The Eurasia Proceedings of
Educational & Social Sciences (EPESS)
ISSN: 2587-1730

The Eurasia Proceedings of Educational & Social Sciences (EPESS), 2014

Volume 1, Pages 341-345

ICEMST 2014: International Conference on Education in Mathematics, Science & Technology

EVALUATION OF ALGORITHM IMPLEMENTATION ASSESSMENT METHODS BASED ON A DATA STRUCTURE COURSE

Kamil Yurtkan
Cyprus International University

Cagin Kazimoglu
Cyprus International University

Umut Tekguc
Cyprus International University

ABSTRACT: Implementing algorithms and making abstractions are two integral parts of computer programming. Algorithm implementation process involves understanding a business problem, designing the most appropriate solution and abstracting this in a programming environment. Thinking within the syntax of a programming language and generating algorithms simultaneously are often found challenging by students. More importantly, majority of students do not necessarily understand the underlying reasons behind the multiple assessment methods used in the evaluation of algorithm implementation. Some students believe that the theoretical measurements for algorithm implementation are not directly related to the practical development of solutions and hence, they do not see the point of theoretical exams particularly in algorithm implementation process. In this paper, we analyze the methodologies used for the evaluation of algorithm implementation. The Data Structures and Data Organization course thought at the Faculty of Engineering in Cyprus International University (CIU) has been selected as a pilot course to perform a rigorous study in order to compare the theoretical and the practical exam results of students. The aim of the study was to understand whether or not there was a significant relationship between the theoretical and practical exam results. The raw data of the study came from 100 students who were randomly selected without considering their previous background or programming knowledge. The correlation between students' programming capabilities and their theoretical knowledge were analyzed in order to state whether or not their performance in theoretical exams are authentic when compared to their practical exams.

Key words: Algorithm implementation, programming assessment, assessment methodologies, data structures, c programming.

INTRODUCTION

Majority of institutions in the Cyprus follow a common academic grading system which offers the options of using quizzes, theoretical exams (e.g. written exams), practical exams (laboratory exams), oral exams and/or project development as major knowledge assessment methods. The variation of the academic grading system provides academicians flexibility in measuring students' knowledge and skills both in theory and in practice. Despite a rich variation in the academic grading system provides many advantages, it also brings considerable problems as the balance among the assessment methods might be diverse. As multiple assessment methods are used to measure students' ability and knowledge, it is arguable whether or not those students who did well in the theoretical exams would also do well in the practical exams. Although this is a generic problem in academy, we have observed that a number of students are suffering particularly in algorithm development courses as they believe the theoretical part of algorithm development is independent from practical solution development. Furthermore, various studies discussed these issues and offered new instructional design and assessment methods in order to measure students' ability and knowledge in computer algorithms accurately. (Ala-Mutka, 2005; Barros *et al.*, 2003; Chamillard & Braun, 2000; Daly & Waldron, 2004).
In this study, we focus on the evaluation of assessment methods for an algorithm implementation course particularly the "Data Structures and Data Organization" course taught in the 5th semester of Computer

- Selection and peer-review under responsibility of the Organizing Committee of the conference

*Corresponding author: Kamil Yurtkan- e-mail: kyurtkan@ciu.edu.tr

Engineering, Information Systems Engineering and Management Information Systems departments of Cyprus International University (CIU).
The rest of the paper discusses a) a rigorous study and its objective; b) the methods used in the statistical analyses of the rigorous study; c) distribution of data and d) the statistical results along with their discussion. The paper concludes with future work based on the statistical results obtained.

A RIGOROUS STUDY and ITS OBJECTIVE

A rigorous study was undertaken to measure the correlation between the assessment methods used in measuring students' knowledge in an algorithm implementation course particularly the Data Structure and Data Organization course thought at CIU. The study had two main purposes. Firstly, it was aimed to investigate whether or not the practical exams and theoretical exams were genuinely evaluating students' knowledge at the same level of accuracy in terms of programming and algorithm implementation. Secondly, it was aimed to analyze the correlations between the algorithm development projects and the theoretical exams. As a result of this, it was intended to investigate whether or not the results of the theoretical exams were correlated with the results of the practical exams.

EXPERIMENTAL BACKGROUND

The study is performed using the IBM software package, SPSS, which is used for statistical analysis particularly in social studies. In order to perform the statistical analysis, a total of 100 students' exam/project results were randomly selected from the Computer Engineering, Information Systems Engineering and Management Information Systems Departments at CIU. As the students studied in different semesters, there had been some absence in their results due to a variety of reasons. Hence, only 84 valid responses were gathered in this study. The performances of the students in the theoretical exams were compared with the practical performances of the same students. The identities of the randomly selected students were kept confidential in order to provide an anonymous study.

Table 1. Stack implementation questions contained in midterm exams of the semesters observed.

