

# Does Education Culture Influence Factors in Learning Programming: A Comparative Study between Two Universities across Continents

**Ms. Ritu Sharma**  
Flinders University  
Australia

**Dr. Haifeng Shen**  
Flinders University  
Australia

**Abstract.** Learning programming has been considered challenging by students and remains a daunting task for educators despite their efforts into finding innovative ways to teach programming. A lot of factors have been studied since 1970s and educators today are still trying to unearth the factors that play significant roles in learning programming. As most of the important research in this regard has been done in developed countries, it is imperative to understand whether differences in education culture between developed and developing country can influence how the factors affect learning of programming. This paper presents the results of a comparative study on the effects of differences in education culture between two universities in Australia and India respectively on the chosen factors in learning introductory programming. The results show that prior programming experience, gender, reason to study programming, attendance, and revision had different effects, while activities performed in the lecture theatre and preliminary preparation before lecture and lab had the same effect in the two universities. The findings help gain insight whether certain factors are culture dependent/independent so that educators can focus on the specific factors that will help students better learn programming in a particular education culture context.

**Keywords:** Programming; Factors; Gender; Attendance; Preparation before lecture; Revision.

## 1. Introduction

Learning programming is important for computer science as well as engineering students." In computer science, an expected outcome of a student's education is programming skill (McCracken et al., 2001)." Learning programming plays an

important role to be well-versed with technology (Lau & Yuen, 2011). It's not uncommon to hear students talking about the difficulty in learning programming and many of them discussing that they will never ever choose to study programming again. Many discussions in computer science are based on Teaching introductory programming at the tertiary level (Moström, 2011). (Caspersen & Kolling, 2009) suggested that it's a daunting task to teach programming to beginners. An entire volume of papers, called 'Studying the Novice Programmer', also documented the problems of learning to program (Collins, Brown, & Holum, 1991). At most universities academics believe that their first semester programming class is not as per the expectation. This belief leads to the reluctance to talk about their first semester programming classes outside their own institution (Lister, 2005).

Studies also confirm that this problem is not only prevalent in a particular country but many countries where programming is taught face the same problem irrespective of demographics, race and ethnicity. (McCracken, et al., 2001) conducted a multi-national, multi-institutional study of assessment of programming skills of first-year Computer Science students. In analysing the data from universities in different countries, they found that the problems they observed with programming skills seemed to be independent of country and educational system and the most obvious similarity observed was that the most difficult part for students seemed to be abstracting the problem to be solved from the exercise description (McCracken, et al., 2001).

A lot of factors have been studied since 1970s, such as prior programming experience(de Raadt et al., 2005; Wilson, 2002), gender(Byrne & Lyons, 2001; Lau & Yuen, 2009), Cognitive, behavioural, and attitudinal factors(Sheard, Simon, Hamilton, & Lönnberg, 2009; Simon et al., 2006). Learning approaches like deep learning approach and surface learning approach(Fincher et al., 2006; Simon, et al., 2006).

Educators today are still trying to unearth the factors that play significant roles in learning programming. However, as most of these factors have been studied in developed countries, it is imperative to understand whether education cultural differences between developed and developing country can influence how the factors affect learning of programming.

In this paper, we report the results of a comparative study between two universities representing west and east education cultures, one in Australia and one in India respectively in terms of effects of education cultures on the chosen factors in learning introductory programming. Education culture in this study refers to the teaching methodology used to teach programming, the assessment criteria, the attendance criteria and the exam structure. The factors taken into account are prior programming experience, gender, family background, preliminary preparation and revision, family background and study choices. A comparison between university students who have different education cultures motivated us to choose Australia and India as the countries to conduct this study. It is important to study if the factors affecting learning programming are

similar or different in these two universities as learning programming is considered a difficult task, but most of the research conducted has focused on a single education culture. The results show that prior programming experience, gender, reason to study programming, attendance, and revision had different effects, while activities performed in the lecture theatre and preliminary preparation before lecture and lab had the same effect in the two universities. These findings help gain insight whether certain factors are education culture dependent/independent so that educators can focus on the specific factors that will help students better learn programming in a particular education culture context.

The rest of the paper is organised as follows. Next section introduces some factors proposed by researchers in the area. After that, we describe the research methodology. The following section presents the results and discussions on the findings. The final section concludes the paper with a summary of major contributions and future work.

## 2. Related Works

A varied number of factors were analysed in the past.(Wilson & Shrock, 2001) investigated twelve possible predictive factors including math background, attribution for success/failure (luck, effort, difficulty of task, and ability), domain specific self-efficacy, encouragement, comfort level in the course, work style preference, previous programming experience, previous non-programming computer experience, and gender the spatial reasoning and mathematical ability. The study revealed three predictive factors in the following order of importance: comfort level, math, and attribution to luck for success/failure. Comfort level and math background were found to have a positive influence on success, whereas attribution to luck had a negative influence.

(Roddan, 2002) conducted research and concluded that the best indicators of success appeared to be self-predicted success, attitude, keenness and general academic motivation. The results suggested that students who carry on being successful programmers tend to have pre-existing strengths in a strategic/algorithmic style of articulation. (de Raadt, et al., 2005) studied the approaches to learning in computer programming students and their effect on success and concluded that in computing, like in other disciplines, learning approaches were a powerful determinant of success. Another study conducted by (Bergin & Reilly, 2005) found that a student's perception of their understanding of the module had the strongest correlation with programming performance.

A summary of the factors analysed in the past are as follows.

- Previous computing experience (Wilson, 2002);(Bergin & Reilly, 2005)
- Previous programming experience(Wilson, 2002);(de Raadt, et al., 2005; Hagan & Markham, 2000)
- Previous non-programming computer experience (Wilson, 2002)
- Attribution (Wilson, 2002)
- Self-efficacy (Wilson, 2002)

