Brick and Mortar Education vs. SCORM-based Education in Computer-programming Courses: A Comparative Study

Mohammed Abu Shquier[1]

[1] Jerash University, Jordan

Correspondence: Mohammed Abu Shquier, Jerash University, Jordan.

Abstract

Online education has positively influences student performance during universities lockdown nowadays due to COVID-19, in fact both educators and students have proven their ability to develop their teaching skills by emerging several technological tools. This article analyses the performance of two cohorts of students, the first cohort was taught traditionally while the other was taught online, the scope of this study is the students enrolled in programming languages at the Faculty of Computer Science and Information Technology at Jerash University, the study was carried out between the years 2017 - 2020. 1210 students have participated in the study. This study investigates a comparative study between different methods of delivering programming-languages courses over the 3-year period, the study also aims to shed light on the impact of traditional methods on delivering computer-programming courses and how it could be improved by emerging a SCORM learning multimedia and other learning modules, activities and resources. Result shows that online delivering of courses with the use of SCORM and other tools improves students' scores and performance slightly, the article concludes that emerging technology to learning can improve the students' creativity, understanding and performance overall.

Keywords: e-learning, programming, Brick and Mortar, Jerash, SCORM, Moodle

1. Introduction

Brick and mortar education refers to the traditional way of teaching, by which students come to campus and attend classes physically, while the term e-learning is usually referring to the education take place online or over the internet, most of the universities use nowadays different platform of learning management systems such as Blackboard and Moodle. eLearning cut-off a lot of expenses and make teaching more flexible, accessible and even more convenient. On the other hand, eLearning may suffer from lower social interaction and lower retention rates among students, and sometimes it involves lack of focus and determination, different backgrounds of online learners is also another challenge.

Several techniques might be emerged with online learning such as SCORM which stands for Shared Content Object Reference Model, SCORM enable educators to merge different multimedia element in designing their courses (HTML, text, animation, image, video, audio, etc…), however, eLearning platform then connect all of these elements altogether to generate one hyperlink, this particular tools make students life much easier as the course will be displayed as a set of well-organized hyperlinks for each topic, technically, SCORM courses are designed using certain authoring tools (i.e., Lectora, Adobe Captivate, courselab, etc…) to merge and integrate all multimedia elements to produce the content, furthermore, the SCORM consists of three components: Content Aggregation Model, Run-Time Environment, and Navigation. These particular authoring tools enable the user to generate a zip file that is compatible with LMS platforms to be uploaded to learners.

In this paper, we have developed a SCORMs for our programming-language courses so as to allow further choices for students to learn with passion, moreover, the students could download to entire SCORM and browse the course offline, basically, SCORM helps in reaching more students since it supports sophisticated eLearning delivery, it also evaluated the students during their progress in navigating the content of the course.

2. Related Work

E-learning has increasingly grown and impact various educational aspects, usability and accessibility are key factors for successful e-learning adoption (Lam, Maria SW, et al, 2008)

According to Bakhouyi, A., et al, (2019) Learning management systems like Moodle is intended to improve the

learning process, as it reduces the cost and enhance the quality of learning (Bakhouyi, A., et. al., 2019).

Distante, D., et al, (2020) presented a new prototype for interactive Learning using SCORM, their prototype aimed at measure the user session duration at the University of Rome Unitelma Sapienza (Distante, D., et. al., 2020).

Computer programming as one of the most demanded skills nowadays will be considered at this study as a field that requires further understanding of coding, compiling, debugging and comprehending the integrated development environment (IDE) (Nahar, et al., 2016) (Cedazo, R., et al, 2015)

Learning anywhere anytime might be the best definition of e-learning in general (Ho: Dzeng, 2010), others, defined e-learning as any learning using technologies outside classrooms (Jama, P., 2008)

(Allen et. al., 2013) stated that the main barriers of online learning in higher education institutions are the lack of faculty member's acceptance to the idea, and the low retention rates of students, (Hentea et. al., 2003) concluded that students who are taking programming language courses online are more exposed to significant risk of failures, alternatively, they recommended embedded learning that merge both online and traditional methods.

