Effect of medium of instruction on programming ability acquired through screencast

Yogendra Pal
Inter-Disciplinary Program in Educational Technology
Indian Institute of Technology Bombay
Mumbai, India
yogendra.pal3@gmail.com

Sridhar Iyer
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
sri@iitb.ac.in

***Abstract*— Students who have studied in their native language face difficulty in acquiring programming skills through English medium instruction. One solution for this is to create screencasts of live coding along with instructor explanations in the native language. In this paper we examine the impact of screencast based self-paced learning for Hindi medium (native language) students. Our study had three groups - One group had Hindi as the medium of prior instruction, and learned programming through screencasts in Hindi. Another group, also had Hindi as the medium of prior instruction, but learned through screencasts in English. The third group had English as the medium of prior instruction, and learned through screencasts in English. We compared the performance of the three groups using a post-test having items of fact, concept and process content types. We found that Hindi medium students who studied from Hindi screencasts performed as well as English medium students who studied from English screencasts. Both groups performed better than Hindi medium students who studied from English screencasts. The effect of medium of instruction was significant for the content types of fact, process and concept.**

Keywords—Computer programming education; screencast; native language instruction

I. Introduction

India has a large number of students who study in their native language in K-12 and then have to adapt to English language instruction for their undergraduate education [1]. These students have significant difficulty in acquiring programming skills, as evidenced by their low success in the university exams [2] [3].

While there is work on the effectiveness of using native languages for classroom teaching of computer science subjects [4] [5] [6] and programming [7], there are only a few studies on use of native language in screencast-based teaching [8] [9]. However there is little experimental data on the effect of teaching programming using native language screencasts, on student achievement. Towards this end, we conducted the following study.

We created six screencasts on introductory programming, in English and Hindi (native language). We then chose a reputed engineering college that had a mix of students from English and Hindi medium schools. We created 3 groups of students. The control group (HE) had Hindi medium students watching the screencasts in English. The experimental group (HH) had Hindi medium students watching the corresponding Hindi screencasts. As a baseline group (EE), we had English medium students also watching the English screencasts. We carried out the treatment over six days for topics taken from ACM CS curriculum [10], as shown in Table I.

TABLE I. Topics of each day screencasts

Topic number	Subtopics of each topic (comma separated)	Day
T1	Introduction to programming, program, development process	1
T2	Identifiers, data type, memory representation, integer, use of variable.	2
T3	Arithmetic instructions, operators, operators precedence	3
T4	printf, scanf	3
T5	Relational operators, equality operators, branching statement, if, if-else	4
T6	Functions, Function call, pass by value, return types	5
T7	Recursion	6

We measured differences between the groups using a post-test of 59 items, across different content types such as fact, concepts and process. We found that the difference between the experimental group (HH) and control group (HE) was statistically significant, while the difference between experimental group (HH) and baseline group (EE) was not significant. We further analyzed the data as per their content types [11], i.e., fact, concept, process, procedure and principle. We found that native language instruction is essential for the content types of fact, process and concepts.

In Section II we present the related work on teaching programming through native language instruction. In Sections III and IV, we give the details of our research questions and methodology, respectively. The results are in Section V, followed by discussion in Section VI.

II. Related Work

There are studies that show the benefit of native language instruction, in mathematics [12] and physics [13], However there is not much experimental work in the context of teaching

978-1-4799-9967-5/15 $31.00 © 2015 IEEE
DOI 10.1109/LaTiCE.2015.38

programming to native language learners. We found only one work [7], that suggests a bilingual model for teaching programming to undergraduate students in China, but it is still being implemented and no experimental data is available. Moreover it uses a classroom-based model, while we propose a screencast based approach.

Some benefits of using screencast, such as low cost and high availability, are well-known [14]. Screencasts also provide several learning benefits over traditional classroom environment [15], such as the facility to rewind and replay. While learning from screencast students give more time to understand the material than taking notes as compared to classroom environment because they know that screencasts will be available to them for multiple viewing [15]. Some distractions from a traditional classroom environment, such as noise, inability to hear the instructor, or inability to see the board, are also mitigated while learning from screencast [16].

In a study on text vs screencast based instruction on statistics [17], authors found that students who were watching screencasts took less time to learn statistics concepts, less time to solve the problem, and scored higher as compared to those who learn from text based instructions.

While students and teachers agree that screencasts should be created in native languages [18], there is no work on the effect of native language screencasts on student achievement, in the context of programming. In this paper we explore the use of screencast for teaching programming to native language learners.

