9-2018

Teaching basic programming to pre-university students through blended learning pedagogy: A descriptive study

Vandana RAMACHANDRA RAO
Singapore Management University, VANDANARR@SMU.EDU.SG

Ngee Mok HENG
Singapore Management University, nmheng@smu.edu.sg

Citation

RAMACHANDRA RAO, Vandana and HENG, Ngee Mok. Teaching basic programming to pre-university students through blended learning pedagogy: A descriptive study. (2018). *Transactions of ISATE: The 12th International Symposium on Advances in Technology Education, Nurturing Professionals for Smart Cities: Way Forward for Technology Education, 18-21 September 2018, Hong Kong*. 1-6.

Available at: https://ink.library.smu.edu.sg/sis_research/5948

TEACHING BASIC PROGRAMMING TO PRE-UNIVERSITY STUDENTS THROUGH BLENDED LEARNING PEDAGOGY– A DESCRIPTIVE STUDY

Ramachandra Rao Vandana [a] and Heng Ngee Mok [b]

School of Information Systems, Singapore Management University, Singapore

[a] vandanarr@smu.edu.sg
[b] mok@ieee.org

Abstract

Students enrolling for undergraduate programmes in Singapore would have either finished their polytechnic diploma or completed Junior College (JC) studies. Most pre-university students coming through the JC pathway are not exposed to programming as computing is offered as a subject in a very few JCs. The authors of this paper conducted four runs of an introductory programing course between 2016 and 2017 for a research project funded by the Ministry of Education, Singapore. The project named "Let's Code!" was intended to introduce fundamental programming concepts to students and guide them to consider taking a computer-science related degree for their university education. Pre-university students who had no background in programming could enrol in one of the runs of the "Let's Code!" programming course. Blended learning pedagogy was adopted to deliver the course content in three weeks. The purpose of this study is to gain insights into the delivery of an introductory programming course to a heterogeneous group of pre-university students through a blended learning pedagogy. This paper analyses the survey responses and the test scores of the participants who attended the course in the two runs of June and December 2017. Based on the test scores taken on the final day of the course, it was found that (i) male students performed better than the female students regardless of whether they had prior programming exposure, and (ii) students who had exposure to programming performed better than those with no prior background.

Keywords: *introductory programming course, blended learning pedagogy, university education choice, computer science studies, open educational resource, student outreach.*

Introduction

Students applying for undergraduate university admissions in Singapore primarily come from two streams. They are either (i) students with a polytechnic diploma or (ii) students who have completed their A-level at one of the Junior colleges (JC) (Singapore Education - Pre-University, n.d.), or students with an International Baccalaureate (IB) diploma (International Baccalaureate – Programmes, n.d.). The IB diploma is considered equivalent to the A-levels in the pre-university education landscape; Bhardwa (2017) compares the two (A-levels and IB) in her article in the Times Higher Education. In most of the JCs in Singapore, "Computer Studies" is not offered as a subject. The authors of this paper hence decided to offer an introductory programming course to all JC students. The authors got funding from the Singapore Ministry of Education's "MOE Academies Fund" for the project, named "Let's Code!". The project's primary objective was to expose JC students to programming by conducting four similar runs of an introductory programming course in a span of two years in order to help them consider an undergraduate degree related to computer science. The secondary objective was to create awareness about the computer science-related degree options available at the Singapore Management University's School of Information Systems (the school the authors work at). The programming course was open to all students from the JCs, the IB diploma program, and students in their fourth year from the Integrated Programme at the secondary level (Other Programmes in Secondary School - Integrated Programmes, n.d.). This meant that the participants of the course were between the tenth and twelfth year of their school education.

The programming course was designed to cater to participants with no computing background. The focus of the course was problem solving. Only fundamental programming concepts such as variables, conditional and loop structures, functions, and arrays were covered. The Ruby programming language was chosen as the language of instruction.

The four runs of the course were scheduled in June and December (of 2016 and 2017) to coincide with the school holidays in Singapore in order to facilitate the attendance of school-going participants. Each run lasted for three weeks. The course content for the four runs of the course was identical albeit minor differences in the delivery.

A major portion of the project funds was used to pay the teaching assistants who assisted the instructors (authors of this paper) in the delivery of the course. The teaching assistants were first or second year Information Systems' undergraduate students and were selected by the instructors. More than 500 pre-university students enrolled to attend one of the four runs.

