, pp. 880–886, 2020.

[15] Robert Deloatch, Brian P. Bailey, and Alex Kirlik, *Measuring Effects of Modality on Perceived Test Anxiety for Computer Programming Exams*, in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE '16, ACM, New York, NY, USA, pp. 291–296, 2016.

[16] *Zybook*, 2023 Available : zyBooks.com, accessed 2023.

[17] Beth Johanns, Amber Dinkens, Jill Moore, A systematic review comparing open-book and closed-book examinations: Evaluating effects on development of critical thinking skills, Nurse Education in Practice, Volume 27, 2017, Pages 89-94, ISSN 1471-5953, https://doi.org/10.1016/j.nepr.2017.08.018.