Does Native Language Play a Role in Learning a Programming Language?

Adalbert Gerald Soosai Raj
Department of Computer Sciences and Education
University of Wisconsin-Madison
gerald@cs.wisc.edu

Kasama Ketsuriyonk
Department of Computer Sciences
University of Wisconsin-Madison
ketsuriyonk@cs.wisc.edu

Jignesh M. Patel
Department of Computer Sciences
University of Wisconsin-Madison
jignesh@cs.wisc.edu

Richard Halverson
Department of Educational Leadership and Policy Analysis
University of Wisconsin-Madison
rich.halverson@wisc.edu

ABSTRACT

Computer Science (CS) is taught in India, using English as the medium of instruction, to students whose native language is not English. This places a high cognitive load on students who learn programming for the first time and who are not very proficient in English. The problems these students face become even harder since learning to program can be an incredibly difficult task. Our study aims to find out if a student's native language has any effect on the student's ability to learn programming. We taught linked list, a basic concept in CS, to two groups of undergraduate students for a week in Tamil Nadu, India. We used *English* to teach one group of students and *English and Tamil* (the native language in Tamil Nadu) to teach the other group. Our intervention consisted of three lectures and one live-coding session. We collected quantitative and qualitative data using technical tests and open-ended feedback respectively. We found that although teaching programming using both English and Tamil is no different when compared to using only English with respect to student learning, students have expressed positive opinions about teaching and learning CS using two languages.

CCS CONCEPTS

•**Social and professional topics → Computer science education;**

KEYWORDS

CS Education; Native Language; Bilingual Education

ACM Reference format:
Adalbert Gerald Soosai Raj, Kasama Ketsuriyonk, Jignesh M. Patel, and Richard Halverson. 2018. Does Native Language Play a Role in Learning a Programming Language?. In *Proceedings of The 49th ACM Tech-nical Symposium on Computer Science Education, Baltimore, MD, USA, February 21–24, 2018 (SIGCSE'18),* 6 pages.
DOI: http://dx.doi.org/10.1145/3159450.3159531

SIGCSE'18, February 21–24, 2018, *Baltimore, MD, USA*
© 2018 ACM. ISBN 978-1-4503-5103-4/18/02…$15.00
DOI: http://dx.doi.org/10.1145/3159450.3159531

1 INTRODUCTION

Students in India learn their subjects during their K-12 (kindergarten (K) through twelfth grade (12)) either in English or their *native/first language* (i.e., the language that a person has spoken from earliest childhood). This choice of language depends on whether the student studies in an English-medium school or a native-language-medium school during their K-12. Although there are two mediums of instruction during K-12, almost all the STEM (Science, Technology, Engineering and Mathematics) subjects in undergraduate education are taught mostly in English [1, 12]. Computer Science, one of the STEM subjects, is also taught in Indian colleges in English. The main reason for teaching CS in English is because software companies in India use English as their language of communication as most of them are affiliated with U.S. based companies [3]. Therefore, students who aren't very comfortable in English (e.g., students who studied in a Tamil-medium school during their K-12 and students who aren't proficient with English even though they may have studied in an English medium school) find it difficult to understand programming concepts since the subject is already new to them and they are also forced to learn it in a language that they are not comfortable with [18, 22]. As a result, these students end up failing their programming courses, and eventually develop inferiority complexes about their programming abilities [17].

Our research started with the following question: Can we reduce the difficulties faced by these students by teaching them CS using both English and their native language? To answer this question and to better understand the impact of the medium of instruction to teach CS, we conducted an experiment where we taught programming using both *Tamil and English* to a group of students (experimental group) whose native language was Tamil. We also taught programming to another set of students (control group) *only in English* even though their native language was also Tamil. We used *English along with Tamil* to teach the experimental group since we believe that even though Tamil may help students to better understand programming concepts, English is needed for them to communicate with other programmers around the world. Moreover since the documentation of the programming languages (e.g., K&R C [13]) and the online programming forums (e.g., Stack Overflow) are in English, it is very important for these students to learn English along with programming.

To teach CS using two languages, we use techniques like *code-switching* [8] and *translanguaging* [9, 16] while communicating with the students. *Code-switching* is a technique of alternating between two or more languages in the same conversation. Code-switching, especially from the secondary to the primary/native language helps the students to focus, clarify and reinforce lesson materials that leads to better understanding of the subject [8]. *Translanguaging* is a pedagogical practice where certain aspects of the pedagogy (e.g., classroom discussion) are done in the primary language and other aspects (e.g., writing, exams, etc.) are done in the secondary language. This helps the students to gain a deeper understanding of the subject matter and to improve their proficiency with the secondary language [2].