The course content (video lectures, assignments) was uploaded onto the Singapore Management University's learning management system. The learning material was also made publicly available on a website created for this project (Let's code!, n.d.). The website that had the registration page was sent to all JCs and IB / IP schools in Singapore. The teachers in the schools helped to publicise the course before the beginning of every run. The participants of this course self-registered to one of the runs.

Initial data for this study comes from responses by the registrants to the survey questions. The purpose of this pre-course survey was to collect data regarding the profile of the registrants enrolling for the course (sex of the student, which school they were from, which year of study they belonged to, subjects taken at school etc.) as well as for the instructors to understand registrants' exposure and background in programming (Has the student experienced programming using Scratch, Alice, or other languages? What is the student's self-assessment of his experience in programming?). At the end of the course, participants provided course feedback. The post-course feedback survey followed a modified version of FACETS (Student feedback on teaching, n.d.), a survey instrument used at Singapore Management University for collecting teaching feedback. In addition to the above two sources (pre-course and post-course survey), data collected from the learning portal through the process of participants' course of study (online quizzes, attendance) and data from the results of the exam conducted at the end of the course (for every run) constituted the data for this study.

Although data for the four runs of "Let's Code!" programming course was available, this study analyses the data collected from the last two runs. This is because the last two runs adopted the same post-course feedback survey instrument (FACETS); the initial two runs followed a different instrument that was too long and discouraging for the students to fill. This study attempts to find answers for the following questions:

1. How effective is the delivery of the introductory programming course using blended learning pedagogy, related to students' learning and performance?
2. Do participants' prior programming experience affect their results?
3. Is there any other influencing factor that affected the participants' results?

Pedagogy

The participants who hailed from various schools (from 24 schools for the two runs in 2017) registered online for the programming course. All registrants were required to attend a briefing session during which the instructors briefed the participants about blended learning (BL) pedagogy adopted for this course. The instructors were well aware of the BL pedagogy (Mok, 2014). On the briefing day, the students were assigned groups; depending on the size of the group, up to two teaching assistants were assigned as mentors to each group. The students got help to set up their laptop with the programming environment on the briefing day. The briefing session also provided an opportunity for every student to know his mentors and peers.

During the briefing session, the students were given an online Multiple Choice Question (MCQ) quiz to test their programming exposure prior to taking the course as it was expected that a few students would have some programming knowledge. An almost identical test was given to them at the end of the course.

The topics covered in the course were categorized into four units and the students were advised to follow the learning material in sequence. Table 1 shows the topics covered in all the planned seven meet-up sessions of each run.

Table 1: Topics covered during the meet-up sessions

Meet-up session	*Topics covered*
1	Briefing and set up session
2	Variables, Types, Operators (Unit 1)
3	Decision and Loops (Unit 2)
4	Methods (Unit 3)
5	Arrays (including 2D arrays) (Unit 4)
6	Putting all concepts together
7	Debrief and exam session

Apart from the briefing session and the last meet-up session, that served as the exam and de-brief session, the participants were required to attend at least two of five other meet-up sessions (2 to 6) as a requirement to be eligible for a certificate of completion. During these meet-up sessions, tutorial type lessons were conducted. The first four tutorials had problems to be solved based on the new topic covered in the recent lecture videos of the corresponding unit, whereas the fifth tutorial had questions that helped students combine all the concepts taught in the previous units and also to prepare them for the exam.

Every unit required the students to watch online lecture videos, take the self-check quizzes and attempt assignment questions related to the unit. Figure 1 illustrates the learning plan recommended to the students for watching videos. The students were advised to attempt all self-check quizzes related to the unit before attempting the corresponding unit's assignment questions. Mentors provided support to students during this period of self-learning.

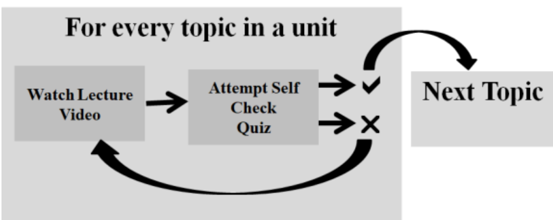

Figure 1: Recommended mode of learning

During the tutorial sessions, more questions pertinent to the topics covered in the recent unit were given to reinforce the concepts. Figure 2 shows the sequence of activities lined up for the student for every unit.