- Comfort (Wilson, 2002)
- Encouragement from others (Wilson, 2002)
- Work style preference (Wilson, 2002)
- Math background (Wilson, 2002)
- Midterm grade (Wilson, 2002)
- Learning approaches like deep learning approach and surface learning approach (Fincher, et al., 2006); (Robins, Rountree, & Rountree, 2003)
- Learning style (Byrne & Lyons, 2001)
- Standard paper-folding test (de Raadt, et al., 2005; Fincher et al., 2005; Fincher, et al., 2006)
- a cognitive task focusing on spatial visualisation and reasoning (Fincher, et al., 2006)
- map sketching (Fincher, et al., 2005; Fincher, et al., 2006); (Tolhurst et al., 2006)
- a behavioural task used to assess the ability to design and sketch a simple map and to articulate decisions based on that map (Fincher, et al., 2006)
- searching a phone book (Fincher, et al., 2006)
- a behavioural task used to assess the ability to articulate a search strategy (Fincher, et al., 2006)
- a standard study process questionnaire (Fincher, et al., 2006)
- an attitudinal task focusing on approaches to learning and studying (Fincher, et al., 2006)
- Bigg's instrument. (de Raadt, et al., 2005)
- Shortened Intellectual Development (ID) predictor (Barker & Unger, 1983)
- Cognitive (Bergin & Reilly, 2005; Sheard, et al., 2009)
- Behavioural (Sheard, et al., 2009) attitudinal factors (Robins, et al., 2003; Sheard, et al., 2009)
- Reading and Tracing Skills in Novice Programmers (Lister et al., 2004)
- The ability to articulate strategy (Cutts et al., 2006)
- Measures of general intellectual ability and thinking skills (Mayer, Dyck, & Vilberg 1986)
- Self-predicted success (Robins, et al., 2003)
- Keeness and general academic motivation (Robins, et al., 2003)
- Previous academic experience (Bergin & Reilly, 2005; Byrne & Lyons, 2001)
- Personal information (Bergin & Reilly, 2005)
- Experience on the module (Bergin & Reilly, 2005)
- Gender (Byrne & Lyons, 2001; Lau & Yuen, 2011)
- Mental models (Lau & Yuen, 2011)
- Prior composite academic ability (Lau & Yuen, 2011)
- Medium of instruction (Lau & Yuen, 2011)
- perception of the students (ÖZYURT & ÖZYURT, 2015)

These factors were studied to facilitate the process of learning programming as learning programming has been a topic of discussion among the Computer Science educators since 1970s. Teaching programming to novices was considered a big challenge and – according to the number of people attending conferences on programming – it still seems to be (Dijkstra, 1970); (Gries, 1974); (McCracken,

et al., 2001); (Robins, et al., 2003); (Collins, et al., 1991); (Mayer, et al., 1986) as cited by (Bartlett & Burt, 1933). The study of literature suggests that the challenge still remains.

This study is aimed to investigate the influence of cultural differences on the effects of these factors. If a factor is found to positively affect learning programming in a particular cultural context, it should then be incorporated into the teaching and learning methodology. For example, if revision is found to be a positive indicator of marks in an education culture, the teaching methodology should be designed in such a manner that the students in a university of that education culture are required to regularly revise the course material. If there are some common factors that have positive effect on learning programming despite the difference in education cultures, then they may be focused upon in the teaching methodology of programming.

### **3. Methodology**

A comparative study was conducted between a University in Australia and a University in India to study the influence of education culture differences devised on the factors chosen on the basis of Tinto's conceptual model (Tinto, 1975), which analyses the effect of various categories of factors on dropout decisions. These two universities were chosen as both of them are reputed universities and students of similar academic ability are enrolled in these universities. The chosen factors included prior programming experience, gender, reasons to study programming, attendance in lectures, activity in lecture theatre, effect of preliminary preparation before lectures, effect of preliminary preparation before laboratory, and effect of revision. It is worth mentioning that the factor of social integration, which plays an important role in university persistence, was not included from Tinto's model as it is out of the scope of this paper. Exam scores were used as a measure of student performance and understanding of the programming topic concerned.

Questionnaire responses regarding various aspects of programming skills were recorded for students in both universities where student participation was completely voluntary. Students at both universities had to complete the same questionnaire with 28 questions. The questionnaire was designed in a Likert scale format. The questionnaire included questions from various areas. The areas included language studied before studying the undertaken topic, gender, reason to study programming, study choices which included the number of programming lectures attended and activity in lecture theatre, kind of preliminary preparation (before laboratory and lectures) and revision (during semester or during exams). The results and discussion section briefly explains the contents of the questionnaire as they are analysed and discussed. In a significant study conducted at Monash University, Australia by (Butler & Morgan, 2007), self-completed questionnaire was used as a tool to collect data from the students enrolled in the programming topic. Another substantial study conducted by (Fincher, et al., 2006; Simon, et al., 2006) also used a questionnaire as one of the tools to collect data from the students. So questionnaire was found to be the most suitable tool to collect data in this study.

The purpose of this study was to determine the influence of differences in education culture on effects of factors that were significantly related to exam scores for each university.

The data was collected across three semesters at both universities where the topic was delivered on campus. At the Australian University, the assessment components included quiz, laboratory assignments or projects and a practical exam or theory exam at the end, while at the Indian University, the assessment components included mid-semester exams, practical exam, quiz and viva voce(oral exam) exam scores for the students were recorded. The purpose was to determine factors that were statistically significantly related to exam scores for each university.

Frequency tables were used to summarize the questionnaire responses. Descriptive statistics were used to summarize the exam scores. Skewness, kurtosis, Shapiro-Wilk normality test, and QQ plots were used to determine if the exam scores were normally distributed. To determine if there was a relationship between exam score and variables of interest, Wilcoxon ranked-sum tests (used when the variable of interest had 2 levels) and Kruskal-Wallis tests (used when the variable of interest had more than 2 levels) were conducted. Wilcoxon ranked-sum tests determine if there was a statistically significant difference in exam score between the two groups of interest. Kruskal-Wallis tests determine if there was a statistically significant difference in exam score among the groups of interest. If the results of Kruskal-Wallis test were significant, Dunn's procedure for pairwise comparisons was performed to investigate which two levels of the variable were statistically significantly different in the exam score. A p-value less than 0.05 indicated significance.

### 3.1 Sample description for Australian University

Data of 198 students were collected for Australian University, of which 4 students with no exam scores were excluded. The total questionnaire items were 28. Students who had answered over 90% of the questionnaire items (i.e., with 9 or less missing responses) were kept in the data analysis for this study. The final sample size for Australian University was 184.

**Table 1: Frequency counts of missing responses (Note: N = 194)**

Number of missing responses	Frequency (%)
0	178 (91.8)
1	5 (2.6)
9	1 (0.5)
11	1 (0.5)
28	1 (0.5)
63	2 (1.0)
73	1 (0.5)
79	2 (1.0)
89	1 (0.5)
99	2 (1.0)

### 3.2 Sample description for Indian University

Data of 94 students were provided for Indian University. The total questionnaire items were 28. Students who had answered over 90% of the questionnaire items (i.e., with 6 or less missing responses) were kept in the data analysis for this study. The final sample size for Indian University was 79.

**Table 2: Frequency counts of missing responses (Note: N = 94)**

Number of missing responses	Frequency (%)
0	34 (36.2)
1	20 (21.3)
2	7 (7.4)
3	7 (7.4)
4	5 (5.3)
5	5 (5.3)
6	1 (1.1)
13	3 (3.2)
16	1 (1.1)
22	1 (1.1)
54	1 (1.1)
80	1 (1.1)
94	3 (3.2)
98	1 (1.1)
113	2 (2.1)
114	2 (2.1)

### 3.3 Distribution of Data

For Australian University students, the negative skewness (-0.81) suggested that more data points lay to the right of the mean and the positive kurtosis (0.18) suggested that the distribution was taller (more peaked) than the normal distribution.

For Indian University the negative skewness (-0.23) suggested that more data points lay to the right of the mean and the negative kurtosis (-0.92) suggested that the distribution was flatter than the normal distribution.