On the other hand, several studies concluded that there were remarkable differences between online and traditional learning, most of these studies recommended the use of e-learning tools with almost every course by any means, they also share some disadvantages for such type of learning that includes lack of integration between students and (Ellis A., 2009)(Gholamhosseini, L., 2008)(Mason, R: Martin W., 2001)

3. Methodology

The SCORM-based education has been developed to assist the student throughout the entire educational process, it does not only provide them with the course materials (i.e., PPT/PDF slides, textbook, etc..) but it also provides them with an interactive tools that allow them to navigate among developed materials, such as videos, GIFs, and many other activities and resources. Furthermore, the interactive SCORM contents show the students the progress of running and debugging the code and highlight the errors and suggest possible solutions, furthermore, it redirects the students to the content where they have to look to get their errors solved.

On the other hand, we have set a number of modules to the e-course page, such as an adaptive online-quizzes with instant feedback, this particular module allows students to test their understanding and let them try several times with a certain hint per each incorrect answer.

Throughout the process of developing the SCORM of programming-language courses, we have considered certain points:

- What Course Learning Objective(s) (CLOs) should be met when an activity is added to the SCORM.
- Navigation throughout the contents as a pre-set sequence or freely.
- What goals should be targeted before progressing to the next content (i.e., section or chapter).
- Evaluate each topic before navigating to the next one.
- Usability and accessibility of contents considering the look-and-feel design.
- Responsive design to match the proper size of the device it opened with.

It is worthy stressing that SCORM could be integrated or merged with various LMS platforms:

- Moodle (PHP/MySQLi)
- Blackboard
- Sakai (Java)
- ILIAS (PHP/MySQLi)
- Teach-base
- SAP Success-Factors LMS
- Share-Point LMS

3.1 Study Design

In this study, we present a method to compare the performance of two groups of enrolled students in the same graduate program (i.e., Computer Science) based on their final grades, the first group was taking the course as Brick and mortar (traditional method) while the other took the course online using the SCORM-based learning tool, which has been emerged to Moodle, a screenshot on how SCORM is emerged with course activity and
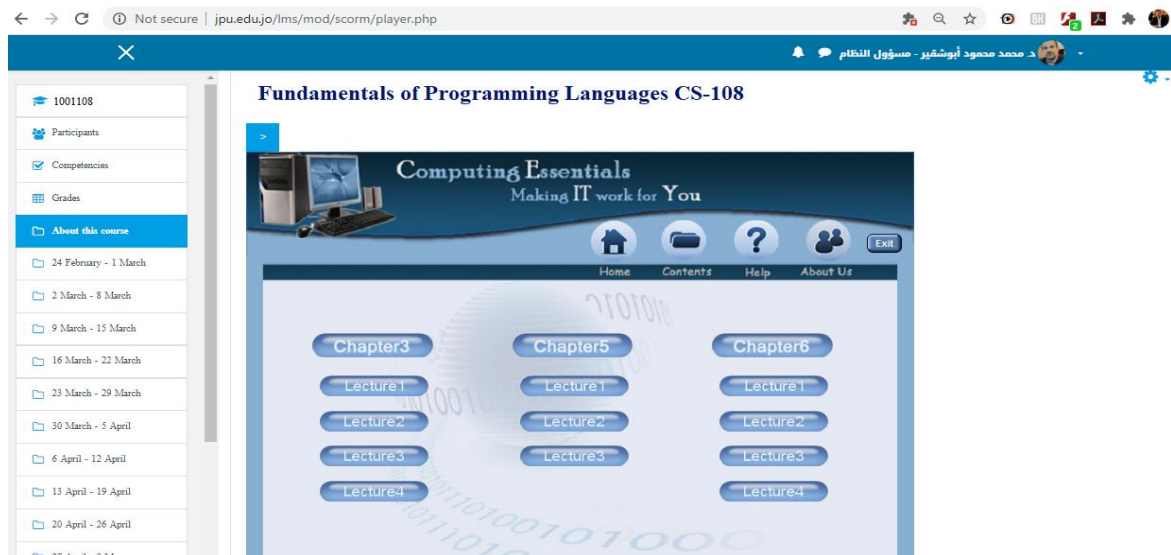
resources is shown in Figure 1.