III. Research Questions

We use the term "medium" to denote the medium of instruction in K-12 years of schooling. In our experiment, the medium could be the same as the native language (Hindi) or different (English). We use the term "MoI" to denote the medium of instruction in the treatment. In our study, the MoI for the screencast is either English or Hindi.

At a broad level, our question is: What is the impact of the MoI on the programming abilities of native language learners? This is operationalized into the following specific question:

Do undergraduate Hindi medium students learning introductory programming by watching screencasts in Hindi, perform better than similar students who watch the same screencast in English?

In order to determine which content types should be taught in Hindi and which content types could be taught in English, we had the additional question:

What is the effect of MoI for varying content types in screencast based programming instruction?

IV. Methodology

A. Sample

The sample consisted of 105 engineering 1st year undergraduate students of North India. The sample was divided into 3 groups according to their prior medium of instruction and medium of instructions in the treatment (MoI), as shown in Table II.

TABLE II. Medium of Instructions for various groups

Medium in K-12	MoI	Group	N
Hindi	Hindi	HH (Experimental)	35
Hindi	English	HE (Control)	35
English	English	EE (Baseline)	35

We included only those learners who are studying programming in their current semester. Moreover we used purposive sampling, i.e., participation was made voluntary thereby excluding students who are not interested in learning programming. Further, we selected only those students who had no or little prior knowledge of programming. We ensured equivalence of the groups on prior academic achievement.

B. Instruments and Data Collection

To measure programming ability, performance scores on a post-test were collected. To determine prior knowledge of programming a 10-item pre-test was conducted. To determine prior academic achievement levels, overall percentage of marks in 12th grade final examination were collected.

We used a 3-item survey to collect data about students' background. The items for each student were: (i) MoI in 12th standard (English or Hindi), (ii) Overall percentage of marks in 12th standard, and (iii) Whether they have prior knowledge of programming (yes or no). We verified their self-reported knowledge of programming using the pre-test.

We used paper-based post-test every day after workshop. We looked for a concept inventory for programming but found that the standardization of assessment instruments for programming ability is still ongoing [19]. So we created the post-test based on questions that typically appear in the University exams and those given in standard textbooks. We included only those questions that directly mapped to the learning objectives in our screencasts. There were 59 items in the post-test, 44 multiple choice, 7 short answer questions, 3 write a program and 5 matching type questions. Also, 22 of the 59 items were on factual knowledge, 31 on conceptual knowledge and 6 were on knowledge of process. One sample post-test question from each category is given in Table III: -

TABLE III. Sample post-test questions from each category

Checking knowledge of	Sample Question	
Fact	**Q16. Match the following**	
	Relational Operator	%
	Equality Operator	>=
	Unary Operator	=
	Arithmetic Operator	==
	Assignment Operator	++

Process	**Q3. What will be the output of following program?**
	`#include<stdio.h>`
	
	`int main()`
	`{`
	`    printf("One\tTwo");`
	`    return 0;`
	`}`
	a. One\tTwo
	b. One
	Two
	c. One Two
	d. None of the above

Concept	**Q8. Value of K if?** **K = 5 == 6** a. 6 b. 5 c. 0 d. 1 e. Other please write ....

C. Procedure

1) *Survey:* We first conducted the survey and then divided the students into 3 groups, based on the medium of their 12th Std, as shown in Table I. We compared the means of the 12th Std marks for the groups and found them to be equivalent. We did ANOVA to confirm the equivalence. We also conducted a pre-test after survey with the selected students. We removed all students who got more than 40% marks in pre-test because we wanted to include only those who had either no or little knowledge of programming.

2) *Arrangement:* We arranged separate computer lab for the three groups. In lab each computer was equipped with headphone and media player was installed on each computer in advance.

3) *Treatment:* Each student was alloted one computer. Each student watched screencast on the alloted computer in computer lab. Each computer was equipped with headphone so that students can not hear outside noise. Students were allowed to watch screencast for 45 continuous minutes. They were free to use video player controls according to their need. Shortest screencast is 22:00 minutes in length and longest is 45:00 minutes. There were no additional tutorials or laboratory exercises. The topics of the screencast of each day is listed in Table I

Screencasts were prepared with the slides and live coding [20]. We used digital pen so that teacher can draw diagrams, symbols and other necessary things in screencast. Fig. 1 shows an example of using digital pen with live-coding method. Slides were completely in English for all three groups. The explanation was in Hindi for HH group and English for EE and HE group. Screencasts in two languages were identical in terms of explanation, source-code, examples and analogy. Sample screenshots of the material are given in Fig. 1, Fig. 2 and Fig. 3. Note that only the vocal explanation is in Hindi for the HH group. A sample of the screencasts can be viewed at [21].