The reason why bilingualism will be beneficial in CS is because it is one of the subjects where students will be greatly benefited if they understand the meaning of keywords. Many of the concepts in programming languages would be easy to remember and recall if students understood the meaning of those words in English. Examples of such concepts/keywords are static, abstract, virtual, inheritance, polymorphism, encapsulation, etc. The number of non-native speakers of the English language (as of 2003) around the whole world outnumbered the native speakers by a ratio of 3 : 1 [6]. We consider our study to be a small first step towards broadening the impact of computing education not just to predominantly English-speaking regions but to the entire world.

2 RELATED WORK

Pal and Iyer analyzed the effects of medium of instruction on acquiring programming abilities among students in North India whose native language is mostly Hindi [19–21]. Programming was taught to two groups of first-year undergraduate students from Hindi-medium background (i.e., students who studied in Hindi-medium schools during their K-12) using English-only, and Hindi-only medium of instruction. The programming abilities of the two groups of students were tested and compared, and it was found that the students from Hindi-medium background learnt programming better when they were taught in Hindi when compared to when they were taught in English. Our work differs from this work in various aspects. First of all, the native language that we used in our study was Tamil (a South-Indian language, spoken by people in Tamil Nadu) which is a completely different language than Hindi (spoken by people in many parts of India but not in Tamil Nadu). We don't use Tamil-only to teach programming but instead we use a combination of Tamil and English to teach programming since we recognize the benefits of using both languages. We don't target students who are from a Tamil-medium background but instead we target all students whose native language is Tamil, even though they may have studied in an English-medium school during their K-12. We compared the effectiveness of our approach by using the gain and the post-test score while Pal et.al. used only the post-test score to compare the effectiveness of their approach even though they had conducted a pre-test in their study.

Lau and Yuen studied the impact of medium of instruction on teaching and learning computer programming [14]. They conducted their study among two groups of K-12 students from Chinese-medium and English-medium schools in Hong Kong. They taught bubble sort using either C or Pascal to these students, and reported that Chinese-medium students appear to understand programming concepts better than their English-medium counterparts. In our study, the students were not separated into two groups based on their medium of instruction during K-12 since CS is taught using English as the medium of instruction in undergraduate colleges in India, irrespective of the students' prior medium of instruction.

Boulet studied the role of language in teaching and learning of Mathematics [4]. The author addresses some specific issues pertaining to languages that students use to define mathematical terms to read and interpret mathematical notations, and to describe mathematical processes. The teacher's role to foster productive mathematical discourse in the classroom using their language as a tool is also highlighted. In our study too, we used all the technical terms in CS as it is in English. For example, the terms like *linked lists*, *arrays*, *pointers*, *structures*, etc, were used as they are used in English. Tamil was used only to explain the programming concepts.

Probyn interviewed some teachers in South Africa who use English along with Xhosa, an official South African language, to teach Science [22, 23]. The study shows that the language of learning and teaching frequently creates a barrier to learning when it is not the native language of the learners. The benefits of code-switching between the two languages for increased comprehension among students is also highlighted. We consider our study as an extension to this study where we try to find if the vernacular has any effect on learning programming, mainly by measuring student learning using pre-and-post technical tests.

Fennema-Bloom studied the value of naturally occuring code-switching during bilingual content instruction in Mandarin/English among non-traditional immigrant high school students [8]. The author found that code-switching is a valuable pedagogic tool used by bilingual teachers to make content more comprehensible. Our work targets on finding the value of code-switching for CS Education.

3 RESEARCH QUESTIONS

Our research aims at addressing the following questions:

(1) Does using the native language (Tamil) along with English for teaching programming have any effect on student learning when compared to teaching only in English?

(2) What are the students' opinions about using their native language along with English for teaching programming within the classroom?

Our *hypothesis* is that using the native language (Tamil) along with English for teaching programming will have a positive impact on student learning when compared to using only English, for first year undergraduate students whose native language is Tamil.

4 METHODOLOGY

In this section, we explain the methodology that we used to conduct the experiment and to collect the data.

