

Affective Learning with Online Software Tutors for Programming

Amruth N. Kumar

Computer Science
Ramapo College of New Jersey
amruth@ramapo.edu

Keywords: POP-I.A Learning to program, POP-II.A Individual differences, POP-II.B. Program comprehension, POP-V.B Cross-sectional studies, POP-VI.E. Computer science education research

Abstract

We conducted a study to see if there was any difference in the affective learning of students who used software tutors to learn programming concepts. We used two software tutors – on arithmetic expression evaluation and tracing selection statements for this study. Data was collected over five semesters with the arithmetic tutor and over four semesters with the selection tutor, yielding a sample size in the thousands. The data was analysed using ANOVA, with sex (male versus female), representation (Caucasians and Asians versus other races), and discipline (Computer Science versus other disciplines) as fixed factors. We found no difference in the affective learning of any demographic group - all the groups felt in equal measure that they had learned using the software tutors. This is a positive result since affective learning complements cognitive learning. But, we did find statistically significant difference between the sexes and between the representation groups on prior-preparedness on arithmetic tutor, but not selection tutor. This may be because arithmetic tutor covered concepts that students had been exposed to in high school, but the concepts covered by selection tutor were unique to programming and unlikely to have been seen by students who had had no prior programming experience. Finally, non-Computer Science majors learned significantly more concepts than Computer Science majors on both the tutors, even though no significant difference was found between the two disciplinary groups on prior-preparedness. This may allude to difference in perceived self-efficacy of the two groups towards learning from software tutors.

1. Motivation

Essentialism permeates the discussion of participation of women in computing. Essentialism posits that people have properties essential to their composition, and that all members of a demographic group share common characteristics (Frieze & Quesenberry, 2013). Gender essentialism has been supported by studies that suggest that girls are concerned about the passivity of their interactions with the computer as a (learning) tool, and that they dislike *narrowly and technically focused* programming classes ((AAUW, 2000). Whereas men view computers as machines, women view them as tools for use within a *societal and/or interdisciplinary context* (Margolis & Fisher, 2002). In response, National Center for Women & Information Technology (NCWIT) recommends that one strategy for retaining undergraduates (women) in computing is to align assignments and coursework with student interests and career goals (Barker & Cohoon, 2009).

All along, there have also been arguments against using essentialism to explain the differences in the attitudes of men and women towards computing, and therefore, the difference in their participation levels in the discipline. Klawe warned about the widespread *misconceptions* (Klawe, 2002) that “girls don’t like using computers” and that “girls are people people and computer people are not people people”. Explaining the differences based on culture rather than essentialism has been recently proposed as a more effective way of increasing the participation of women in computing (Frieze & Quesenberry, 2013).

We have been developing and deploying software tutors called problets (problets.org) that help students learn programming concepts by solving drill-and-practice problems. The problems presented by these tutors are structured around individual programming constructs, and therefore, are *narrowly and technically focused*. Since the problems are automatically generated as instantiations of templates, they lack any *societal or interdisciplinary context* recommended in essentialist proposals for increasing the participation of women in computing, e.g., (Margolis & Fisher, 2002). Therefore, we have been especially interested in the differential effects, if any, of our software tutors on women versus men vis-à-vis cognitive and affective learning. However, users of our software tutors are asked to voluntarily identify their sex (biological notion of male/female) rather than their gender (social/cultural notion of man/woman) (Sears, 1999). Therefore, henceforth, we will be referring to male versus female students rather than men versus women.

In Computer Science, in addition to Caucasians who are traditionally represented, Asians have also been positively stereotyped because of its quantitative nature (Kao, 1995). The other racial groups, viz., Black/African American, Hispanic/Latino, Native American, Native Hawaiian/Pacific Islander and Other, are traditionally classified as under-represented. *Myths* of genetic determinism have been considered to be one of the barriers for participation of under-represented groups in computing (Eglash, Gilbert, & Foster, 2013). Stereotype threat (Steele & Aronson, 1995) has also been listed as one of the factors that could be contributing to problems with recruitment and retention of female and minority students (Beyer, Rynes, Perrault, Hay, & Haller, 2003; Peckham et al., 2007). Here again, culture-based approaches to computing education have been proposed as a way to increase participation of under-represented groups in computing education (Eglash et al., 2013).