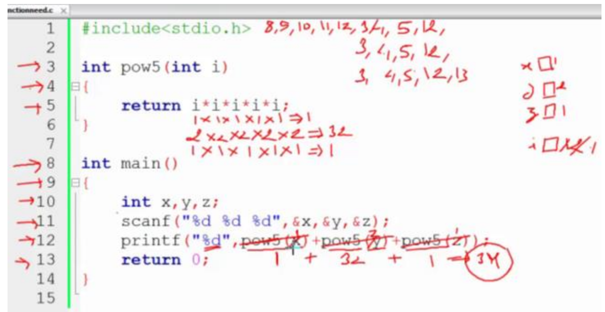

Fig. 1. A screenshot of screencast that shows use of digital pen-tablet and live-coding method.

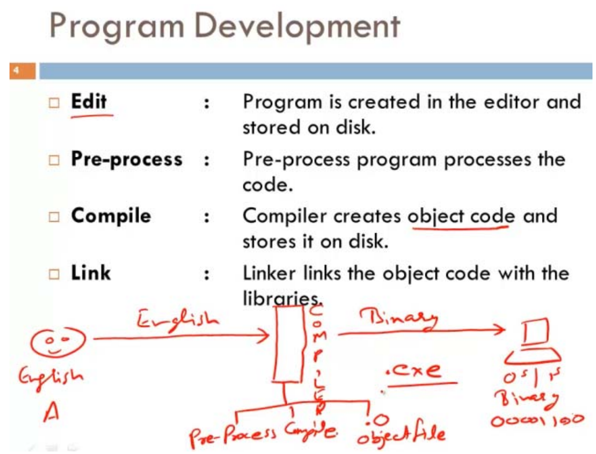

Fig. 2. Screenshot of screencast while explaining program development

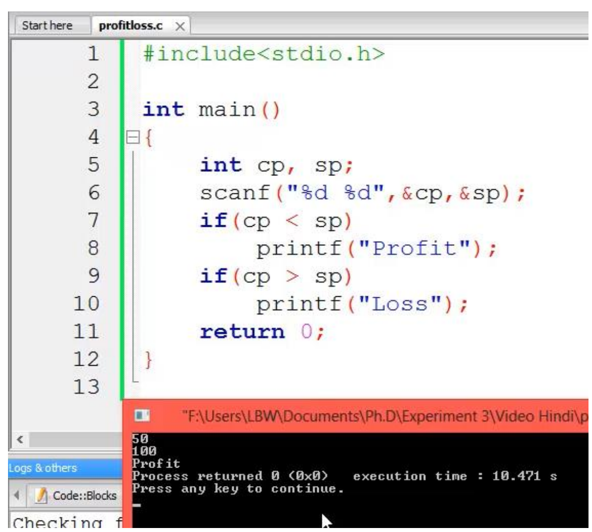

Fig. 3. Screenshot of screencast showing a source code with result

Screencast of each day for each group addressed the same Learning Objectives (LOs). The list of LOs is given below in Table IV.

TABLE IV. LEARNING OBJECTIVES OF SCREENCASTS

LO number	Learning Objective
LO1	Analyze and explain the behavior of simple programs involving the fundamental programming constructs covered by this unit.
LO2	Identify and describe uses of primitive data types.
LO3	Write programs that use each of the primitive data types.
LO4	Modify and expand short programs that use standard conditional structures and functions.
LO5	Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, and standard conditional, the definition of functions, and parameter passing.
LO6	Choose appropriate conditional constructs for a given programming task.
LO7	Describe the concept of recursion and give examples of its use.
LO8	Identify the base case and the general case of a recursively-defined problem.
LO9	Identify and describe the use of standard conditional structures and functions.

4) Posttest: To investigate the effect of the MoI on achievement scores, we conducted a post-test everyday after the treatment using the instrument we had designed earlier. Each student had to attempt the posttest individually, within a time limit of fifteen minutes. There was no negative marking.

5) *Analysis:* We performed quantitative analysis of the post-test scores for the different groups and question categories. We computed the means for each group. We used Welch t'test to know the stastistical significance between post-test scores of HH and EE groups. We used one-way ANOVA for HH and HE groups to determine which means are significantly different from one another.

V. RESULT ANALYSIS

The mean of post-test scores (out of 59) for the three groups (HH, HE and EE) are shown in Table V. From Table V, the mean of scores for HH and EE group is higher than HE group. HE group is the lowest performer while HH group outperform other two groups.