4.1 Participants

The experiment was conducted in a well reputed Engineering college in Tamil Nadu. Two groups of first-year students, enrolled in two different sections of a data structures course were selected for the study. One group was treated as the control group and the other group was treated as the experimental group. The total number of students in the control and experimental group were 52 and 51 respectively. All these students have previously taken a programming course in C [13] and data structures was their second programming course. There was only one student in the experimental group who was from a Tamil-medium background during K-12 and there were two such students in the control group.

4.2 Experimental Design

The experiment was conducted using a *nonrandomized control group pre-test post-test design* [7]. In this design, the participants were not randomly assigned to groups but remained in their pre-assigned groups. This increases the external validity of the design by reducing the reactive effects of the experimental procedure [7]. We acknowledge that the problem with this approach is that even if there are any post-test differences between the groups, they may be attributed to characteristic difference between the groups rather than to the intervention. We take this issue into consideration in our result analysis by choosing our statistical models for pre-test post-test comparison very carefully (see Section 5 for more details).

The reason behind using this experimental design is that we didn't want the students to know that they were part of an experiment comparing the effects of teaching programming using their native language along with English versus using only English. We felt that if students knew these details about our experiment before our intervention then the results may be unreliable since some students may try to bias their test results based on their preferred method for learning programming.

We coordinated with the two teachers who were teaching data structures for these two groups so that we can teach linked lists at the appropriate time during the course. The teachers introduced the researcher (and the observer) as guest lecturers to the students. The researcher taught the students and the observer took field notes during the intervention. After our intervention was over, we informed the students that they were part of a teaching intervention and gave them the option to withdraw their test and feedback data, if they didn't want them to be included in our study. No student withdrew from our experiment.

4.3 Experimental Procedure

The following activities were performed with both the control group and the experimental group as a part of our intervention. There was a pre-test, three in-class lectures, a live-coding session, and a post-test. The programming (coding) was done in C [13] (a high-level programming language) and the questions in the pre-test and the post-test were in English for both groups. We collected an open-ended feedback from the students in both groups to understand what they felt about our intervention.

4.3.1 Pre-test. A pre-test was conducted to determine the students' understanding of key programming concepts like *pointers*, *structures*, and concepts about *stack and heap memory* in C programming. There were a total of 10 questions in the pre-test. There were 4 questions on pointers, 2 questions on structures, 2 questions on stack and heap memory, and 2 questions on some basic concepts in C. The pre-test was conducted for a total of 10 points, one point for each question. The pre-test (and post-test) questions were chosen from a set of frequently asked programming interview questions [11].

These topics were tested in the pre-test as they are the pre-requisites for learning linked lists in C. We also wanted a way to determine how much the students learnt during the one week session, so the pre-test scores were used as a baseline for each student. The complete pre-test can be found at this link: http://bit.do/pretest_questions.

4.3.2 Three lectures on linked lists. Three classroom-based lectures, each of 50 minutes duration, were taught for both groups. The basics of linked lists were taught in those three lectures. Topics discussed were: declaring a node structure for the linked list, adding a node at the beginning of the linked list, deleting a node from a linked list, calculating the length of a given linked list, printing a linked list, etc. The same topics were taught to both the groups.

The main differences between the lectures for the two groups were the following: The lectures were taught *only in English* for the students in the control group. Also, the students in the control group were required to communicate with the instructor and their classmates during the lecture only in English.

On the other hand, the lectures were taught using both *English and Tamil* in the experimental group, and the students were free to communicate in any of those two languages, whichever they felt more comfortable with. The instructor used both English and Tamil nearly equally (i.e., 50% time in English - 50% time in Tamil) while teaching the experimental group. Based on the observer's field notes, we found that more than 90% of the questions that were asked by the students in the experimental group during the lecture were in Tamil. The instructor answered the questions during the lecture using the same language (either English or Tamil) in which they were asked.

The instructor used *code-switching* between Tamil and English in the following way. He used English to introduce a topic, to explain the syntax, and to explain some technical terms (e.g., self-referential structures). He switched to Tamil whenever he felt that a particular topic needed detailed explanation in order to help the students understand the idea in a better way (e.g., How to change the head of a linked list when we add an element at the beginning?). The instructor used *translanguaging* as follows. He used Tamil only for oral explanations, discussion, and answering students' questions. He wrote all the content (e.g., code snippets, definitions, etc.) on the chalk-board during the lectures only in English.