Figure 1. SCORM Emerging with Moodle

4. Data Source

We have selected certain computer-programming courses from the department of computer science at the Jerash University for our data source as shown in Table 1, the reason behind selecting these courses is that they share the same pre-requisite programming skills and they were delivered in both methods mentioned above.

Table 1. Programming Languages Courses

Subject code	Course Symbol	Subject title
1001108	CS-108	Fundamentals of Programming Languages
1001131	CS-131	Computer Programming (I) – C++
1001233	CS-233	Web/App Programming – PHP, ASP.NET
1001328	CS-328	Computer Programming (II) – OOP using C#
1001329	CS-329	Selected Programming Language (III) - Python

The data under investigation has been taken during the academic years 2017/2018, 2018/2019 and 2019/2020. A summary of samples number and delivery method is shown in Table 2.

Table 2. Subject Enrolment in the Program

Course code	Course Title	Delivery method	samples (N)	Total
CS-108	Fundamentals of Programming Languages	Traditional	154	246
		Online	92	
CS-131	Computer Programming (I) – C++	Traditional	158	261
		Online	103	
CS-233	Web/App Programming – PHP, ASP.NET	Traditional	42	63
		Online	21	
CS-328	Computer Programming (II) – OOP using C#	Traditional	149	236
		Online	87	
CS-329	Selected Programming Language (III) - Python	Traditional	54	77
		Online	23	

Table 2 shows the that we have 883 sample-size of students; distributed as follows: 557 students enrolled in the brick and mortar (traditional method), whereas 326 participated in the SCORM-based courses, in fact the students intend to enroll in the traditional course mode rather than the online one due to some reasons:

- Internet access and coverage.
- Hands-on examples and assignments.
- Culture of teaching.

this study considers the point-grading system as shown in Table 3.

Table 3 Grading Method

Description	Grade	Point	Description	Grade	Point
Outstanding	95 –100	4.5	**Satisfactory**	75 - 80	2.5
Excellent	90 - 95	4.0	**Fair**	70 - 75	2.0
Very good	85 - 90	3.5	**Adequate**	65 - 70	1.5
Good	80 - 85	3.0	**Weak**	60 - 65	1.0

The collected data has been analyzed by SPSS statistical software, t-test, Odds ratio (OR), Confidence level (CI) were computed to analyze the performance of the two groups based on courses and teaching methods, eventually, the t-test has been used to measure the effectiveness of both methods of teaching. Tables 5 shows that there is a significance level values of three courses out of five, as they are below the value of (0.05) which is the confidence interval. whereas, the remaining two courses have not shown a significant difference, statistically, CS-233 scores the lowest Mean value (i.e., 2.786) in the traditional method for a 42 enrolled students, while the highest scored Mean value obtained in the course of CS-328 (i.e., 3.855), furthermore, the table shows that the highest Mean difference is -0.063 while the lowest is -0.829, further data is also shown in this particular table (i.e., STD., STD. Error Mean, STD Error Difference, etc…), the computing of t-value is illustrated in Table 4.

Table 4. t-test Calculations

Treatment 1
N_1: 153
$df_1 = N - 1 = 153 - 1 = 152$
M_1: 2.99
SS_1: 93.29
$s^2_1 = SS_1/(N - 1) = 93.29/(153-1) = 0.61$
df means degrees of freedom.

SS means single sample.