TABLE V. MEAN OF POST-TEST SCORE IN EACH GROUP

Group	N	Mean	Std. Deviation	Std. Error of Mean
HH	35	45.00	7.472	1.263
HE	35	37.57	5.937	1.004
EE	35	42.51	5.511	.932

The distribution of percentage of post-test scores for all three categories of questions (fact, process and concept) of the three groups (HH, HE and EE) is presented in Table VI and graphically represented in Fig. 4. From Table VI and Fig. 4 we can say that for all categories of questions-

- HH group outperformed other two groups (HE and EE). This difference was statistically significant for HH vs HE, while it was not so for HH vs EE.
- HE group is the lowest performer among all groups.

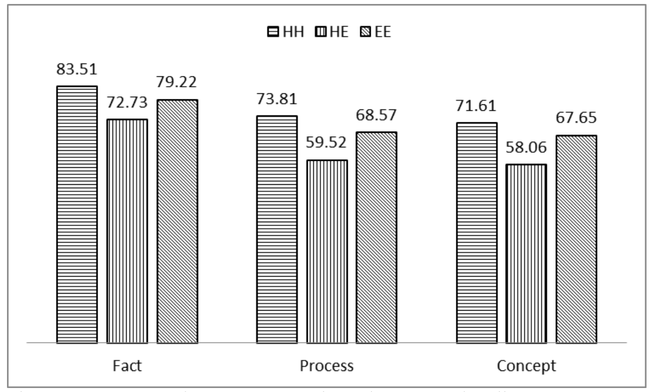

Fig. 4. Percentage of post-test scores in each category for all groups

TABLE VI. PERCENTAGE OF POST-TEST SCORES IN EACH CATEGORY FOR ALL GROUPS

Group	Fact	Process	Concept
HH	83.51	73.81	71.61
HE	72.73	59.52	58.06
EE	79.22	68.57	67.65

A. Comparison of HH and EE groups

We compare HH and EE groups as we expect that there will be no significant difference in post-test scores in total score as well as in each category of questions. We performed one-way ANOVA [23] and found no significant difference in the post-test scores for any question category, as shown in Table VII.

TABLE VII. ONE-WAY ANOVA FOR HH AND EE GROUPS

		Sum of Squares	df	Mean Square	F	Sig.
Fact	Between Groups	15.557	1	15.557	2.088	.153
Process	Between Groups	1.729	1	1.729	1.012	.318
Concept	Between Groups	26.414	1	26.414	1.586	.212
Total	Between Groups	108.129	1	108.129	2.509	.118

B. Comparison of HH and HE groups

We compare HH and HE groups as we expect that there will be significant difference in post-test scores in total score as well as in each category of question. We performed one-way ANOVA [23] and found significant different in total score as well as in fact, process and concept question categories, as shown in Table VIII.

TABLE VIII. ONE-WAY ANOVA FOR HH AND HE GROUPS

		Sum of Squares	Df	Mean Square	F	Sig.
Fact	Between Groups	98.41	1	98.41	12.62	.001
Process	Between Groups	12.86	1	12.86	7.10	.010
Concept	Between Groups	308.70	1	308.70	21.88	.000
Total	Between Groups	965.72	1	965.71	21.21	.000

VI. Discussion and Conclusion

Our treatment of teaching programming using screencast in English versus Hindi, shows significant difference (p-value = 0.000) between total post-test scores of HH, HE and EE group. It shows that MoI plays an important role in teaching programming.

We compared HH and EE groups and found no significant difference. This is as expected since MoI for these groups was same as medium of their K-12. This also confirms that the screencasts were identical in everything except language of vocal explanation.

We compared HH and HE groups and found significant difference in total post-test scores as well as fact, process and concept type questions. Facts were written in slides in English and teacher explained them in MoI of the respective group. In order to understand the fact students have to either read the text from screencast and translate it into their native language or listen to teacher carefully to make sense of the information. For both groups, the slide text was in English and hence both groups had to read-translate-understand the English sentences. However, since the explanation was in English for HE group they had greater difficulty in comprehension, as seen from the scores.

We explained process content type by live-coding [20] method and annotating source-code using digital pen-tablet [22] with audio narration. Thus understanding process was more dependent on visual elements and less dependent on vocal explanations. Since students were able to observe the program writing process and the result of program execution. We were expecting that we will not see any significant difference in post-test scores for process category of questions as the dependency on the MoI will reduce but we found no significant difference for process category, as shown in Table VIII.

Similarly, we found significant difference in scores of concept type. Learning of a concept involves learning the definition, identifying instances of the concept in programming problems, and discrimination with other concepts [11]. The impact of MoI on learning of concepts was expected, and seen.

Based on our findings, we recommend that while making screencasts to teach programming to native language learners-

- Written content (slides, source-code) should be in English.
- Vocal explanation should be in native language for fact, process and concept content types.
- Live coding and on-screen annotation supports learning and should be in English.