For both Universities, as the data were not normally distributed, Wilcoxon ranked-sum tests and Kruskal-Wallis tests were used to determine which factors were statistically significantly related

### 3.4 A conceptual model based on Tinto's conceptual model

The performance of students in the exam may be a deciding factor whether the students want to take up programming later in their career. This suggests that persistence may be achieved based on some factors. To achieve persistence, a conceptual model based on Tinto's conceptual model (Tinto, 1975) was devised and shown in Fig 1. The factors suggested in the model below were studied for their effect on marks that were used to be an indicator of persistence. Factors which formed the basis of Tinto's model were chosen carefully based on the factors previously analysed.

Prior programming experience has been studied in the past but the results obtained varied and studied in a different context. According to a study conducted by (Hagan & Markham, 2000) students who had experience in at least one programming language at the beginning of an introductory programming course performed significantly better in the assessment than those with none. Another study conducted by (de Raadt, et al., 2005) concluded from their study that “while previous programming experience contributing to better marks is logical this was not found to happen in all cases and somewhat surprisingly not for all programming languages. Thus it was explored in detail by exploring the languages studied in the past.

Gender was chosen as a factor to be studied as the number of females enrolled in the programming topic is found to be generally very low in Australia and comparable in India. A study conducted by (Byrne & Lyons, 2001) concluded that female students achieved equally high scores as their male counterparts. It was important to find out if the gender has any effect on student performance.

*Family background* hasn't been explored in this context in the past but is an important factor stated in Tinto's conceptual model.

*Study choices*: This factor has been explored in this study as it hasn't been studied in this context in the past.

*Preliminary preparation and revision*: Both preliminary preparation and revision haven't been explored in the previous studies but a few studies related to other topics suggested that preliminary preparation and revision play an important role in learning. Thus it was analysed in the context of learning programming.

*Interest in programming*: As Information Technology is a vast growing field the numbers of jobs in this area are increasing and the scope is widening. So the Universities have introduced learning programming topic in almost all areas of engineering studies. It was included as part of Tinto's 'individual attributes' section to analyse if the reason to choose to study programming has any impact on student performance.

*Social and Academic Integration*: This factor was explored in a greater detail as social media has become an important part of young generation. It is important to mention that it has not been included in this paper as it is out of scope of this paper.



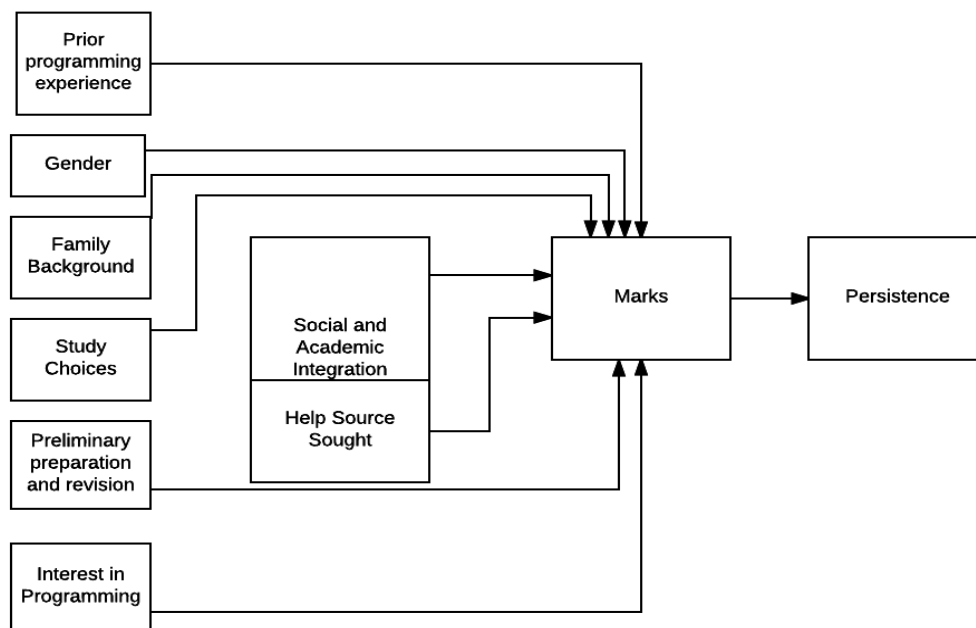


Fig 1: A conceptual model based on Tinto's original model

## 4. Results And Discussion

### 4.1 Prior Programming experience

Table 3 shows the mean exam scores by experience with programming. For Australian University, the results of the Wilcoxon ranked-sum tests suggest that students who studied programming at home before attending the course have statistically significantly higher exam scores than students who studied programming at other levels ( $p = 0.012$ ). This suggests that the students who study programming at home out of self-interest before attending the University perform better. This result is consistent with the results obtained by (Hagan & Markham, 2000) who analysed the effect of prior programming experience and the number of programming languages learnt and concluded that students who had experience in at least one programming language at the beginning of an introductory programming course performed significantly better in the assessment than those with none.

However, for the Indian University, there is no statistically significant difference in exam scores across other categories of experience with programming.

**Table 3: Mean (SD) exam scores of both Universities by prior experience with programming. Note: SD = standard deviation. p = p-value of the Wilcoxon ranked-sum tests. \* indicates significance at the 0.05 level.**

	Australian University			Indian University		
	Yes	No	p	Yes	No	p
9 <sup>th</sup> or 10 <sup>th</sup> grade	61.17 (29.48)	63.69 (21.12)	0.957	59.96 (17.10)	58.63 (14.38)	0.667
11 <sup>th</sup> and 12 <sup>th</sup> grade	60.89 (27.05)	63.98 (21.00)	0.926	61.89 (14.12)	57.93 (14.87)	0.332

At home/self-study	68.10 (26.37)	61.64 (20.47)	0.012*	62.39 (15.15)	57.29 (14.38)	0.136
At Univ.as part of a different degree	55.31 (30.44)	64.13 (21.33)	0.304	57.12 (14.12)	59.17 (14.90)	0.611

## 4.2 Gender

Table 4 summarises the “Mean (SD) exam scores by gender” for both universities. From the analysis of data of Australian University, the results of the Wilcoxon ranked-sum test suggest that there is no statistically significant difference in exam scores between male and female students ( $p = 0.828$ ). This result is congruent with the results achieved by (Lishinski, Yadav, Good, & Enbody, 2016) which suggests that female students performed as well as male students in the programming course by all indicators analysed in the study. However, from the data analysis of Indian University, the results of Wilcoxon ranked-sum tests suggest that females have statistically significantly higher exam scores than those of males ( $M = 65.29$ ,  $SD = 12.30$  for female;  $M = 55.29$ ,  $SD = 14.82$  for male;  $p = 0.003$ ).