Treatment 2
N_2: 92
$df_2 = N - 1 = 92 - 1 = 91$
M_2: 3.82
SS_2: 40.47
$s^2_2 = SS_2/(N - 1) = 40.47/(92-1) = 0.44$
M mean Mean
N Number of Samples

T-value
$s^2_p = ((df_1/(df_1 + df_2)) * s^2_1) + ((df_2/(df_2 + df_2)) * s^2_2) = ((152/243) * 0.61) + ((91/243) * 0.44) = 0.55$
$s^2_{M1} = s^2_p/N_1 = 0.55/153 = 0$
$s^2_{M2} = s^2_p/N_2 = 0.55/92 = 0.01$
$t = (M_1 - M_2)/\sqrt{(s^2_{M1} + s^2_{M2})} = -0.84/\sqrt{0.01} = -8.56$

The t-value is -8.56444. The p-value is $< .00001$. The result is significant at $p < .05$.

Table 5. Summary of the Research Results

Course	Method	Samples	Mean	Standard deviation	std. Error Mean	Mean difference	std. Error Difference	t-Test for Equality of Means
CS-108	Traditional	154	2.978[1]	**0.787**	0.0634	-0.829	0.098	S[2]
	Online	92	3.807	**0.661**	0.0691			
CS-131	Traditional	158	3.120	**0.756**	0.0601	-0.674	0.093	S
	Online	103	3.794	**0.695**	0.0686			
CS-233	Traditional	42	**2.786**	**0.970**	0.1497	-0.129	0.249	NS
	Online	21	2.914	**0.840**	0.1833			
CS-328	Traditional	149	3.449	**0.908**	0.0744	-0.406	-0.111	S
	Online	87	**3.855**	**0.651**	0.0698			
CS-329	Traditional	54	3.763	**0.791**	0.1076	-0.063	0.199	NS
	Online	23	3.826	**0.819**	0.1708			

NS: NOT significant at **95%** of Confidence Interval Percentage.

S: This test is statistically significant.

5. Discussion and Results Analysis

Further analysis has been conducted in Table 6 and Table 7 respectively, basically, we have analyzed each course separately based on the academic year, let us consider the first course of CS-108; we can clearly notice that the online-student performance is slightly better when comparing it to the other group, however, the result shows that there is a significant achievement of student performance throughout the years 2017-2020. Almost similar result is clearly obtained with three other courses (i.e., CS-131, CS-233 and CS-238), nevertheless, CS-238 course is return an insignificant level values. However, these findings prove that SCORM-based learning is slightly better than traditional learning.

5.1 Result Analysis

This section presented the obtained result, we have used the Mean difference overall percentages to support our findings as negative sign indicates that there is an improvement of the student scores when falling in online learning. Furthermore, the overall analysis of the standard deviation shown in table 6 and Table 7 also supports the same findings of the improvement of the online-student's performance.

Table 6. Comparison of the Results Between Courses by Academic Year vs. Delivery Method