Drill-and-practice problem-solving as supported by our software tutors is devoid of any cultural context. So, we have also been interested in differential effects, if any of our tutors on the cognitive and affective learning of traditionally-represented versus under-represented racial groups.

1.1. Prior Work with our Software Tutors

During earlier studies, we had found that female students learned just as well as male students using our software tutors (Kumar, 2007). They rated the usability, ability to learn from, and usefulness of the tutors more favourably than male students (Kumar, 2006a) and this was not an artifact of the design of the survey instrument (Kumar, 2008b). They had lower prior self-confidence (Kumar, 2011), but using the software improved their self-confidence to be on par with that of male students (Kumar, 2006b, 2008a).

We had found that students from under-represented groups rated the tutors more favourably than those from traditionally represented groups (Kumar, 2009b). They had less prior preparation and lower prior self-confidence than students from traditionally represented racial groups (Kumar, 2011). However, students needed and benefited from the tutors in the same proportion, regardless of sex or racial group (Kumar & Kaczmarczyk, 2013).

One study of under-represented groups in computing found intersectionality, i.e., interaction between gender and race to be critical, i.e., under-represented males may have less in common with under-represented females than say, Caucasian females (Trauth, Cain, Joshi, Kvasny, & Booth, 2012). This concurs with our own observations that when evaluating educational interventions in computing, significant interactions exist among demographic groups (Kumar, 2009a).

Given these prior results on self-confidence, assessment of tutor, and to a lesser extent, cognitive learning and prior preparation, we focused on affective learning in this study, but included supplemental investigation of prior-preparation (how well the students were prepared before using our software tutors) and number of concepts learned (cognitive learning) using our software tutors.

1.2. Affective Learning

Affective learning relates to emotional aspects of learning, such as:

- Motivation, which mediates learning by increasing or decreasing cognitive engagement (Gottfried, 1990);

- Attitude towards learning – positive or negative; and
- Emotions such as anxiety, confidence, and boredom, which affect learning (e.g., (Kort, Reilly, & Picard, 2001)).

Researchers are increasingly acknowledging the importance of affective factors in computer-mediated learning (Picard et al., 2004), and are trying to address affective issues in the design of software tutors (e.g., (Baker, D'Mello, Rodrigo, & Graesser, 2010)).

Self-efficacy is one's judgment about what one can or cannot do (Bandura, 1977). It affects one's motivation, influences one emotionally, and is one of the more easily measurable aspects of affective learning. In our work, we focused on self-efficacy, i.e., learners' judgment of what they could or could not do with their knowledge after a cognitive learning session with our tutor.

2. Study Protocols and Instruments

For this study, we used two tutors – on arithmetic expression evaluation and tracing programs containing selection statements. These two tutors were selected because they promised large sample sizes – they are both typically used early in the semester and are used by larger numbers of students than tutors on more advanced topics such as functions and arrays. At the same time, the two tutors are not duplicative – they have different user interfaces, cover different programming concepts and require different problem-solving skills.

Each tutor was configured to administer pretest-practice-post-test protocol as follows:

- **Pretest** – During pretest, the tutor presented one problem per concept. If a student solved a problem correctly, the student was given credit for the corresponding concept. No feedback was provided to the student, and no more problems on the concept were presented to the student. On the other hand, if the student solved a problem incorrectly, feedback was presented to the student immediately after the student submitted his/her solution to the problem. Additional problems were presented on the concept during the subsequent stages.
- **Adaptive practice** – During this stage, additional problems were presented to the student on only the concepts on which the student made mistakes when solving problems during the pretest. For each such concept, the student was presented multiple problems until the student mastered the concept, i.e., solved at least 60% of the problems correctly. On each problem, the student received feedback explaining the correct answer step by step.
- **Post-test** - During this stage, the student was presented test problems on the concepts that the student had mastered during adaptive practice.
- **Affective Learning Survey:** During this stage, the student was asked to respond to 5 - 8 statements on affective learning, using a 5-point Likert-scale questionnaire.
- **Demographics** - Students were provided the option to identify their demographic information, including sex and race. Demographic information was solicited after the pretest-practice-post-test protocol to avoid the effects of stereotype threat (Kumar, 2012).