4.3.3 Live coding session. Following the three classroom-based lectures on linked lists, a live-coding session [25] was conducted for about 90 minutes. The instructor projected his laptop on a screen, and wrote C code for the following linked list functions from scratch: adding a node at the beginning of the linked list, printing all the elements in the linked list, deleting all the nodes from the linked list.

The instructor was thinking aloud throughout the live-coding session. He showed the students how he would go about writing the code for these three functions. He also showed them some common sources of errors while writing code for linked lists. The content taught during the live-coding session was the same for both the control group and the experimental group.

4.3.4 Post-test. The post-test consisted of 15 questions on linked lists from the following three categories:

(1) Syntax of linked lists (5 questions - 1 point each)
(2) Conceptual understanding (5 questions - 2 points each)
(3) Predict the output (5 questions - 3 points each)

All the questions on the post-test were based on the material taught during the classroom-based lectures and the live-coding session. The post-test was conducted for a total of 30 points. The complete post-test can be found at this link:
http://bit.do/posttest_questions.

4.3.5 Open-ended feedback. We collected open-ended feedback from the students about our bilingual teaching methodology. All the students in both groups wrote their feedback in English, although the choice of language for the feedback was left to the students. Sample feedback from students is shown in Section 5.3.

5 RESULTS

The mean of the pre-test scores and the post-test scores for the two group are shown in Table 1 and Table 2 respectively.

Table 1: Mean of pre-test scores for the two groups

Group	N	Mean	Std. Dev.	Std. Error of Mean
Control	52	33.9	15.2	2.1
Experimental	51	25.3	14.5	2.0

Table 2: Mean of post-test scores for the two groups

Group	N	Mean	Std. Dev.	Std. Error of Mean
Control	52	76.7	17.3	2.4
Experimental	51	65.4	17.4	2.4

The mean of the pre-test score for the control group is higher than the mean of the pre-test score for the experimental group (see Table 1). This indicates that the control group and the experimental group were at different levels with respect to the prior knowledge in programming before our intervention. We were aware that this issue may occur in our results since we used a *nonrandomized control group pre-test post-test design.* In order to account for these initial differences between the two groups, we did the following to measure student learning:

(1) Compared the *gain scores* using *Independent Samples t-test*
(2) Compared the *post-test scores* using *ANCOVA (ANalysis of COVAriance)* with the pre-test score as a covariate.

According to the literature on measuring change using a pre-test and post-test, both the gain scores [26] and the post-test scores [15] are considered to be good indicators of student learning [5]. Hence, we compared both these metrics using appropriate tests of significance to find out if there was any statistically significant difference between the two groups due to our intervention. We used an alpha level of .05 for all statistical tests.

5.1 Comparison of Gain Scores

The mean of the gain scores for the two groups are shown in Table 3. The mean gain for the control group is higher than that of the experimental group. We performed Independent Samples t-test [10] to compare the gain scores between the control group and the experimental group and found *no significant difference* in gain scores (t(103) = .702, df = 101, p = .485).

Table 3: Mean of gain scores for the two groups

Group	N	Mean	Std. Dev.	Std. Error of Mean
Control	52	42.8	18.8	2.6
Experimental	51	40.1	20.1	2.8

5.2 Comparison of Post-Test Scores

The pre-test scores of the students in both the groups varied significantly before our intervention. We performed Independent Samples t-test to compare the pre-test scores between the control group and the experimental group and found *a significant difference* in pre-test scores (t(103) = 2.92, df = 101, p = .004). Therefore, we cannot directly compare the post-test scores between the two groups using Independent Samples t-test as we did for the gain scores [7]. Instead we use a statistical model to analyze and find if there was a significant effect on the post-test score of a student (response variable) because of the student's group (categorical variable) while controlling for the pre-test score of the student (predictor variable). Hence, we performed ANalysis of COVAriance (ANCOVA) [10] on post-test scores of the two groups as described below.