Academic Year	Method	Samples #	Mean		Std. Error	Standard Deviation			Level
			Value	Diff.		Value	Error Diff	f-value	
Fundamentals of Programming Languages									
2017/2018	Traditional	54	2.98	-0.83	0.113	0.833	0.182	0.418	S
	Online	30	3.81		0.139	0.733			
2018/2019	Traditional	43	2.89	-0.80	0.112	0.737			S
	Online	31	3.69		0.125	0.70	0.169	0.494	
2019/2020	Traditional	57	3.04	-0.83	0.104	0.786			S
	Online	31	3.86		0.105	0.590	0.163	5.03	
Computer Programming (I) – C++									
2017/2018	Traditional	73	3.05	-0.74	0.094	0.801			S
	Online	45	3.80		0.093	0.627	0.140	4.91	
2018/2019	Traditional	48	3.09	-0.90	0.096	0.664			S
	Online	25	4.00		0.145	0.740	0.168	0.054	
2019/2020	Traditional	37	3.29	-0.34	0.126	0.768			NS
	Online	33	3.63		0.129	0.731	0.181	1.382	
Web/App Programming – PHP, ASP.NET									
2017/2018	Traditional	11	2.46	-1.12	0.380	1.317			NS
	Online	8	3.57		0.347	0.982	0.554	0.268	
2018/2019	Traditional	16	2.42	-0.84	0.170	0.703			NS
	Online	6	3.27		0.443	1.085	0.385	1.673	
2019/2020	Traditional	15	2.91	-0.86	0.258	0.967			S
	Online	7	3.77		0.220	0.582	0.399	2.07	
Computer Programming (II) – OOP using C#									
2017/2018	Traditional	64	3.13	-0.72	0.126	1.010			S
	Online	37	3.85		0.108	0.665	0.184	11.05	
2018/2019	Traditional	49	3.82	0.01	0.110	0.779			NS
	Online	25	3.81		0.170	0.816	0.199	0.245	
2019/2020	Traditional	36	3.57	-0.309	0.114	0.672			S
	Online	25	3.88		0.094	0.470	0.156	5.563	
Selected Programming Language (III) - Python									
2017/2018	Traditional	13	3.89	0.070	0.233	0.840			NS
	Online	9	3.82		0.348	1.046	0.402	0.544	
2018/2019	Traditional	20	3.76	0.004	0.181	0.809			NS
	Online	9	3.75		0.266	0.799	0.324	0.077	
2019/2020	Traditional	21	3.69	-0.274	0.168	0.771			NS
	Online	5	3.96		0.194	0.436	-0.361	0.995	

Table 7 exhibits a result of comparisons between different classes of the same course through 3-year period 2017-2020. Hence, the overall result proves again that the student's tendency toward the mean value of the SCORM-based learning is slightly better.

Table 7. A Comparison of the Results Between Delivery Methods by Course vs. Academic Year

Method	Offered Academic Year	Samples #	Mean Value	Mean Diff.	Std. Error	df	Level
Fundamentals of Programming Languages							
Traditional	2017/2018	54	2.98				
	2018/2019	43	2.89	0.09	0.16	95	**NS**
	2019/2020	57	3.04	-0.15	0.15	98	**NS**
Online	2017/2018	30	3.81				
	2018/2019	31	3.69	0.12	0.19	60	**NS**
	2019/2020	31	3.86	-0.17	0.16	60	**NS**
Computer Programming (I) – C++							
Traditional	2017/2018	73	3.05				
	2018/2019	48	3.09	-0.50	0.14	119	**NS**
	2019/2020	37	3.29	-0.19	0.16	83	**NS**
Online	2017/2018	45	3.80				
	2018/2019	25	4.00	-0.20	0.17	70	**NS**
	2019/2020	33	3.63	0.37	0.19	55	**NS**
Web/App Programming – PHP, ASP.NET							
Traditional	2017/2018	11	2.46				
	2018/2019	16	2.42	0.04	0.38	26	**NS**
	2019/2020	15	2.91	-0.49	0.30	29	**NS**
Online	2017/2018	8	3.57				
	2018/2019	6	3.27	0.30	0.55	12	**NS**
	2019/2020	7	3.77	-0.50	0.47	11	**NS**
Computer Programming (II) – OOP using C#							
Traditional	2017/2018	64	3.13				
	2018/2019	49	3.82	-0.69	0.17	112	**S**
	2019/2020	36	3.57	0.25	0.16	83	**NS**
Online	2017/2018	37	3.85				
	2018/2019	25	3.81	0.04	0.19	59	**NS**
	2019/2020	25	3.88	-1.17	0.19	46	**NS**
Selected Programming Language (III) - Python							
Traditional	2017/2018	13	3.89				
	2018/2019	20	3.76	0.13	0.29	31	**NS**
	2019/2020	21	3.69	0.07	0.25	39	**NS**
Online	2017/2018	9	3.82				
	2018/2019	9	3.75	0.07	0.44	16	**NS**
	2019/2020	5	3.96	-0.21	0.39	12	**NS**

6. Conclusion

This study is primarily designed to compare the student performance of programming-language courses who enrolled in SCORM-based courses versus those who enrolled in traditional learning method.