Semester	Stack Implementation Question
Fall 2011 - 2012	**Q1)** Write **a function** that takes **3 stacks**, **p**, **gr** and **ls**, as arguments and updates the values of the two stacks **gr** and **ls** as follows. The stack **gr** contents are the elements of the stack **p greater than or equal to 50** (>=50), and the stack **ls** contents are the elements of the stack **p less than 50** (<50).
Fall 2012 - 2013	**Q1)** Write a **C/C++ function** that performs **search** operation in the stack. If the search key is found, function should **return 1**, else **return 0**. Stack should **preserve** the contents after the search. **Q2)** Write **a C/C++ function** that takes **2 stacks, p and q**, as arguments and derives the contents of **q** according to **p** such that if the element in p is **greater than or equal to 10**, the corresponding element in q should be **1**, otherwise **0**.
Fall 2013 - 2014	**Q1)** Write a **pseudo code** to separate **even** and **odd** numbers and push them to different stacks. **Q2)** Write **a C/C++ function** that takes **2 stacks, p and q**, as arguments and copies the contents of **stack p** to **q.**

As shown in Table 1, exam results were obtained from three different semesters which are 2011-2012, 2012-2013 and 2013-2014 Fall semesters. For each semester, two analyses have been performed. The first test was performed on stack implementation questions that were matched with a practical lab exam that involved a similar type of question (Table II). Consequently, the second test was performed between the midterm exam and the project which was then graded with an oral exam (Table III). The topics of the implementation questions were the stack implementation which is a well-known topic of data structures. Thus, despite being on different semesters, all midterm questions were based on the same subject. The theoretical and practical exams were all graded by pair reviews in order to keep a fair evaluation.

Table II demonstrates the questions asked to students in their practical exams as the result students obtained from these questions were used in the first test (i.e. comparing midterm exam results with practical exam results).
Table III demonstrates the question asked to students in their project. The results students obtained from this project were compared to stack implementation question results listed in Table 1. As shown from below, the project included implementations related to arrays and structures which then were tested with an oral exam. The questions asked to students in the project were matched with the midterm exam and the result students obtained from these were used in the second test.

Table 2. Lab Final Exam questions asked to students at the end of the observed semesters.

Semester	Lab Final Question
Fall 2011 - 2012	**Q)** Write the following function to print the content of stack and test with the given stack implementation. **void printstack (struct stack * ps);**
Fall 2012 - 2013	**Q)** Write the following function to find the size of stack and test with the given stack implementation. **int sizestack (struct stack * ps);**
Fall 2013 - 2014	**Q)** Write the following function tocount even numbers of stack and test with the given stack implementation. **int countevenstack (struct stack * ps);**

Table 3. Sample project definition.

Sample Project Definition	
Write a complete C or C++ program to implement a phonebook structure. Structure contains id, name, surname and phone number of a person. Your program should have the following menu. Write necessary functions for the menu: 1- To add a person to phonebook 2- To delete a person from phonebook 3- To list data in the phonebook 4- To search for a record 5- To update phoneNo of a record 6- To sort according to ID 7- To quit	struct phone_book { int ID; char name[20]; char surname[20]; char phoneNo[20]; }phone_list[100];

DISTRIBUTION OF DATA

Figure 1 illustrates the distribution of data gathered from the difference of students’ knowledge between the practical exam (i.e. lab final exam) and theoretical exam (i.e. stack implementation question). As it can be observed from the figure, the data came from a non-normally distributed population. The histogram has kurtosis issues as the observations are way over the normal distribution curve. Additionally, the histogram is skewed to the right which proves that the distribution of data is asymmetric. As a result, the histogram shows that the data came from a non-normally distributed population.

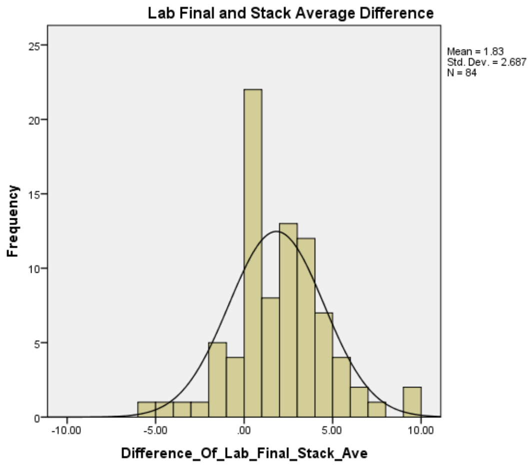

Figure 1. Data histogram showing the distribution.