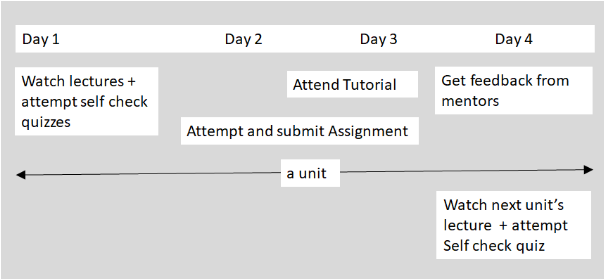

Figure 2: Sequence of activities per unit

During the tutorial session, instructors and mentors guided students on how to approach programming questions and compared various solutions to the same question. The meet-up sessions allowed the mentors to engage with the students and assist them with their assignments.

In the first three runs, the tutorial session served as a review session and was held on Day 3 as shown in Figure 2. However, for the last run, the tutorial was held just after the students watched the lecture video to help clear students' doubts on concepts (on Day 2) before they tackled the assignment questions. The change was implemented based on the feedback from the mentors, and our observation of students' inability to apply the concepts learnt to the assignment questions. The authors of this paper documented the experience of conducting the course, with all the implementation details, and the changes made across the four runs in Mok & Rao (2018).

Each of the first four tutorials corresponding to units 1 to 4 and the final tutorial that consolidated topics of all units had a corresponding assignment (summing to five assignments) that was required to be submitted. The mentors' responsibility was to provide feedback on the submitted assignments of students mentored by them.

The briefing session and the five tutorials were spread over three weeks with each week typically having two tutorials. The exam consisted of MCQs and programming exercise questions for students to solve. The students used their computers to write the solutions for the programming exercises. The MCQs were a mixture of online and paper based questions. As previously mentioned, the online MCQs were very similar to those given on the briefing day. In order to earn a "Certificate of Completion", the students had to meet attendance requirement (at least two out of five tutorials), submit five assignments, and attempt the exam. Students who fulfilled the requirements for the "Certificate of Completion" and passed the exam were given a "Certificate of Merit" in recognition of their good performance.

Methods and Materials

For this study, the sample data comes from 266 students who enrolled in the June and Dec. 2017 runs (out of a total of 535 participants across the four runs).

It was encouraging to observe that there was a good interest from female students to enrol in the course. The percentage of female students (47.4%) enrolled compared to male students (52.6%) did not show a huge difference indicating growing interest among female students to be in the STEM sector (More women working in science, engineering sectors, 2016). The interest could be attributed to Singapore's push to become a "smart nation" (Info-communications Media Development Authority, 2015) and the job opportunities available in the technology sector (Heng, 2017; Tegos, 2017). A review of the participants' responses to the pre-course survey question, "Describe your goals for participating in this course" asserted the above opinion. As examples, there were responses that read "I am interested in taking an IT related course in uni but I have come from a bio background since secondary 3 till JC, so I hope this course can firm my interest on IT and guide me into this path I hope to take.", "wish to learn and understand more about coding as it has always been emphasised as an essential skill in this 21st century…", "With the knowledge that Singapore is striving towards being a smart nation and that technology will definitely be an integral part of society in the future, I hope that through this course I would be able to grasp the basics and foundations of programming...".

Table 2 reports the demographic profile of the students in the sample. Most (85%) of the 266 students were either in their 11th or 12th year of education compared to 9% in the 10th year of education.

Table 2: Demographic profile of students from the June and Dec. 2017 runs

	Percentage	n
Gender		
Female	47.4%	126
Male	52.6%	140
Year of study		
JC1 or IB Year 5	33.5%	89
JC2 or IB Year 6	51.5%	137
IP year 4 or IB Year 4	9%	24
Graduated from JC / IB	6%	16
Prior Programming Exposure		
With Programming Experience	22.2%	59
No programming Experience	76.7%	204

Although this course was catered to students with no programming experience, we had questions in the survey related to the extent of exposure the students had to programming prior to joining our course. We used a binary measure to capture participants' prior programming exposure. Based on the students' responses, we considered those who had exposure to at least one programming language such as C, C++, Python, Java, JavaScript and other non-visual programming languages as having had “programming experience” whereas those who mentioned that they have tried HTML, or Alice, or Scratch with no other typical programming language experience to be having “No programming experience”. Based on this criterion, the percentage of students with programming experience was 22.2%, compared to 76.7% with no prior programming experience as shown in Table 2. Only 11.9% of the female participants had programming experience compared to 31.4% of the male participants.