References

[1] SK Mitra (2010). "Internationalization of Education in India: Emerging Trends and Strategie." In the Journal of Asian Social Science, June 2010, pp. 105-110.

[2] Prabhakar Kaushik and Dinesh Khanduja (2010). "Utilising six sigma for improving pass percentage of students: A technical institute case study." In the journal of Educational Research and Review, September 2010, pp. 471-483.

[3] Jens Bennedsen and Michael E. Caspersen (2007). "Failure Rates in Introductory Programming." In SIGCSE bulletin, June 2007, pp. 32-36.

[4] Leo Gomez, David Freeman and Yvonne Freeman. Dual Language Education: A promising 50-50 model. Bilingual Research Journal, volume 29, Issue 1, 2005. pages 145-164

[5] Jinshu Han (2010). Bilingual Teaching Practice of Undergraduate Computer Graphics. Second International Workshop on Educational Technology and Computer Science. Pages 441-444.

[6] Shuang Liu, Xizuo Li and Li Zuo (2011). Adopting communicative teaching method in computer major bilingual teaching. Theory and Practice in language studies, vol 1, no 2, pp. 187-190, February 2011

[7] Hanjing, L., W. Kuanquan and W. Yuying (2009). A Bilingual Teaching Modal in a Programing Language Course. Education Technology and Training, 2009. ETT '09. Second International Conference on.

[8] Pal, Y. and S. Iyer (2012). Comparison of English versus Hindi Medium Students for Programming Abilities Acquired through Video-Based Instruction. Proceedings of the 2012 IEEE Fourth International Conference on Technology for Education, IEEE Computer Society: 26-30.

[9] Oswald, D., Wild, G., Hinckley, S., (2012), Developing an Educational Video for Utilising an Oscilloscope in First Year Physics. Proceedings of the 20th Australian Institute of Physics Congress, 1-4, Australia.

[10] Joint Task Force on Computing Curricula, A. f. C. M. and I. C. Society (2013). Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, ACM.

[11] Clark, R.C. and Mayer, R.E. (2002). E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning. San Francisco: Jossey-Bass Pfeiffer.

[12] Dong-Joong Kim, Joan Ferrini-Mundy, Anna Sfard (2012), How does language impact the learning of mathematics? Comparison of English and Korean speaking university students' discourses on infinity", International Journal of Educational Research. Feb. 2012.

[13] D. Venkatesan, Dr. RM Chandrasekaran and Dr. A. Velangani Joseph (2010), "Significance of Medium of Instruction and Personality in Designing Adaptive e-Learning Systems.", In the International Journal of Computer Applications, Oct. 2010, pp. 25-28.

[14] Kathleen Barnes, John Bowers, Julie H. Dent (2009), "Video Tutorials: A Sustainable Method for Campus Technology Training", In EDUCASE Quarterly (EQ), 2009 Vol. 32, No. 3.

[15] Ellis, C. (2008). 'You Can't Do That in a Classroom!': How Distributed Learning Can Assist in the Widespread Adoption of Hybrid Learning Strategies. Hybrid Learning and Education. J. Fong, R. Kwan and F. Wang, Springer Berlin Heidelberg. 5169: 1-16.

[16] Simpson, N. (2006). "Asynchronous access to conventional course delivery: a pilot project." British Journal of Educational Technology 37(4): 527-537.

[17] Lloyd, S. A. and C. L. Robertson (2012). "Screencast Tutorials Enhance Student Learning of Statistics." Teaching of Psychology 39(1): 67-71.

[18] J. Ravi, H.J. Jani (2011), "A Critical Study of NPTEL", In International conference on Technology for Education (T4E), Aug. 2011, pp. 35-42

[19] Allison Elliott Tew and Mark Guzdial (2011). The FCS1: A Language Independent Assessment of CS1 Knowledge." SIGCSE, Mar. 2011, pp. 111-116.

[20] Gaspar, A. and S. Langevin (2007). Restoring "coding with intention" in introductory programming courses. Proceedings of the 8th ACM SIGITE conference on Information technology education. Destin, Florida, USA, ACM: 91-98.

[21] https://www.youtube.com/watch?v=m4zqbaiUj4sM as seen on 28 nov. 2012

[22] Mock, K. (2004). "Teaching with Tablet PC's." J. Comput. Sci. Coll. 20(2): 17-27.

[23] Glass, G. V. and K. D. Hopkins (1996). Statistical methods in education and psychology (3rd ed.). Needham Heights, MA, US: Allyn & Bacon. pp. 295, 387-388