**Table 4: Mean (SD) exam scores by gender. Note: SD = standard deviation. p = p-value of the Wilcoxon ranked-sum tests for gender**

Gender				
	Australian University		Indian University	
	Mean (SD)	p	Mean (SD)	p
Male	63.17 (22.62)	0.828	55.29 (14.82)	0.003*
Female	64.44 (20.71)		65.29 (12.30)	

This result is consistent with the dominant pattern of male vs female performance in academics in general in India and other countries as well. A study conducted by (Montmarquette, Mahseredjian, & Houle, 2001), concluded that “In those more competitive academic sectors with entrance quotas, being female (Gender) increases the probability of persisting at the university”. Another study conducted by (Arulampalam, Naylor, & Smith, 2004) concluded that “There are significant differences by gender, with males more likely to drop out”. For the past few years, the females have been performing better than males academically in India. The Civil services exam topper in India for year 2016 was a female. The class 12 results witnessed a similar trend with girls scoring better than boys as quoted by a newspaper “Girls have performed better than boys this time as well, with 88.58 per cent girls clearing the exams compared to 78.85 per cent of boys” (Sharma, 2016). Another similar study conducted by (Wilson, 2002) suggested that females reported having more encouragement to study computer science than the males in the sample. This result suggests that education culture may affect the performance based on gender, with females tending to perform better than boys in an eastern education culture. The cause of this variation needs further investigation.

### 4.3 Interest in programming

Table 5 shows the mean exam scores by reasons of studying programming for Australian University students. The various parameters evaluated include “interested to know about programming”, “It is upcoming in the work market”, “High paying work in the industry”, and “Mandatory in the degree”.

The results of the Kruskal-Wallis tests suggest that there is a statistically significant difference in exam scores among students viewing the importance of “Interested to know about programming” differently ( $p = 0.006$ ). The results of pairwise comparisons suggest that students who regard “Interested to know about programming” as not important have statistically significantly lower exam scores ( $M = 55.91, SD = 20.86$ ) than those regard it as most important ( $M = 66.20, SD = 25.76$ ) ( $p = 0.005$ ).

Table 6 shows the mean exam scores by reasons of studying programming for Indian University students. The analysis results of the Kruskal-Wallis tests suggest that there is no statistically significant difference in exam scores across categories of reasons of studying programming ( $p > 0.05$ ).

For the Australian University, the results show that the students who study programming because they are interested to know about programming score better, suggesting that students generally score better if they study programming by choice and not because of any forceful factor. This result is in close conjunction to the results of the study conducted by (Carter, 2006) in “Why students with an apparent aptitude for computer science do not choose to major in computer science” and concluded that the students’ understanding of the amount of money to be made in the field was not a significant influence in the choice not to study computer science. It is also similar to the results obtained by (Carter, 2006), as the students who chose to study programming as they were “interested to know about programming” scored better average score and the reason they chose to study programming was not “High paying work in the industry.” However, the different result obtained at the Indian University suggests that there may be some cultural factors behind it, which needs to be explored further.

**Table 5: Mean (SD) exam scores by reasons of studying programming for Australian University. Note: 1 = not important, 2 = somewhat important, 3 = most important. SD = standard deviation. p = p-value of the Kruskal-Wallis tests. \* indicates significance at the 0.05 level for both tables**

Why did you choose to study programming?				
	1	2	3	P
Interested to know about programming	55.91 (20.86)	65.39 (18.91)	66.20 (25.76)	0.006*
It is upcoming in the work market	60.95 (21.68)	68.78 (20.08)	59.51 (26.12)	0.055
High paying work in the industry	63.30 (21.68)	63.31 (22.01)	63.83 (27.74)	0.976
Mandatory in the degree	66.97 (24.17)	59.59 (23.01)	63.50 (21.62)	0.273

**Table 6: Mean (SD) exam scores by reasons of studying programming for Indian University. Note: 1 = not important, 2 = somewhat important, 3 = most important. SD = standard deviation. p = p-value of the Kruskal-Wallis tests. \* indicates significance at the 0.05 level for both tables**

Why did you choose to study programming?				
	1	2	3	P
Interested to know about programming	54.90 (12.10)	55.64 (16.24)	61.88 (13.29)	0.182
It is upcoming in the work market	62.09 (15.59)	60.34 (15.02)	56.39 (14.17)	0.367
High paying work in the industry	59.13 (16.75)	57.58 (15.20)	59.87 (14.17)	0.831
Mandatory in the degree	62.09 (13.35)	55.69 (16.66)	59.68 (13.66)	0.360

#### 4.4 Study Choices

Table 7 summarises Mean (SD) exam scores by number of programming lectures attended for Australian University. The results of the Wilcoxon ranked-sum tests suggest that there is a statistically significant difference in exam scores among students attending different amount of lectures. In particular, according to the results of pairwise comparisons, students who attended 100% of the lectures have statistically significantly higher exam scores ( $M = 71.80$ ,  $SD = 22.55$ ,  $p = 0.002$ ) than those who attended no more than 80% of the lectures ( $M = 57.77$ ,  $SD = 22.53$ ,  $p = 0.002$ ).

Table 8 summarises Mean (SD) exam scores by number of programming lectures attended for Indian University. The analysis results of Wilcoxon ranked-sum tests suggest that there is no statistically significant difference in exam scores across categories of attendance of programming lectures/lab ( $p > 0.05$ ).

The result obtained at Australian University is consistent with the results of the study conducted by (Massingham & Herrington, 2006), where they concluded that "At the same time it is clear that attendance has an impact on performance. Students who attended lectures and tutorials had a better chance of success on all assessment tasks in particular the final exam". Another study conducted by (Credé, Roch, & Kieszczynka, 2010) also concluded that attendance correlated strongly with both performance in an individual class and a college GPA. Also, the attendance-grade relationship was slightly stronger for science classes than for non-science classes. A study conducted by (Marburger, 2006) found that students who missed class on a given date were significantly more likely to respond incorrectly to questions relating to material covered on that day than those who were present. Another study conducted by (Stanca, 2006) also concluded that after controlling for unobservable student characteristics, attendance has a statistically significant and quantitatively relevant effect on student learning.

However, the result obtained at Australian University could not be replicated at Indian University where 70% attendance is compulsory for lectures and laboratories. The literature presents mixed response to this question. A study conducted by (Marburger, 2006) concluded that an enforced mandatory

attendance policy significantly reduces absenteeism and improves exam performance. On the contrary, the study conducted by (Credé, et al., 2010) concluded that class attendance is a generally desirable behaviour, and there is encouraging evidence that mandatory policies are not necessary for dramatically improving class attendance or class performance. Another study conducted by (Veerasamy et al., 2016) concluded that attendance doesn't positively correlate with students' performance in the exams. Since attendance is compulsory at Indian University, further study needs to be conducted to understand why attendance proves to be ineffective towards student marks.

The purpose of studying this factor is twofold: 1) to determine if attending lectures helps score better marks 2) to determine if making attendance mandatory leads to better performance. The results suggest that attending more lectures may lead to better marks; however making attendance mandatory may not necessarily improve student performance. Further study needs to be conducted to find out: 1) why attendance has no correlation with student marks at Indian University 2) The reasons behind the high correlation of marks with attendance at Australian University so that recommendations can be made to other Universities to improve students' scores in learning programming.