Computer programming source-codes and explanations are transferred into SCORM environment and e-learning supporting resources are developed, SCORM has been selected as a way to deliver operational and functional programming courses for many advantages: Accessibility, Adaptability, Durability, Reusability, and Interoperability

The student's final examinations scores are the key factor for our analysis, the study shows that the SCORM-based learning usually outperforms the traditional method students for some context.

We have to mention here that same course has been taught by different teachers during this period of time, the reason why this may negatively affect the result obtained by this study.

We could not draw any conclusions in regard to the correlation between performance fluctuation and methods of learning in this study due to insufficient evidences.

SCORM-based learning usually integrates three areas altogether (i.e., Content, Communications, and Management) to improve the delivery of programing learning in addition to provide flexible timetable for a computer programming subjects.

This study contributes in providing key information for further studies in similar domains, it helps researchers, decision-makers and educators utilizes the data and make a proper decision in regard to their teaching method.

References

Allen, I. E., & Jeff, S. (2013). *Changing course: Ten years of tracking online education in the United States*. Sloan Consortium. PO Box 1238, Newburyport, MA 01950, 2013.

Bakhouyi, A., Dehbi, R., Banane, M., & Talea, M. (2019, July). A semantic web solution for enhancing the interoperability of e-learning systems by using next generation of SCORM specifications. In *International Conference on Advanced Intelligent Systems for Sustainable Development* (pp. 56-67). Springer, Cham. https://doi.org/10.1007/978-3-030-36653-7_5

Nahar, K. M. O., Mohammed, M. A. S., & Izzat, A. (2016). Students performance between classical and online education: a comparative study. *International Journal of Continuing Engineering Education and Life Long Learning, 26*(4), 359-371.

Cedazo, R., Cecilia, E., Garcia, C., & Basil, M. A. (2015). A friendly online C compiler to improve programming skills based on student self-assessment. *Computer Applications in Engineering Education, 23*(6), 887-896. https://doi.org/10.1002/cae.21660

Distante, D., Villa, M., Sansone, N., & Faralli, S. (2020, July). MILA: A SCORM-Compliant Interactive Learning Analytics Tool for Moodle. In *2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)* (pp. 169-171). IEEE. https://doi.org/10.1109/ICALT49669.2020.00056

Ellis, R. A., Paul, G., & Leanne, P. (2009). E-learning in higher education: some key aspects and their relationship to approaches to study. *Higher Education Research & Development, 28*(3), 303-318. https://doi.org/10.1080/07294360902839909

Gholamhosseini, L. (2008). E-learning and its place in higher education system. *Paramedical Medicine magazine of IRI army force, 2*(2), 28-35.

Hentea, M., Mary, J. S., & Lisa, P. (2003). A perspective on fulfilling the expectations of distance education. *Proceedings of the 4th conference on Information technology curriculum*. 2003. https://doi.org/10.1145/947121.947158

Ho, C. L., & Ren-Jye, D. (2010). Construction safety training via e-Learning: Learning effectiveness and user satisfaction. *Computers & Education, 55*(2), 858-867. https://doi.org/10.1016/j.compedu.2010.03.017

Jama, M. P., Mabokang, L. E. M., & Adriana, A. B. (2008). Theoretical perspectives on factors affecting the academic performance of students. *South African Journal of Higher Education, 22*(5), 992-1005. https://doi.org/10.4314/sajhe.v22i5.42919

Lam, M. S. W. et al. (2008). Designing an automatic debugging assistant for improving the learning of computer programming. *International Conference on Hybrid Learning and Education*. Springer, Berlin, Heidelberg, 2008. https://doi.org/10.1007/978-3-540-85170-7_32

Mason, R., & Martin, W. (2001). Factors Affecting Students' Satisfaction on a Web Course. *Ed at a Distance, 15*(8), 8.

Copyrights