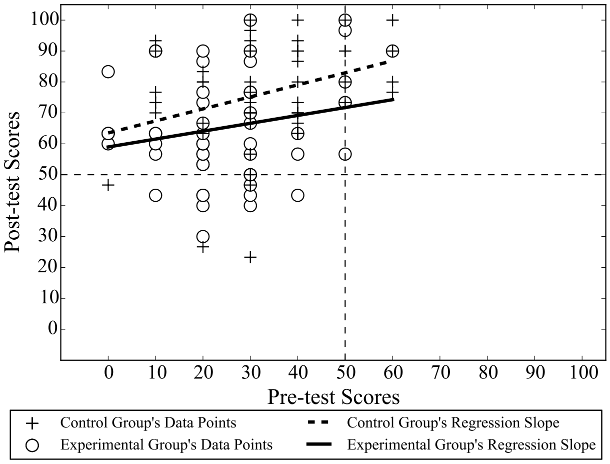

Figure 1: A scatter plot showing each student's pre-test score plotted on the x-axis and the post-test score plotted on the y-axis. The regression lines are also plotted for each group which shows the relationship between the pre-test and post-test score.

We plot a scatter plot in Figure 1 using the pre-test and the post-test scores of the students in both the groups. In this scatter plot the x-axis represents the pre-test scores of the students and the y-axis represents the post-test scores of the students. The two lines shown in this scatter plot are the regression lines for a particular group that summarizes the relationship between the post-test score and the pre-test score for that group. The regression lines for both the groups have nearly equal *positive slopes* (0.26 for the experimental group and 0.39 for the control group) which means that our intervention had a similar positive effect on both the groups. The y-intercept of the control group's regression line (63.51) is slightly higher than the y-intercept of the experimental group's regression line (58.98) which means that for a given pre-test score, a student in the control group had a better post-test score when compared to a student in the experimental group. The homogeneity of the regression slopes tells us that it is valid to create an ANCOVA regression model for finding out if there was any significant difference between the two groups due to the native language.

The results from our analysis of covariance are shown in Table 4. The p-values shown in this table are for the interaction of each variable with the response variable (i.e., post-test score).

Table 4: ANCOVA Results

Variable	p-value
Pre-test score	0.000324
Group	0.016056
Interaction between the pre-test score and the group	0.554225

The *pre-test score of a student* had a *significant effect on the post-test score* of the student after controlling for the effect of the student's group (p-value = 0.000324). This means that irrespective of the group that a student belongs to, generally, if a student did well on the pre-test then that student also did well on the post-test.

The *group of a student* had a *significant effect on the post-test score* of the student after controlling for the effect of the student's pre-test score (p-value = 0.016056). This is due to the fact that even before our intervention, the control group was better than the experimental group with respect to the knowledge of programming.

The interaction between the *pre-test score and the group of a student* has *no significant effect on the post-test score* of that student. In other words, the post-test scores of two students with the same pre-test score were not statistically significant based on the student's group. This means that the post-test scores of the two groups are not statistically significant due to our intervention.

5.3 Sample Students' Feedback

Tamil was used to explain difficult concepts in the experimental group. All the students' comments about the usage of Tamil to explain programming were very positive. Selected comments include: *"Your class made me more attentive in the class."; "The lecture was really very useful and it was easy to understand since the mixture of English and Tamil language helps us to learn better."; "The usage of both the languages Tamil and English actually kept us engaging."*

Some students in the experimental group also mentioned that they *felt more comfortable within the classroom* during our intervention. We believe that the usage of the native language within the classroom may be one of the main reasons for these students to have felt this way. Some sample students' comments: *"It is very helpful for us to understand the concept. I really felt comfortable in learning this topic that taught us in both the languages. "; "This class was just rocking, we feel comfortable when we are taught in both the language. As far from my part linked list is the toughest portion in C programming, but you made us understand easily. There was full freedom throughout the class (with respect to using the language of our choice)."; "Really the lectures are awesome. It is very useful for us. The method is really good. I feel very comfortable and it is little easy to understand when compared to normal teaching (using only English)."; "The lecture was simply awesome. True to my heart it was excellent. It was very comfortable for me to learn programming in this way."*

The feedback from the control group was also positive about our intervention. A detailed analysis of the comparison of students' sentiments between the two groups can be found in this study [24].