**Table 7: Mean (SD) exam scores by number of programming lectures attended for Australian University**

		Mean (SD)	p
Number of programming lectures	0%	52.50 (17.25)	0.002*
	Up to 20%	65.06 (19.65)	
	Up to 40%	59.09 (20.06)	
	Up to 60%	66.92 (21.39)	
	Up to 80%	57.77 (22.53)	
	100%	71.80 (22.55)	

**Table 8 : Mean (SD) exam scores by number of programming lectures attended for Indian University**

		Mean (SD)	P
Number of programming lectures attended	Up to 20%	50.00 (NA)	0.207
	Up to 40%	51.15	
	Up to 60%	58.79	
	Up to 80%	58.39	
	Up to 100%	70.46 (7.83)	

#### **4.5 Effect of activity in the lecture theatre on marks**

Table 9 summarises Mean (SD) exam scores by sources of help and activities in lecture for Australian University. The results of the Kruskal-Wallis tests are significant for the students who opted for "Listen to the lecture", but the results

of the pairwise comparisons are not significant at the 0.05 level. Thus, these factors are concluded not statistically significant to the exam scores.

Table 10 summarises Mean (SD) exam scores by sources of help and activities in lecture for Indian University. For “Look up for terms discussed in the lecture”, the results of the Kruskal-Wallis tests are significant, but the results of the pairwise comparisons are not significant at the 0.05 level. Thus, this factor is also concluded not statistically significant to the exam scores.

The results obtained from this study are in contrast with the results obtained by a study conducted by (Di Vesta & Gray, 1972) which suggested that “taking notes clearly led to an increase in the number of ideas recalled”. Further studies need to be conducted to investigate the impact of activities that students perform in lecture theatre on marks as certain factors were somewhat significant but not statistically significant.

**Table 9: Mean (SD) exam scores by sources of help and activities in lecture for Australian University**

**Note: 1 = never, 2 = sometimes, 3 = large part of lecture, 4 = whole lecture. SD = standard deviation. NA = not available. p = p-value of the Kruskal-Wallis tests. \*indicates significance at the 0.05 level. \*\* indicates significant at the 0.05 level for the Kruskal-Wallis tests, but not significant at the 0.05 level for the pairwise comparisons.**

What do you do in the programming lecture theatre?					
	1	2	3	4	p
Listen to the lecture	42.33 (8.02)	61.38 (23.74)	62.20 (21.43)	68.06 (22.52)	0.044**
Listen and make notes	62.39 (21.66)	60.63 (22.37)	64.30 (22.11)	68.09 (23.23)	0.314
Annotate if you have printed notes	63.25 (22.45)	65.51 (20.40)	62.82 (23.50)	57.75 (27.05)	0.798
Playing games on mobile phone/laptop	65.76 (22.43)	59.68 (21.57)	59.67 (19.05)	58.75 (33.39)	0.214
Look up for terms discussed in the lecture	64.27 (22.42)	63.56 (22.85)	60.63 (18.87)	53.40 (25.44)	0.582
Use social media	66.41 (21.88)	60.88 (21.73)	50.33 (23.16)	56.25 (32.63)	0.094
Browse internet general	65.90 (22.02)	61.09 (22.18)	64.56 (21.64)	55.00 (39.85)	0.471

**Table 10: Mean (SD) exam scores by sources of help and activities in lecture for Indian University**

**Note: 1 = never, 2 = sometimes, 3 = large part of lecture, 4 = whole lecture. SD = standard deviation. NA = not available. p = p-value of the Kruskal-Wallis tests. \*indicates significance at the 0.05 level. \*\* indicates significant at the 0.05 level for the Kruskal-Wallis tests, but not significant at the 0.05 level for the pairwise comparisons.**

What do you do in the programming lecture theatre?					
Listen to the lectures	60.75 (15.88)	64.33 (13.02)	57.35 (15.17)	59.00 (15.17)	0.581
Listen and make notes	53.75 (14.50)	61.78 (14.41)	55.91 (15.04)	65.57 (12.08)	0.173
Annotate if you have printed	66.88 (12.94)	56.72 (14.09)	55.04 (17.10)	65.88 (10.34)	0.061
Playing games on mobile	58.72 (14.47)	58.29 (15.40)	59.21 (15.52)	75.50 (NA)	0.725
Look up for terms discussed in the lecture	68.18 (12.79)	56.33 (14.42)	56.22 (14.53)	64.88 (14.91)	0.043**
Use social media	56.62 (15.00)	61.53 (14.78)	63.30 (8.12)	75.50 (NA)	0.320
Browse internet	57.47 (15.79)	61.38 (13.48)	49.46 (12.27)	63.25 (14.37)	0.227

#### 4.6 Effect of Preliminary preparation in improving performance of the students

Table 11 summarises Mean (SD) exam scores by habits of studying before lecture for Australian University, including “Study lecture slides related to the current lecture available on the course management system (CMS)”, “Study textbook slides related to the current lecture available on CMS”, “Study lecture slides related to the previous lecture available on CMS”, “Study textbook slides related to the previous lecture available on CMS”, “Read paper based textbook”, and “Do online tutorials/read about the topic to be covered online before lecture”. The results of the Kruskal-Wallis tests on the variables of “Lecture slides related to current lecture”, “Read paper based textbook”, and “Online tutorials” are significant, but results of the pairwise comparisons are not significant at the 0.05 level. Thus, it is concluded that these variables are not statistically significant to the exam scores.

Table 12 summarises Mean (SD) exam scores by habits of studying before lecture for Indian University, including “Study textbook chapter related to the current lecture”, “Study lecture slides from the previous lecture given by the lecturer”, “study textbook chapter related to the previous lecture”, “Read paper based textbook”, and “Do online tutorials/read about the topic to be covered online before lecture”. Similar to Australian University, there is no statistically significant difference in exam scores across any of the categories of habits of studying before lecture.

The results are similar for both universities which suggest that despite the differences, preliminary preparation before lecture has no effect on student performance. This is in contrast to the results achieved by (Chen & Lin, 2008) who studied the effects of downloading PowerPoint slides before lecture and concluded that downloading lecture slides before a class improved students' examination performance by 3.48%, suggesting that instructors could help students improve their academic performance by supplying PowerPoint slides. Another study was conducted by (Moravec, Williams, Aguilar-Roca, & O'Dowd, 2010) to explore the effects of learning before lecture on student performance in biology. The results showed that learning before lectures combined with interactive exercises can be implemented incrementally and results in significant increases in learning gains in large introductory biology classes.

There may be a few reasons for the contrasting results obtained through this study. As learning programming is considered different from learning other topics/subjects, it is possible that learning before lecture may not prove to be as effective in learning programming as it may be in learning other topics like biology, where it is more important to learn and understand facts whereas in learning programming the facts learnt have to be implemented through abstract reasoning. A study conducted by (Berg, 2016) suggests the use of programming practice tools in pre-lecture preparation to use the lecture time for instructor-supported high order thinking, which supports the proposition that doing some preparation before lecture may facilitate students in learning programming. Further studies need to be conducted to analyse the effect of preliminary preparation on student performance.