Although the participants were required to attend only two out of five tutorials, we observed that a substantial number of students (60.9%, n = 266) attended at least four tutorials.

Among the 266 students who enrolled for the course in 2017, 215 (80.8%) completed the course and were eligible for the “Certificate of Completion”. The attrition rate of 19.2% was considered low, as the students had no obligation to complete the course. Table 3 captures the course’ completion rate statistics by gender in the two runs.

Table 3: Course completion rate statistics

	Percentage	n
Earned “Certificate of Completion”		
Male	77.9%	109
Female	84.1%	106
Total	80.8%	215
Earned “Certificate of Merit”		
Male	53.6%	75
Female	25.4%	32

Among the 266 students, 40.2% (n=107) were eligible for the “Certificate of Merit” which meant that they completed all the requirements for the “Certificate of Completion” and also passed the exam.

Results and Discussion

In order to measure the learning achievement of students, we compared the results of MCQs attempted by the students on the briefing day (beginning of the course) and compared it with their scores (out of a maximum score of 8) on the exam day when they attempted a similar set of MCQs (as on the briefing day). Data from the participants who had not attended one of the tests, pre-course or the post-course MCQ test was not considered. The mean difference between the scores before the run of the course (1.53/8 i.e. 19.1 %) and after the course (6.1/8 i.e. 76.3 %) was noted to be statistically significant with a p value less than 0.01 at 95% confidence level (n = 228).

In order to examine the impact of prior programming exposure of the students, we conducted the t-test again to compare the MCQ scores of participants who had no prior programming experience. Table 4 tabulates the result of paired two-sample t-test performed on the unfiltered and the filtered sample. The significant difference in the mean scores (pre-course 16.5 %, post-course 74.6 %, n = 174, p value < 0.01) for students with no prior programming background indicated that there was a positive learning achievement. The mean difference for those with prior programming experience (p value < 0.01, n=54) also seemed to be statistically significant indicating that even the group with prior programming exposure improved in their competence.

Table 4: Comparison of paired two sample t-test results

	n	Pre-course MCQ mean (in %)	Post-course MCQ mean (in %)
All participants	228	19.1	76.3
Participants with no prior programming experience	174	16.5	74.6
Participants with prior programming experience	54	27.8	81.0

The post-course online MCQ test constituted to about 1/6th of the total exam score. We then studied the data by comparing the total exam score of all students with no prior programming experience and those with prior programming exposure. The mean score for the sample data with “no programming experience” was 46.5%, “with programming experience” was 61.0%, and the mean difference was significant at p < 0.01.

We also compared the results of the total exam score by gender. It was observed that male students performed better (mean 58.1%) than the female students (mean 41.4%) and the t-test showed that the difference is significant (p < 0.01) at 95% confidence intervals. Table 5 shows the comparison of the mean exam scores for male and female participants by their prior programming experience.

Table 5: Comparison of exam mean scores for male and female participants

	Average Exam Score	
	Males	Females
Regardless of prior programming experience	58.1 %	41.4 %
No prior programming experience	54.7 %	40.0 %
With prior programming experience	64.8 %	50.9 %

Running the t-test taking only females and males with no prior programming exposure also showed better performance of male students (p < 0.05). In comparison, running t-test taking sample data of females and males with prior programming experience, the mean difference was observed to be 50.9% for females and 64.8% for males (p < 0.02).

We had attendance records of the students for every tutorial and explored the data to find if there exists a correlation between the number of tutorials attended by a student and his exam score. Table 6 illustrates the average exam score by students in relation to the number of tutorials attended. The correlation coefficient showed positive relationship but the strength was not very high (correlation coefficient = 0.54).