The entire protocol was administered online, back-to-back, with no break in between, all by the software tutor. The pretest-practice-post-test problem-solving session was limited to 30 minutes.

If a student solved all the problems correctly during pretest, the student was not presented any practice or post-test. If the student did not meet the minimum percentage correctness on a concept during practice, regardless of how many problems the student had solved on the concept during practice, the student was not presented any post-test problems on that concept. If a student did not get to practice or post-test stage, it could also have been because the student ran out of time. The concepts on which a student solved the problem incorrectly during pretest, demonstrated mastery during practice and solved a post-test problem are *practiced concepts*. Each practiced concept on which a pre-post increase in score was observed is also a *learned concept*.

2.1. Arithmetic Expression Evaluation Tutor

Tutor on Arithmetic Expression evaluation covered 25 concepts such as correct evaluation, precedence and associativity of the five arithmetic operators, divide-by-zero error, use of parentheses and coercion of data types. Students were presented with an expression involving one or more arithmetic operators, and were asked to evaluate it one operator at a time, i.e., pick the operator and the operands to which it applied, and enter the intermediate result of its evaluation (See Figure 1).

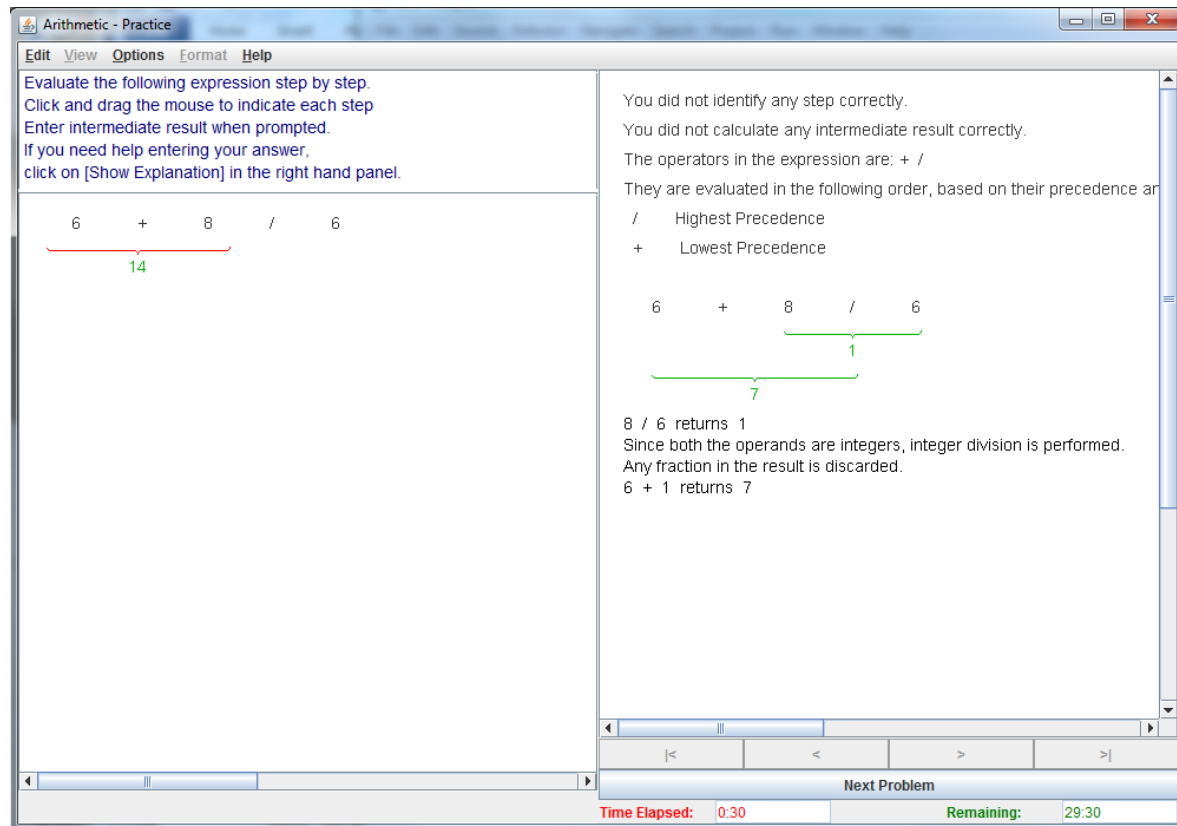


Figure 1: Snapshot of Arithmetic Tutor

The pretest contained 16 problems. The problems were selected to together cover all 25 concepts at least once. For example, the problem in the left panel of Figure 1 covers 4 concepts - correct evaluation of addition and integer division operators, and their relative precedences.

The affective survey administered with the arithmetic tutor consisted of the following 5 statements:

After using the tutor, I can do the following better:

1. Read arithmetic expressions
2. Evaluate arithmetic expressions
3. Find errors in arithmetic expressions
4. Write arithmetic expressions
5. Critique arithmetic expressions

These statements were picked to reflect the levels of revised Bloom's Taxonomy (Anderson & Krathwohl, 2001) – applying (2), analysing (1, 3), evaluating (5) and creating (4). Students responded on a 5-point Likert scale of Strongly Agree (coded as 1), Agree (2), Neutral (3), Disagree (4) and Strongly Disagree (5). Students had the option to skip the survey altogether or respond to only some of the statements.

2.2. Selection Statement Tutor

Tutor on selection statements covered 12 concepts such as one-way and two-way selection statements, nested selection statements, execution of the statement when the condition is true/false, etc. Students were presented a program containing selection statement(s) and were asked to identify the output of the program one at a time, along with the line number of the code that produced each output (See Figure 2).