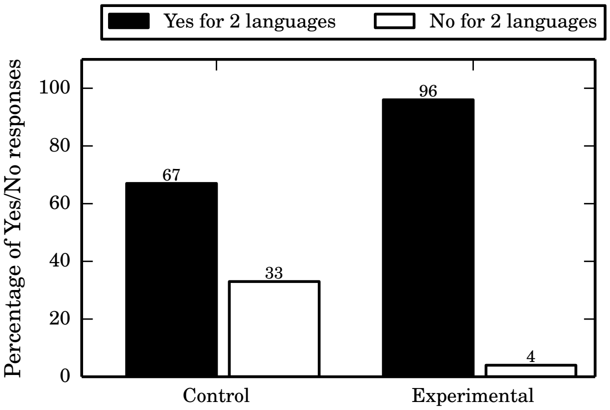

Figure 2: The student responses to the survey question "If given an option would you learn programming in a classroom where the teacher would use both English and Tamil to teach the concepts in programming?"

We also asked the following question in the feedback that we collected from the students at the end of our intervention: *If given an option would you learn programming in a classroom where the teacher would use both English and Tamil to teach the concepts in programming?*

This question was asked to students in both the groups, even though we didn't use Tamil in the control group. We wanted to get a sense of what might be the difference in students' opinions about learning using two languages between the two groups, one of them exposed to such an intervention while the other wasn't.

The student responses for this question are shown in Figure 2. The percentage of students favouring the use of both the languages was higher in both the groups.

6 DISCUSSION

Our study tried to find if using the native language (Tamil) along with English for teaching programming had any effect on students' learning of programming when compared to using only English. We measured the student learning in terms of gain scores and post-test scores.

The difference between the two groups with respect to the gain score is not statistically significant (see Section 5.1). This shows that there is no difference between teaching programming using only English and teaching programming using the native language (Tamil) along with English.

Also, the difference between the two groups with respect to the post-test score (with the pre-test score as a covariate) is not statistically significant (see Table 4). Therefore, based on these ANCOVA results we understand that since our intervention had similar positive effects on both the groups, we need more quantitative experiments in this area to better understand the impact of the native language on students' understanding of programming.

This further confirms that there is no difference between the two teaching methodologies on student learning.

Our finding that using the native language (Tamil) along with English is no different from using only English for teaching programming is different from the findings of Pal and Iyer [19–21] which suggests that the native language (Hindi) had a significant positive effect on students who did their schooling in a Hindi-medium school. The main reason for this difference is that in our study even though the native language of all students was Tamil, the majority of students (100 out of 103 students) studied in an English-medium school during their K-12. On the other hand, 50% of students in Pal and Iyer's study were from a Hindi-medium background. Also, while their study compared the difference between using Hindi-only and English-only medium of instruction, our study compared the difference between using two languages (English and Tamil) with English-only medium of instruction.

The feedback received from the students about using their native language (Tamil) along with English for teaching programming were completely positive. No student had any issues or concerns with using their native language for teaching programming within the classroom although they were only used to an English-only medium of instruction within the classroom until now. Also, many students in the experimental group have expressed that they felt very comfortable during our lectures since we used both the languages for instruction within the classroom. The students in the experimental group have expressed more positive opinions about learning using two languages when compared to those in the control group. We believe that the reason for this increased interest among the experimental group may be due to the fact that they were exposed to the teaching methodology of using two languages while the control group wasn't. Therefore, our other finding is that almost all the students exposed to our bilingual teaching methodology have preferred it over the current practice of teaching only in English.

One limitation with our study is that since we used the *non-randomized control group pre-test post-test design*, the two groups varied significantly with respect to their programming knowledge before the start of our intervention. Although we have taken this pre-test difference into consideration in our analysis of results, we acknowledge that the results would have been more reliable if these initial differences didn't exist among these groups. To minimize the effects due to the initial differences among the two groups, as a part of our future work, we plan to conduct more controlled experiments with a *randomized control group pre-test post-test design* [7] to better understand the effects of the native language for learning programming. Another limitation with our study is that we did not control for the students' English proficiency which we plan to do in our future experiments by conducting tests for English proficiency.

7 CONCLUSIONS

Our study didn't find any significant impact of using Tamil along with English on student learning of linked lists. We conclude that we need more studies in this area to verify if there is any significant difference in student learning between teaching programming using only English and using two languages (English and Tamil). We also conclude that, even though the effectiveness of using Tamil and English for teaching programming with respect to student learning is still unknown, students have expressed positive opinions about learning programming using the two languages. We believe that our work has potential implications in making CS education accessible to everyone around the world irrespective of their native language.