Despite the fact that a histogram can provide a generic overview regarding the distribution of data, it is not a standalone reliable tool to measure the distribution of data. That is why, a Q-Q plot was generated in order to ensure regarding the distribution of data. As it can be observed from the Figure 2, the observations do not embrace the linear line on the Q-Q plot. The values at both ends are scattered and do not approach to the parallel line. Additionally, very few observations hug the linear line (i.e. the expected normal curve). This shows that the data came from a non-normally distributed population and hence, the result of the Q-Q plot supports the findings of the Histogram.

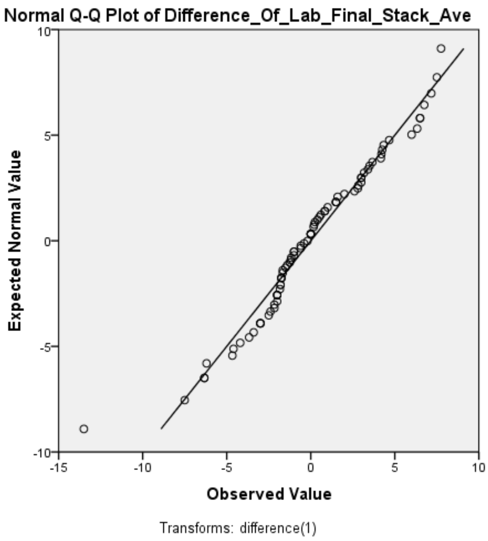

Figure 2. Quantile – Quantile (Q-Q) Plot.

RESULTS and FINDINGS

As the results show that the data came from a non-normally distributed population, it is required to use the Spearman's rank test in order to investigate the correlations between the student knowledge in test 1 (i.e. stack implementation question and lab final exam) and in test 2 (i.e. the midterm exam and the project implementation). As shown in Table IV, the Spearman's rank correlation coefficient regarding the first test was found to be significant and moderately strong ($r=0.636$; $p<0.001$). This means that, the students who performed well in the midterm exam for the stack implementation questions also did well in the lab final practical exam. As the correlation is found to be significant, modestly strong and positive, we have strong reasons to believe that the results of the algorithm implementation in the theoretical exams (i.e. midterm exam) are closely related to the results in practical exams (i.e. final exam). In other words, those students who solved stack implementation questions in the midterm exam also solved the same type of questions in the lab final exam.

Table IV. Correlations between stack implementation questions and lab final exams.

			Lab_Final	STACK_Ave
Spearman's rho	Lab_Final	Correlation Coefficient	1.000	**.636****
		Sig. (2-tailed)	.	.000
		N	84	84
	STACK_Ave	Correlation Coefficient	**.636****	1.000
		Sig. (2-tailed)	.000	.
		N	84	84

**. Correlation is significant at the 0.01 level (2-tailed).

Table V shows the Spearman's rank correlation coefficient regarding the second test. As shown from the table, the correlation among the pair was found to be significant and modestly strong ($r=0.535$; $p<0.001$). Although this is not as high as the first tests' coefficient, the correlation between the midterm exam and project is found to be significant, positive and moderately strong. This provides strong reasons to believe that those students who did well in the midterm exam also performed a similar performance in their projects.

Table V. Correlations between midterm exams and projects.

			Midterm	Project
Spearman's rho	Midterm	Correlation Coefficient	1.000	**.535****
		Sig. (2-tailed)	.	.000
		N	84	84
	Project	Correlation Coefficient	**.535****	1.000
		Sig. (2-tailed)	.000	.
		N	84	84

**. Correlation is significant at the 0.01 level (2-tailed).

CONCLUSION

The correlation between students' programming capabilities and their theoretical knowledge were analyzed on a data structure course. The results of the statistical analyses showed that there is a positive, significant and moderately strong correlation among the first test pair (stack implementation questions and lab final exams.) and the second test pair (midterm exam and project). In both cases, the results are similar, and therefore it is possible to conclude that the theoretical and the practical exams are related and assessing the same criteria. In other words, the statistical results provide strong reasons to believe that those students who have a theoretical understanding of algorithm implementation tend to have a good understanding in producing practical solutions through using algorithms. As future work, the finding of this study could be investigated further by conducting the study with a larger sample size. Although the sample size of this study (i.e. 84 valid responses) was enough to generate accurate results, a larger sample size could provide a more realistic and detailed distribution.

REFERENCES

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. Computer Science Education, 15(2), 83-102.

Barros, J. P., Estevens, L., Dias, R., Pais, R., & Soeiro, E. (2003, June). Using lab exams to ensure programming practice in an introductory prog. course. In ACM SIGCSE Bulletin (Vol. 35, No. 3, pp. 16-20). ACM.

Chamillard, A. T., & Braun, K. A. (2000). Evaluating programming ability in an introductory computer science course. ACM SIGCSE Bulletin, 32(1), 212-216.

Daly, C., & Waldron, J. (2004, March). Assessing the assessment of programming ability. In ACM SIGCSE Bulletin (Vol. 36, No. 1, pp. 210-213). ACM.