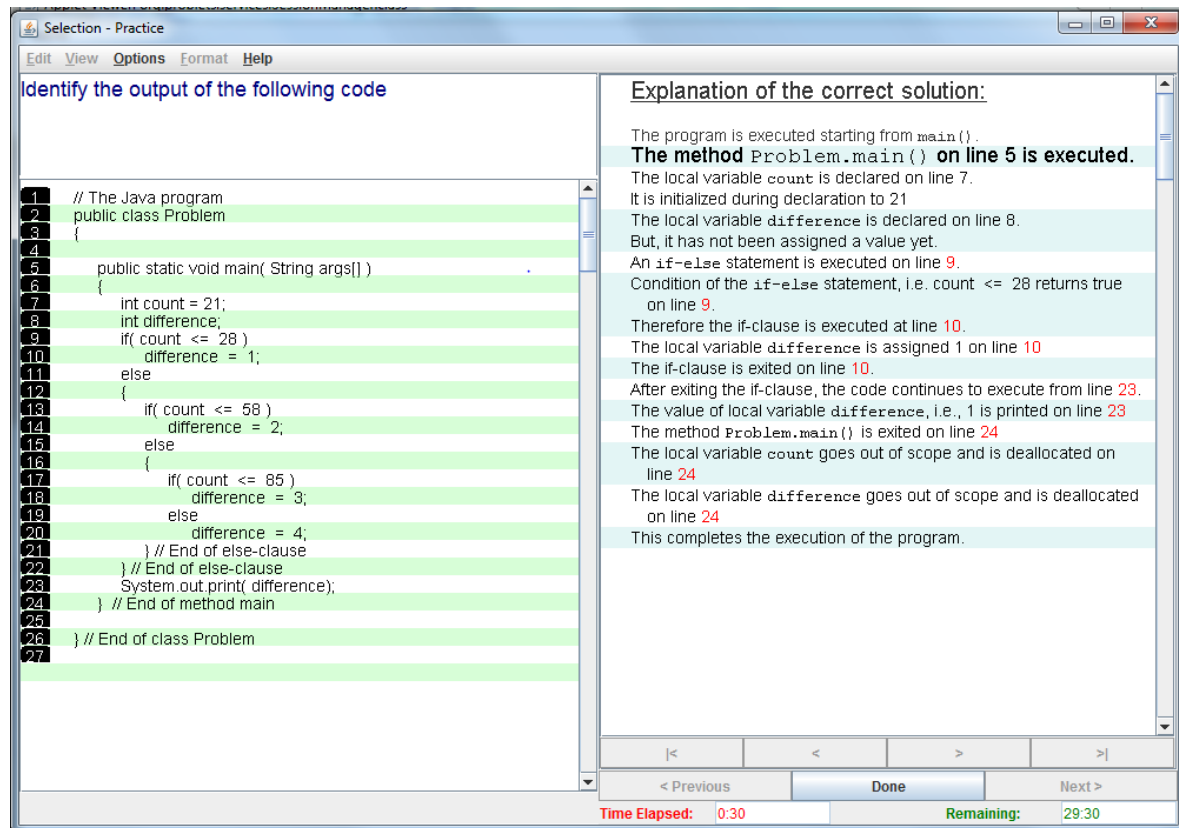


Figure 2: Snapshot of Selection Tutor

The pretest contained 12 problems, one for each of the 12 concepts covered by the tutor. For example, the program in the left panel of Figure 2 covers the concept of nested `if-else` statements.

The affective survey administered with selection tutor consisted of the following 8 statements:

After using the tutor, I can do the following better:

1. Understand the grammar rules of `if-else` statements
2. Understand the meaning of `if-else` statements
3. Read `if-else` statements
4. Predict the output of `if-else` statements
5. Debug `if-else` statements
6. Write `if-else` statements
7. Design `if-else` statements
8. Critique `if-else` statements

These statements were also picked to reflect the levels of revised Bloom's Taxonomy – understanding (1, 2), applying (4), analysing (3, 5), evaluating (8) and creating (6, 7). Once again, students

responded on a 5-point Likert scale and had the option to skip the survey altogether or respond to only some of the statements.

3. Data Collection and Analysis

Data Collection: The two tutors were used by students of instructors at several institutions each semester. Typically, these were students in introductory programming courses. Students typically used the tutors after class, on their own time, over the web, without instructor supervision. Data was collected over multiple semesters when the versions of the two tutors were kept unchanged.

Data Filtering: Students could use each tutor as many times as they wished. When they used a tutor more than once, only the first attempt when the student had solved most of the pretest problems was considered for analysis. Only those students were considered for analysis who had solved at least a minimum number of pretest problems.

Data Analysis: The following data was analysed for this study:

1. **Prior preparation:** The score per problem on the **pretest**, which is indicative of prior-preparation of the student, i.e., how well prepared the student was before using the tutor. Score per problem was used rather than the raw score in order to eliminate the effect of difference in the number of problems solved by students during pretest. The range of values for score per problem was 0 → 1.0.
2. **Learned concepts:** The number of learned concepts, i.e., the concepts on which the student solved the pretest problem incorrectly, solved sufficient practice problems to demonstrate mastery and solved a post-test problem correctly.
3. **Affective learning:** The aggregate Likert-scale score on the affective survey statements presented after using each tutor – 5 statements for Arithmetic tutor and 8 statements for Selection tutor. So, the range of possible values was 5 → 25 for Arithmetic tutor and 8 → 40 for Selection tutor.

The following demographic groupings were used for analysis:

1. **Sex:** male or female as self-identified by the student;
2. **Representation:** Based on the race self-identified by the students, Caucasians and Asians were combined into traditionally represented group and all other racial groups were combined into under-represented group;
3. **Discipline:** Based on the major self-identified by the students, Computer Science majors were compared against students from all other majors, which included Information Systems, Engineering, Basic Sciences, Business, Arts, Humanities, Social Sciences, and Other.

Since providing demographic information was optional, not all the students entered all their information. Since the tutors were adaptive and timed, not all the students got credit for learned concepts – some knew all the concepts before using the tutor and some ran out of time before being able to practice any concept all the way from pretest to post-test. Since filling out the affective learning questionnaire was not mandatory, not all the students filled it out. Therefore, N value varies with each analysis and has been individually noted in the ANOVA results in the next section.