ACKNOWLEDGMENTS

We thank Prof. Saswati Mukerjee and Prof. Annie Bennet for helping us conduct experiments in India. Our sincere thanks to Prof. Michael Swift, Devin Wixon, Harshad Deshmukh, Ayebuke Turker, and Kaviya Lakshmipathy for their valuable feedback on our work. We also thank the anonymous reviewers for their comments on the earlier versions of this paper.

REFERENCES

[1] E Annamalai, JW Tollefson, and ABM Tsui. 2004. Medium of power: The question of English in education in India. *Medium of Instruction Policies–Which Agenda? Whose Agenda?* (2004).
[2] Colin Baker. 2003. Biliteracy and transliteracy in Wales: Language planning and the Welsh national curriculum. *Bilingual education and bilingualism* (2003).
[3] Peri Bhaskararao. 2002. English in contemporary India. *ABD* (2002).
[4] Genevieve Boulet. 2007. How does language impact the learning of mathematics? Let me count the ways. *Journal of Teaching and Learning* (2007).
[5] Edward Roland Cimler. 1973. *Gain vs. posttest scores as predictors of future performance.* Master's thesis. Citeseer.
[6] David Crystal. 2012. *English as a global language.* Cambridge university press.
[7] Dimiter M Dimitrov and Phillip D Rumrill Jr. 2003. Pretest-posttest designs and measurement of change. *Work* (2003).
[8] Jennifer R Fennema-Bloom. 2009. Code-scaffolding: A pedagogic code-switching technique for bilingual content instruction. *Journal of Education* (2009).
[9] Ofelia García and Claire E Sylvan. 2011. Pedagogies and practices in multilingual classrooms: Singularities in pluralities. *The Modern Language Journal* (2011).
[10] Gene V Glass and Kenneth D Hopkins. 1970. *Statistical methods in education and psychology.* Prentice-Hall Englewood Cliffs, NJ.
[11] Sandeep Jain. 2017. GeeksforGeeks: A computer science portal for geeks. http://www.geeksforgeeks.org/. (2017). http://www.geeksforgeeks.org/
[12] N Jayaram. 1993. The language question in higher education: trends and issues. *Higher Education* (1993).
[13] Brian W Kernighan and Dennis M Ritchie. 2006. The C Programming Language. (2006).
[14] Wilfred WF Lau and Allan HK Yuen. 2011. The impact of the medium of instruction: The case of teaching and learning of computer programming. *Education and Information Technologies* (2011).
[15] Frederic M Lord. 1956. The measurement of growth. *ETS Research Report* (1956).
[16] Catherine M Mazak and Claudia Herbas-Donoso. 2015. Translanguaging practices at a bilingual university: A case study of a science classroom. *Bilingual education and bilingualism* (2015).
[17] T Murugavel. 2011. The Problems of Non-English Medium Engineering Students and Possible Solutions. (2011). http://worldlitonline.net/the-problems-of-non-english.pdf
[18] Yogendra Pal. 2016. *A Framework for Scaffolding to Teach Programming to Vernacular Medium Learners.* Ph.D. Dissertation. IIT, Bombay.
[19] Yogendra Pal and Sridhar Iyer. 2012. Comparison of English versus Hindi Medium Students for Programming Abilities Acquired through Video-Based Instruction. In *T4E*. IEEE.
[20] Yogendra Pal and Sridhar Iyer. 2015. Classroom Versus Screencast for Native Language Learners: Effect of Medium of Instruction on Knowledge of Programming. In *ITiCSE*. ACM.
[21] Yogendra Pal and Sridhar Iyer. 2015. Effect of medium of instruction on programming ability acquired through screencast. In *LaTiCE*. IEEE.
[22] Margaret Probyn. 2001. Teachers voices: Teachers reflections on learning and teaching through the medium of English as an additional language in South Africa. *Bilingual Education and Bilingualism* (2001).
[23] Margaret Probyn. 2005. Learning science through two languages in South Africa. In *The 4th International Symposium on Bilingualism, Somerville, MA.*
[24] Adalbert Gerald Soosai Raj, Kasama Ketsuriyonk, Jignesh M Patel, and Richard Halverson. 2017. What Do Students Feel about Learning Programming Using Both English and Their Native Language?. In *LaTICE 2017*. IEEE.
[25] Marc J Rubin. 2013. The effectiveness of live-coding to teach introductory programming. In *SIGCSE*. ACM.
[26] Donald W Zimmerman and Richard H Williams. 1982. Gain scores in research can be highly reliable. *Journal of Educational Measurement* (1982).