Some parameters prove to be somewhat effective for Australian University such as "Lecture slides related to current lecture", "Read paper based textbook", and "Online tutorials". These factors may be explored in the further studies with larger sample size in order to confirm the results. This proposal suggests the use of programming practice tools in formative assessments in a CS1 course to both increase the thinking skills required in pre-lecture preparation, provide feedback of progress to students and instructors and leave more time in lecture for instructor-supported, high-order thinking.

**Table 11: Mean (SD) exam scores by habits of studying before lecture for Australian University**

Note: N = 184. 1 = never, 2 = sometimes, 3 = often, 4 = very often, and 5 = always. SD = standard deviation. p = p-value of the Kruskal-Wallis tests. \* indicates significance at the 0.05 level. \*\* indicates significant at the 0.05 level for the Kruskal-Wallis tests, but not significant at the 0.05 level for the pairwise comparisons.

Frequency of studying before going to the programming lecture						
	1	2	3	4	5	P
Study lecture slides related to the current lecture available on CMS	67.69 (20.38)	56.52 (22.78)	63.02 (22.32)	72.21 (14.50)	66.44 (28.01)	0.024**
Study lecture slides related to the previous lecture available on CMS	62.58 (22.94)	61.33 (21.32)	64.13 (22.28)	66.33 (21.48)	73.14 (29.81)	0.325



Study lecture slides related to the previous lecture available on CMS	65.36 (21.27)	58.50 (23.66)	63.38 (22.08)	69.63 (18.26)	69.10 (25.72)	0.178
Study textbook slides related to the previous lecture available on CMS	62.90 (22.38)	59.58 (23.22)	70.10 (18.01)	67.84 (21.42)	60.67 (30.52)	0.320
Read paper based textbook	56.96 (24.39)	70.19 (15.61)	60.00 (24.10)	68.41 (19.11)	67.55 (24.37)	0.025**
Do online tutorials/read about the topic to be covered online before lecture	61.57 (22.11)	59.11 (22.70)	62.87 (22.79)	75.06 (15.34)	72.93 (23.63)	0.021**

**Table 12: Mean (SD) exam scores by habits of studying before lecture for Indian University. Note: N = 79. 1 = never, 2 = sometimes, 3 = often, 4 = very often, and 5 = always. SD = standard deviation. p = p-value of the Kruskal-Wallis tests. \* indicates significance at the 0.05 level. \*\* indicates significant at the 0.05 level for the Kruskal-Wallis tests, but not significant at the 0.05 level for the pairwise comparisons.**

Frequency of studying before going to the programming lecture						
	1	2	3	4	5	P
Study textbook chapter related to the current	59.66 (14.09)	58.51 (14.90)	57.93 (14.57)	60.50 (19.17)	NA	0.948
Study lecture slides from the previous	61.20 (15.77)	54.29 (14.55)	63.33 (12.78)	55.64 (15.07)	68.25 (12.37)	0.194
Study textbook chapter related to the previous	63.23 (15.24)	57.97 (13.97)	54.09 (13.97)	67.06 (10.06)	51.88 (35.89)	0.136
Study textbook chapter related to the previous	58.75 (14.64)	58.47 (17.21)	55.89 (14.40)	64.97 (7.77)	61.90 (18.23)	0.685
Do online tutorials/read about	60.63 (14.31)	61.89 (13.59)	56.06 (15.07)	54.78 (14.59)	62.29 (18.76)	0.491
watch content related to lecture on YouTube	59.66 (14.09)	58.51 (14.90)	57.93 (14.57)	60.50 (19.17)	NA	0.948

#### 4.7 Effect of Preliminary preparation before laboratory in improving performance of the students

Table 13 and Table 14 respectively show the mean exam scores by experience with programming and study habits before going to the laboratory for both universities. The parameters include "Study lecture slides related to the lab", "Study textbook slides related to the lab", "Read paper-based textbook", "Do online tutorials", "Read previous lab work", "Practice previous lab work", "Read new programs related to previous lab work", "Practice new programs related to previous lab work", "Read new similar programs related to the lab", and "Practice new similar programs related to the lab". The results show that there is no statistically significant difference in exam scores based on study habits before going to the laboratory ( $p > 0.05$ ) for both Australian University and Indian University, which suggest that despite the differences, preliminary preparation before laboratory has no effect on student performance.

**Table 13: Mean (SD) exam scores by study habits before going to the laboratory for Australian University**

What do you study before going to the laboratory?			
	Yes	No	P
Study lecture slides related to the lab	63.12 (21.13)	63.75 (24.29)	0.617
Study textbook slides related to the lab	62.69 (24.03)	63.95 (20.78)	0.920
Read paper-based textbook	66.56 (21.76)	61.06 (22.51)	0.092
Do online tutorials	65.41 (22.42)	61.98 (22.22)	0.261
Read previous lab work	62.85 (22.56)	63.81 (22.18)	0.784
Practice previous lab work	65.30 (21.42)	62.55 (22.69)	0.409
Read new programs related to previous lab work	62.28 (22.86)	63.72 (22.19)	0.750
Practice new programs related to previous lab work	68.30 (19.89)	31.31 (22.99)	0.052
Read new similar programs related to the lab	64.32 (22.15)	62.83 (22.46)	0.632
Practice new similar programs related to the lab	64.56 (23.75)	62.76 (21.63)	0.468

**Table 14: Mean (SD) exam scores by study habits before going to the laboratory for Indian University**

**Note: SD = standard deviation. p = p-value of the Wilcoxon ranked-sum tests.\* indicates significance at the 0.05 level.**

What do you study before going to the laboratory?			
	Yes	No	P
Study lecture slides related to the lab	55.97 (15.50)	60.50 (15.14)	0.218
Study textbook slides related to the lab	57.94 (15.90)	59.76 (15.78)	0.698
Read paper-based textbook	59.66 (15.98)	58.48 (15.23)	0.701
Do online tutorials	56.89 (14.81)	62.20 (14.45)	0.132
Read previous lab work	55.63 (15.82)	61.81 (15.28)	0.090
Practice previous lab work	58.40 (14.09)	59.28 (15.40)	0.784
Read new programs related to previous lab work	59.45 (14.51)	58.26 (15.05)	0.695
Practice new programs related to previous lab work	60.33 (14.89)	57.45 (14.58)	0.336
Read new similar programs related to the lab	56.02 (16.42)	60.06 (13.88)	0.365
Practice new similar programs related to the lab	55.65 (15.52)	60.59 (14.09)	0.154

#### 4.8 Effect of Revision on performance of students

Table 15 shows the mean exam scores by habits of revising programming topics for Australian University. The parameters include “During mid-semester break”, “During mid-semester exams”, “Both during mid-semester break and mid-semester exams”, and “Revised while the semester is in progress”. The results of the Kruskal–Wallis tests suggest that there is a statistically significant difference in exam scores among students with habits of revising programming topic while the semester is in progress ( $p = 0.005$ ). In particular, the results of pairwise comparisons suggest that students who always revise while the semester is in progress have statistically significantly higher exam scores ( $M = 74.76$ ,  $SD = 12.85$ ) than those who never ( $M = 55.50$ ,  $SD = 25.84$ ;  $p = 0.019$ ) or sometimes ( $M = 55.50$ ,  $SD = 25.84$ ;  $p = 0.045$ ) revise while the semester is in progress.