3.1. Results of Analysis – Arithmetic Tutor

Data was collected over five semesters - Fall 2011 through Fall 2013. A controlled study of the effects of visualization was being conducted during these semesters. While the hypothesis of the controlled study is not relevant to the current study, treatment was taken as one of the factors during analysis of the collected data. Anyone who had solved fewer than 10 of the 16 pretest problems was dropped from analysis. After this, 1065 students remained in the control group and 1304 students in the test group, from 56 different institutions.

The pre-post change in score on learned concepts was as follows, all pre-post changes being statistically significant ($p \leq 0.05$):

Arithmetic Tutor		Pretest	Post-test
Control (N=415)	Mean	0.05 ± 0.0185	0.9228 ± 0.0185
	Std-Deviation	0.1926	0.1923
Test (N=485)	Mean	0.0424 ± 0.0139	0.9349 ± 0.0155
	Std-Deviation	0.1557	0.1746

We conducted a 2 x 2 x 2 x 2 ANOVA analysis, with sex (male versus female), representation (Caucasians and Asians versus other races), discipline (Computer Science versus other majors) and treatment (without versus with visualization) as fixed factors.

On affective learning (sum of the five Likert-scale responses), we found no significant main effect for sex, representation or discipline. When we eliminated students who had scored greater than 95% on the pretest, we found a marginal main effect for representation [$F(1,971) = 3.54$, $p = 0.06$]: Caucasians and Asians reported greater affective learning (9.353 ± 0.374 – confidence intervals being at 95% confidence level) than other racial groups (10.049 ± 0.622).

On the pretest average score per problem, we found:

- Significant main effect for sex [$F(1,1729) = 5.26$, $p = 0.022$]: male students scored higher on the pretest (0.882 ± 0.09 points) compared to female students (0.859 ± 0.18 points). However, once all the students who had scored more than 95% on the pretest were eliminated, the difference was no longer significant. In other words, more male students scored greater than 95% on the pretest than female students.
- Significant main effect for representation [$F(1,1729) = 17.637$, $p < 0.001$]: Caucasians and Asians scored higher on the pretest (0.891 ± 0.1 points) than students from under-represented groups ($0.849 \pm .18$ points).
- No significant main effect for discipline, and no significant interaction among the factors

In other words, male students and traditionally represented students were better prepared than their counterparts before using the software tutors. On the number of concepts learned we found:

- Significant main effect for discipline [$F(1,665) = 5.611$, $p = 0.018$]: non-Computer Science students learned more concepts (1.794 ± 0.157) than Computer Science majors (1.497 ± 0.189)
- No significant main effect for sex or representation

3.2. Results of Analysis – Selection Tutor

Data was collected over four semesters – Fall 2010 through Spring 2012. All the students got exactly the same treatment while using the tutor. Anyone who had solved fewer than 6 of the 12 pretest problems was dropped. After this, 908 students remained in the study from 37 different institutions.

The pre-post change in score on learned concepts was as follows, all pre-post changes being statistically significant ($p \leq 0.05$):

Selection Tutor		Pretest	Post-test
(N=438)	Mean	0.0413 ± 0.0080	0.9156 ± 0.0127
	Std-Deviation	0.1226	0.1949

We conducted a 2 x 2 x 2 ANOVA analysis, with sex (male versus female), representation (Caucasians and Asians versus other races) and discipline (Computer Science versus other majors) as fixed factors:

- On affective learning (sum of the 8 Likert-scale responses) and average pretest score, we found no significant main effect for sex, representation or discipline.
- On the concepts learned, the only significant main effect we found was for discipline [$F(1,310) = 4.482, p = 0.035$]: once again, non-Computer Science students learned more concepts (1.95 ± 0.21) than Computer Science majors (1.488 ± 0.375).

3.3. Discussion of Results

Whereas arithmetic tutor covers concepts that students would have been exposed to in high school, selection tutor covers concepts that are unique to programming and are entirely new to anyone who has never programmed before. This might explain why we found male and traditionally represented students to be better prepared than their counterparts on arithmetic tutor concepts, but not on selection tutor concepts.