Table 6: Mean Exam Score Vs Attendance at Tutorials

Number of tutorials attended	*Percentage of students*	*Mean exam score of students*
1	0.4 %	41.25
2	4.7 %	51.86
3	16.7 %	47.87
4	28.2%	44.79
5	50 %	53.37

The students provided data related to the average number of hours spent on the course per day during the three weeks. Putting the figures through multiple linear regression test showed that the variables - Number of tutorials attended, Sex of the student, Pre-course MCQ test score, Prior Programming Experience and Number of hours spent by the student on an average per day, determined about 34% of the variability in the exam scores. Figure 3 shows the multiple regression test result performed using Excel.

SUMMARY OUTPUT		
		Dependant Variable: Exam Score
Regression Statistics		Independent Variables:
Multiple R	0.587012	Number of Tutorials attended
R Square	0.344584	Sex of the student
Adjusted F	0.33021	Prior Programming Experience
Standard E	17.50979	Pre-Course MCQ Score
Observatic	234	Average number of hours spent per day

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regressior	5	36751.39	7350.277	23.97409	2.33597E-19
Residual	228	69903.11	306.5926		
Total	233	106654.5			

Figure 3: Multiple Linear Regression Analysis Result

Our research study was more exploratory in nature. Further study would be required to understand the differences in learning achievement between males and females. Perhaps studies that capture students' attitudes towards computer studies, learners' cognitive disposition, confidence level, motivation level and learners' persistence would help to understand the differences in the results.

Conclusions

This study showed that there was a significant improvement in student's learning based on the pre-course and post-course MCQ test. This suggests the effectiveness of using BL approach in teaching introductory programming to pre-university students. There were two findings that emerged from this study; male students performed better than the female students regardless of the prior programming exposure they had, and students who had exposure to programming performed better than those with no prior background. This could perhaps be because of the steep learning curve for those with zero background in programming. Further research is required to determine the cause of gender differences in order to address the gap.

Acknowledgements

The authors would like to thank their colleagues Ms. Fiona Lee and Mr. Timothy Chan who helped review the paper, Associate Professor Manoj Thulasidas who helped verify the statistical results, General Office staff who provided administration support, the Ministry of Education (Singapore), Hwa Chong Institution as well as the following organizations: Science Centre Singapore, IEEE (Singapore branch), Singapore Computer Society, and the Info-communications Media Development Authority, that supported this initiative.

References

Bhardwa, S. (2017, July 11). The International Baccalaureate versus A levels. *Times Higher Education*. Retrieved from https://www.timeshighereducation.com/student/news/international-baccalaureate-versus-a-levels

Heng, L. (2017, January 20). Singapore IT sector set for job growth. *The New Paper*. Retrieved from https://www.tnp.sg/news/singapore/singapore-it-sector-set-job-growth-2017

Info-communications Media Development Authority. (2015, June 2) *Singapore is positioning itself for the next wave of opportunities arising from the convergence of the infocomm and media sectors*. [Media Release]. Retrieved from https://www.imda.gov.sg/about/newsroom/archived/ida/media-releases/2015/new-ideas-and-initiatives-to-lead-singapore-towards-smart-nation

International Baccalaureate – Programmes (n.d). Retrieved from https://www.ibo.org/programmes/

Let's Code! (n.d.). Retrieved July 18, 2018 from code.smu.edu.sg

Mok, H. N. (2014). Teaching tip: The flipped classroom, *Journal of Information Systems Education, 25*(1), 7-11, Spring 2014

Mok, H.N. & Rao. V. (2018). Introducing basic programming to pre-university students: A successful initiative in Singapore. *17th International Conference on Information Technology Based Higher Education and Training (ITHET)*, Portugal, 26-28th Apr 2018. IEEE

More women working in science, engineering sectors. (2016, March 25). *The Straits Times*. Retrieved from https://www.straitstimes.com/singapore/more-women-working-in-science-engineering-sectors

Other Programmes in Singapore – Integrated Programmes (n.d.). Retrieved from https://www.moe.gov.sg/education/secondary/other/integrated-programme

Singapore Education: Pre-University (n.d.). Retrieved from https://www.moe.gov.sg/education/pre-university

Student Feedback on Teaching (n.d.). Retrieved from https://cte.smu.edu.sg/feedback-teaching/student-feedback-teaching

Tegos, M. (2017, May 25). Singapore plans to spend $1.7b this year in contracts for Smart Nation apps. *Tech In Asia*. Retrieved from https://www.techinasia.com/govtech-smart-nation-update-2017