This result reflects the expectation that students who revise throughout the semester should perform better. The results of a study conducted by (Roddan, 2002) concluded that statistical evidence from the post exam questionnaire demonstrated that students who stated they had kept up with the course performed better in the exams. These findings combine to reinforce the view that students should make every effort to stay on top of the course and take care not to lag behind. (Ebbinghaus, 1985) proved the fact that revision improves learning in general. When a new piece of information is learned, at the beginning the rate of retention is 100% and with the passage of time the retention drops down to 40% in the first few days. The fact which is proved to be true in general also proves to be true for Australian University.

Table 16 shows the mean exam scores by habits of revising programming topics for Indian University. In contrast, there is no statistically significant difference in exam scores across any categories of habits of revising programming topics ( $p > 0.05$ ).

Further research needs to be conducted to find out the (cultural and non-cultural) reasons why “revision throughout the semester is ineffective for Indian University students but effective for Australian University students.

**Table 15: Mean (SD) exam scores by habits of revising programming topics for Australian University. Note: N = 183 for Q12. 1 = never, 2 = sometimes, 3 = often, 4 = very often, and 5 = always. For Q20, 1 = not at all, 2 = slightly useful, 3 = useful, 4 = very useful, and 5 = extremely useful. SD = standard deviation. p = p-value of the Kruskal–Wallis tests. \* indicates significance at the 0.05 level.**

Habits of revising the programming topic						
	1	2	3	4	5	p
During mid-semester break	62.12 (20.96)	60.46 (23.61)	65.50 (18.68)	70.62 (20.23)	60.67 (31.83)	0.411
Both during mid-semester break and mid-semester exams	61.47 (23.64)	64.38 (22.83)	61.25 (22.22)	66.32 (20.59)	69.16 (19.91)	0.609

Both during mid-semester break and mid-semester exams	60.71 (22.43)	63.67 (22.38)	63.10 (21.68)	62.04 (22.88)	71.92 (22.97)	0.485
Revised while the semester was in progress	46.17 (29.98)	60.10 (20.34)	68.47 (20.22)	61.97 (25.78)	74.76 (12.85)	0.005*

**Table 16: Mean (SD) exam scores by habits of revising programming topics for Indian University.**

**Note:** N = 79 for Q12. 1 = never, 2 = sometimes, 3 = often, 4 = very often, and 5 = always. For Q20, 1 = not at all, 2 = slightly useful, 3 = useful, 4 = very useful, and 5 = extremely useful. SD = standard deviation. p = p-value of the Kruskal-Wallis tests. \* indicates significance at the 0.05 level.

Habits of revising the programming topic						
	1	2	3	4	5	p
During mid-semester break	61.08 (16.05)	60.47 (15.23)	62.87 (13.81)	50.25 (14.91)	55.20 (10.34)	0.116
Both during mid-semester break and mid-semester exams	54.57 (12.52)	52.67 (15.02)	54.75 (17.86)	60.27 (13.89)	63.72 (12.22)	0.120
Both during mid-semester break and mid-semester exams	59.89 (19.03)	57.80 (14.77)	59.10 (15.16)	61.77 (14.93)	54.53 (14.93)	0.836
Revised while the semester was in progress	53.25 (14.78)	58.48 (15.38)	59.17 (13.92)	57.98 (16.45)	66.10 (12.33)	0.734

## 5. Conclusion and Future work

Some interesting results were achieved from the study. The prior programming experience had some effect on student performance at Australian University. The data analysis results of this study suggested that the gender of the students affected the performance of students at Indian University but not at Australian University. The reason to study programming had significant effect on the performance of students at Australian University but not at Indian University. The attendance had positive effect on the performance of students at Australian University. Any particular activity performed in the lecture theatre did not have any statistically significant effect on the performance of students at both Universities. The preliminary preparations before lecture had no effect on the performance of students at both Universities. The preliminary preparation before laboratory did not have any effect on the performance of students at both Universities. The effect of revision had a statistically significant effect on the performance of students at Australian University. The students always revising while the semester was in progress had statistically significantly higher exam scores than students never or sometimes revising while the semester was in progress. Based on the factors that positively affect learning programming demonstrated in this study, recommendations can be made for teaching and learning programming.

The differences in the results of two universities may be due to the difference in education culture at two universities. The study culture at Australian university is different from that at Indian university. Students follow different codes of conduct; at Australian university, students are allowed to bring and use their portable/mobile devices in the lecture theatre, which however are prohibited at the Indian university. Literature suggests that collaborative learning enhances learning experience (Teague & Roe, 2008). At Australian university, there is no provision of residence on campus so students make their own arrangement for residence. As students live independently and meet each other only during the lectures, laboratory sessions or workshops, the opportunities to collaborate with each other for study purposes are limited. In contrast, at Indian University, there is a paid provision of hostel and most of the students stay in different hostels. As a result, there are opportunities to collaborate informally and it is common for students to support each other when one asks for help.

Another cause of the difference in the results between two universities may be due to the difference in the exam structure, exam frequency, assessment criteria and number of lectures per week. Further research needs to be conducted to find out the aspects of education culture that caused the differences in results.

**We are conscious of the limitations of this work.** Large scale studies are required to validate the results achieved through this study. We also need to extend this study to more Universities in Australia and India in particular and more universities across continents in general to validate the results. The parameters which showed effect on either of the two Universities could be retained to see if they have similar or different effect on the other Universities if the sample size is increased. The Parameters from which nothing concrete could be concluded can be retained to determine their effect on a larger sample.

**Furthermore,** the methodology used is based purely on the quantitative research paradigm, and the effect of culture on factors affecting learning programming investigation can be further supplemented with some form of qualitative study. For example, the analysis can be supplemented with posterior interviews and group discussions to understand how the education culture affects the factors. This can be done in the future to gain a better insight into the topic.

## References

- Arulampalam, W., Naylor, R., & Smith, J. (2004). Factors affecting the probability of first year medical student dropout in the UK: a logistic analysis for the intake cohorts of 1980–92. *Medical Education*, 38(5), 492-503.
- Barker, R. J., & Unger, E. (1983). *A predictor for success in an introductory programming class based upon abstract reasoning development*. Paper presented at the ACM SIGCSE Bulletin.
- Bartlett, F. C., & Burt, C. (1933). Remembering: A study in experimental and social psychology. *British Journal of Educational Psychology*, 3(2), 187-192.
- Berg, C. (2016). *From buds to Bloom's: A proposal for growing strong students*. Paper presented at the Proceedings of the 21st Western Canadian Conference on Computing Education.
- Bergin, S., & Reilly, R. (2005). *Programming: factors that influence success*. Paper presented at the ACM SIGCSE Bulletin.