On both the tutors, non-Computer Science majors learned more concepts than Computer Science majors. This was true even though no significant difference was found between the two disciplinary groups on prior-preparedness (pretest average score) on either tutor. So, the greater learning of non-Computer Science majors cannot be attributed to greater need. Instead, it may allude to difference in perceived self-efficacy of the two groups towards learning from software tutors. We plan to explore this in the future.

Regardless of the differences found between groups on prior-preparation and concepts learned, no difference was found between demographic groups on affective learning on either tutor. In other words, all the demographic groups felt in equal measure that they had learned using the software tutors, which in itself is encouraging, since affective learning complements cognitive learning (Hurd, 2008). In the future, we plan to repeat this study with tutors for higher level concepts such as loops and functions.

One confounding factor of this study is that the survey instruments used for affective learning were not validated. The survey instruments were designed to elicit students' self-efficacy – their judgement about what they could or could not do on the topic of the tutor, in terms of revised Bloom's taxonomy. Self-efficacy associated with the use of computers is called Computer Self-efficacy (Compeau & Higgins, 1995). Numerous factors have been identified as contributing to computer self-efficacy, and are classified into 12 categories: enactive mastery, task characteristics, perceived effort, situation support, degree/quality of feedback, emotional arousal, vicarious experience, verbal persuasions, assigned goals/anchors, degree of professional orientation, age and attribution of cause (Marakas, Yi, & Johnson, 1998). The instruments used in this study deal only with enactive mastery, i.e., gaining relevant experience with the problem-solving task.

On the other hand, a supporting factor of the current study is that it was conducted with students using the tutors on their own time, unsupervised, i.e., *in-natura*. This reduced the likelihood of Hawthorne effect (Franke & Kaul, 1978) affecting the results. That these students chose to answer the survey on affective learning on their own time after solving problems with the tutor for up to 30 minutes, even though answering the survey was optional, adds credence to the results.

4. Acknowledgements

Partial support for this work was provided by the National Science Foundation under grant DUE-0817187.

5. References

- AAUW. (2000). *Tech-Savvy: Educating Girls in the New Computer Age*: American Association of University Women.
- Anderson, L. W., & Krathwohl, D. R. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Longman.
- Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68, 223-241.
- Bandura, A. (1977). Self-Efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, 8(2), 191-215.
- Barker, L., & Cohoon, J. M. (2009). Key Practices for Retaining Undergraduates in Computing National Center for Women and Information Technology.
- Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003). *Gender differences in computer science students*. Paper presented at the 34th SIGCSE Technical Symposium.
- Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2), 189-211.
- Eglash, R., Gilbert, J. E., & Foster, E. (2013, July). Broadening Participation - Toward Culturally Responsive Computing Education. *Communications of the ACM*, 56, 33-36.
- Franke, R. H., & Kaul, J. D. (1978). The Hawthorne experiments: First statistical interpretation. *American Sociological Review* 43, 623-643.
- Frieze, C., & Quesenberry, J. L. (2013). *From difference to diversity: including women in the changing face of computing*. Paper presented at the Proceeding of the 44th ACM technical symposium on Computer science education, Denver, Colorado, USA.
- Gottfried, A. (1990). Academic intrinsic motivation in young elementary school children. *Journal of Educational Psychology*, 82, 525-538.
- Hurd, S. (2008). Affect and strategy use in independent language learning. In S. Hurd & T. Lewis (Eds.), *Language learning strategies in independent settings* (pp. 218-236). Bristol: Multilingual Matters.
- Kao, G. (1995). Asian Americans as model minorities? A look at their academic performance. *American Journal of Education*, 103, 121-159.
- Klawe, M. (2002, June). Girls, Boys and Computers. *SIGCSE Bulletin Special Issue on Women and Computing*, 34, 16-17.
- Kort, N., Reilly, R., & Picard, R. (2001). *An affective model of interplay between emotion and leaning: reengineering educational pedagogy - building a learning companion*. Paper presented at the IEEE International Conference on Advaned Learning Technology, Madison, WI.
- Kumar, A. N. (2006a). *Do female students feel differently than male students about using software tutors?* . Paper presented at the Frontiers in Education Conference (FIE 2006), San Diego, CA. <http://pages.ramapo.edu/~amruth/r/c/fie/06/paper.pdf>
- Kumar, A. N. (2006b). *The Effect of Using Problem-Solving Tutors on the Self-Confidence of Students*. Paper presented at the 18th Annual Psychology of Programming Workshop (PPIG 06), Brighton, U.K. <http://pages.ramapo.edu/~amruth/r/wkshp/ppig/06/paper.pdf>
- Kumar, A. N. (2007). *Software Tutors Help Female Students Learn Programming Concepts Just as Well as Male Students*. Paper presented at the E-LEARN 2007, Quebec City, Canada. <http://pages.ramapo.edu/~amruth/r/c/elearn/07/paper.pdf>