- Butler, M., & Morgan, M. (2007). Learning challenges faced by novice programming students studying high level and low feedback concepts. *Proceedings ascilite Singapore*, 99-107.
- Byrne, P., & Lyons, G. (2001). *The effect of student attributes on success in programming*. Paper presented at the ACM SIGCSE Bulletin.
- Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin*, 38(1), 27-31.
- Caspersen, M. E., & Kolling, M. (2009). STREAM: A first programming process. *ACM Transactions on Computing Education (TOCE)*, 9(1), 4.
- Chen, J., & Lin, T.-F. (2008). Class attendance and exam performance: A randomized experiment. *The Journal of Economic Education*, 39(3), 213-227.
- Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: Making thinking visible. *American educator*, 15(3), 6-11.
- Credé, M., Roch, S. G., & Kieszczynka, U. M. (2010). Class attendance in college a meta-analytic review of the relationship of class attendance with grades and student characteristics. *Review of Educational Research*, 80(2), 272-295.
- Cutts, Q., Fincher, S., Haden, P., Robins, A., Sutton, K., Baker, B., . . . Hamilton, M. (2006). *The ability to articulate strategy as a predictor of programming skill*. Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.
- de Raadt, M., Hamilton, M., Lister, R., Tutty, J., Baker, B., Box, I., . . . Haden, P. (2005). *Approaches to learning in computer programming students and their effect on success*. Paper presented at the Proceedings of the 28th HERDSA Annual Conference: Higher Education in a Changing World (HERDSA 2005).
- Di Vesta, F. J., & Gray, G. S. (1972). Listening and note taking. *Journal of educational psychology*, 63(1), 8.
- Dijkstra, E. W. (1970). Notes on structured programming: Technological University, Department of Mathematics.
- Ebbinghaus, H. (1985). Remembering Ebbinghaus. *Contemporary Psychology*, 30(7), 519-523.
- Fincher, S., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., . . . Petre, M. (2005). Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses.
- Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., . . . Lister, R. (2006). *Predictors of success in a first programming course*. Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.
- Gries, D. (1974). What should we teach in an introductory programming course? *ACM SIGCSE Bulletin*, 6(1), 81-89.
- Hagan, D., & Markham, S. (2000). *Does it help to have some programming experience before beginning a computing degree program?* Paper presented at the ACM SIGCSE Bulletin.
- Lau, W. W., & Yuen, A. H. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40(4), 696-712.
- Lau, W. W., & Yuen, A. H. (2011). Modelling programming performance: Beyond the influence of learner characteristics. *Computers & Education*, 57(1), 1202-1213.
- Lishinski, A., Yadav, A., Good, J., & Enbody, R. (2016). *Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance*. Paper presented at the Proceedings of the 2016 ACM Conference on International Computing Education Research.
- Lister, R. (2005). *One small step toward a culture of peer review and multi-institutional sharing of educational resources: a multiple choice exam for first semester programming*

- students. Paper presented at the Proceedings of the 7th Australasian conference on Computing education-Volume 42.
- Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., . . . Seppälä, O. (2004). *A multi-national study of reading and tracing skills in novice programmers*. Paper presented at the ACM SIGCSE Bulletin.
- Marburger, D. R. (2006). Does mandatory attendance improve student performance? *The Journal of Economic Education*, 37(2), 148-155.
- Massingham, P., & Herrington, T. (2006). Does attendance matter? An examination of student attitudes, participation, performance and attendance. *Journal of University Teaching and learning practice*, 3(2), 82-103.
- Mayer, R. E., Dyck, J. L., & Vilberg, W. (1986). Learning to program and learning to think: what's the connection? *Communications of the ACM*, 29(7), 605-610.
- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125-180.
- Montmarquette, C., Mahseredjian, S., & Houle, R. (2001). The determinants of university dropouts: a bivariate probability model with sample selection. *Economics of education review*, 20(5), 475-484.
- Moravec, M., Williams, A., Aguilar-Roca, N., & O'Dowd, D. K. (2010). Learn before lecture: a strategy that improves learning outcomes in a large introductory biology class. *CBE-Life Sciences Education*, 9(4), 473-481.
- Moström, J. E. (2011). A study of student problems in learning to program.
- ÖZYURT, Ö., & ÖZYURT, H. (2015). Clustering Analysis of Attitudes of Prospective Computer Programmers towards Programming. *International Journal of Learning, Teaching and Educational Research*, 10(4).
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172.
- Roddan, M. (2002). The determinants of student failure and attrition in first year computing science. *Computing Science, Glasgow University, project Summer*.
- Sharma, S. (2016). Delhi girl tops CBSE Class XII exams, pass percentage in private schools dips, *The Indian Express*. Retrieved from <http://indianexpress.com/article/education/sukriti-gupta-delhi-girl-tops-cbse-class-xii-exams-pass-percentage-in-private-schools-dips-2813094/>
- Sheard, J., Simon, S., Hamilton, M., & Lönnberg, J. (2009). *Analysis of research into the teaching and learning of programming*. Paper presented at the Proceedings of the fifth international workshop on Computing education research workshop.
- Simon, S., Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., . . . Hamilton, M. (2006). *Predictors of success in a first programming course*. Paper presented at the Proceedings of the Eighth Australasian Computing Education Conference (ACE 2006).
- Stanca, L. (2006). The effects of attendance on academic performance: Panel data evidence for introductory microeconomics. *The Journal of Economic Education*, 37(3), 251-266.
- Teague, D., & Roe, P. (2008). *Collaborative learning: towards a solution for novice programmers*. Paper presented at the Proceedings of the tenth conference on Australasian computing education-Volume 78.
- Tinto, V. (1975). Dropout from higher education: A theoretical synthesis of recent research. *Review of Educational Research*, 45(1), 89-125.
- Tolhurst, D., Baker, B., Hamer, J., Box, I., Lister, R., Cutts, Q., . . . Fincher, S. (2006). *Do map drawing styles of novice programmers predict success in programming?: a multi-national, multi-institutional study*. Paper presented at the Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.

- Veerasamy, A. K., D'Souza, D., Lind, R., Kaila, E., Laakso, M.-J., & Salakoski, T. (2016). The impact of lecture attendance on exams for novice programming students. *International Journal of Modern Education and Computer Science*, 8(5), 1.
- Wilson, B. C. (2002). A study of factors promoting success in computer science including gender differences. *Computer Science Education*, 12(1-2), 141-164.
- Wilson, B. C., & Shrock, S. (2001). *Contributing to success in an introductory computer science course: a study of twelve factors*. Paper presented at the ACM SIGCSE Bulletin.