- Kumar, A. N. (2008a). *The Effect of Using Problem-Solving Software Tutors on the Self-Confidence of Female Students*. Paper presented at the 39th SIGCSE Technical Symposium, Portland, OR. <http://pages.ramapo.edu/~amruth/r/c/sigcse/08/paper.pdf>
- Kumar, A. N. (2008b). *Female Students Assess Software Tutors More Positively Than Male Students*. Paper presented at the Frontiers in Education Conference (FIE 2008), Saratoga Springs, NY. <http://pages.ramapo.edu/~amruth/r/c/fie/08/paper.pdf>
- Kumar, A. N. (2009a). *Need to Consider Variations within Demographic Groups When Evaluating Educational Interventions*. Paper presented at the Innovation and Technology in Computer Science Education, Paris, France. <http://pages.ramapo.edu/~amruth/r/c/iticse/09/paper.pdf>
- Kumar, A. N. (2009b). *Patterns in Student Assessment of Problem-Solving Software*. Paper presented at the Frontiers in Education Conference, San Antonio, TX. <http://pages.ramapo.edu/~amruth/r/c/fie/09/paper.pdf>
- Kumar, A. N. (2011). *Results from Repeated Evaluation of an Online Tutor on Introductory Computer Science*. Rapid City, SD.
- Kumar, A. N. (2012). *A Study of Stereotype Threat in Computer Science*. Haifa, Israel.
- Kumar, A. N., & Kaczmarczyk, L. C. (2013). *Programming Tutors, Practiced Concepts, and Demographics*. Oklahoma City, OK.
- Marakas, G. M., Yi, M. Y., & Johnson, R. D. (1998). The multilevel and multifaceted character of computer self-efficacy: Toward clarification of the construct and an integrative framework for research. *Information Systems Research*, 9(2), 126-163.
- Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: women in computing*: MIT Press.
- Peckham, J., Harlow, L. L., Stuart, D. A., Silver, B., Mederer, H., & Stephenson, P. D. (2007). *Broadening participation in computing: issues and challenges*. Paper presented at the Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education, Dundee, Scotland. <http://dl.acm.org/citation.cfm?doid=1268784.1268790>
- Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., . . . Strohecker, C. (2004). Affective learning - a manifesto. *BT Technology Journal*, 22(4), 253-269.
- Sears, J. (1999). Teaching queerly: Some elementary propositions. . In W. J. Letts & J. Sears (Eds.), *Queering elementary education: Advancing the dialogue about sexualities and Schooling* (pp. 97-110). Lanham, MD: Rowman & Littlefield, Inc.
- Steele, C. M., & Aronson, J. (1995). Stereotype threat and the intellectual test performance of African Americans. *Journal of Personality and Social Psychology*, 69, 797-811.
- Trauth, E. M., Cain, C. C., Joshi, K. D., Kvasny, L., & Booth, K. (2012). *Embracing intersectionality in gender and IT career choice research*. Paper presented at the Proceedings of the 50th annual conference on Computers and People Research, Milwaukee, Wisconsin, USA. <http://dl.acm.org/citation.cfm?doid=2214091.2214141>