Promoting Reflection and its Effect on Learning in a Programming Tutor

Amruth N. Kumar

Ramapo College of New Jersey
505 Ramapo Valley Road
Mahwah, NJ 07430
amruth@ramapo.edu

Abstract

We studied the effect of post-practice reflection on learning, using programming tutors, and multiple-choice format for reflection. We conducted *in-vivo* controlled studies with introductory programming students from multiple schools over 3 semesters, and used mixed-factor ANOVA to analyze the collected data. We found that reflecting on the concept underlying each problem neither promotes greater learning, measured as pre-post increase in the average score per problem, nor promotes faster learning, measured as the problems solved per concept learned. We conjecture that the benefits of reflecting on the concept underlying each problem may be limited if a tutor already promotes deep understanding of the domain.

Problets

We have been developing software tutors, called problets (www.problets.org) to help students learn C/C++/Java/C# programming language concepts by solving problems. To date, we have developed, evaluated and deployed problets on expression evaluation (arithmetic, relational, logical, assignment), selection statements, loops (while and for) and C++ pointers. The problets present programs to the learner, ask the learner a question about the program, such as predicting its output or identifying bugs in it, grade the student's answer, and provide delayed feedback. Figure 1 shows a snapshot of a problet on selection statements, with the program in the left panel and the feedback in the right panel.

Problets generate problems as instances of parameterized problem templates. Each template is associated with a concept in the domain, e.g., some selection statement concepts include executing a selection statement when the condition is true/false, executing nested if statements, executing if-else statements nested in cascaded/classification style, and executing a program with multiple dependent/independent selection statements. Similarly, some loop concepts include nested dependent and independent loops, multiple dependent and independent loops, loops that iterate zero or one time, and loops that update the loop variables multiple times.

Problets administer the pre-test-practice-post-test protocol: pre-test to evaluate the learner's knowledge; an adaptive practice on only the concepts that the learner has not yet mastered [1], followed by post-test on only the concepts that the learner has practiced. During the pre- and post-tests, problets do not provide any feedback. During practice, problets provide delayed feedback, which includes a narrative of the step-by-step execution of the program [2]. Problets use the concept map of the domain, enhanced with learning objectives, as the overlay student model [3].

Problets use reified interfaces [4] which promote the use of mental models when solving problems, e.g., the learner enters the output of the program one step at a time; and enters each step by first clicking on the line of code that produces the output. A learner identifies a bug in a program by identifying the line of code first, the program object to which the bug applies next, and finally, the specific type of bug that applies to the object.

Reflection

Reflection is conducive to learning [5][6] - it encourages learners to analyze their performance, compare and contrast their actions against those of others, in particular experts, and generalize from the actions they used in similar situations [5]. Therefore, we have been studying the effect of reflection in problets.

Intelligent tutors have promoted reflection of different aspects of learning. The work of [7] focused on reflection on the learners' own thought processes and learning and was done in the context of tutors for learning the skills of tennis, problem solving in algebra and geometry, writing, and reading. The Sherlock II tutor [8][9][10] encouraged learners to reflect on their approach to solving a problem after they had solved it. Our approach in problets has been to encourage the learner to reflect on the concepts underlying the problems.

Several researchers have studied providing reflection during instructional activities [11][6][12]. [13] studied reflection after the instructional activity, and have shown that such post-practice reflection can play a significant role in instructing students in the conceptual knowledge

underlying tutoring tasks. Similarly, in problets, we provide an exercise promoting reflection after each problem.

The learning companion Lucy provided with the PROPA ITS to teach explanatory analysis of satellite activity encourages reflection using menu-driven dialog [14]. Several researchers have used natural language dialogs to promote reflection (e.g., [13]). Researchers have studied the effect of reflection on learning by providing reflection through self-explanation (e.g., [11]), and through inspection of the open student model (e.g., [15]). In problets, we introduced reflection in the form of a multiple-choice question presented after each problem. The question states "This problem illustrates a concept that I picked based on your learning needs. Identify the concept." The learner is provided five choices, each of which is a different concept in the domain. The learner must select the most appropriate concept on which the problem might be based, and cannot go on to the next problem until (s)/he selects the most appropriate concept. The problet records the number of unique concepts selected by the learner up to and including the most appropriate concept. See Figure 2 for a snapshot of the reflection question presented after a problem in a problet on selection statements.

Evaluation

We wanted to evaluate *whether post-practice reflection on the concept underlying each problem helped improve learning in problets*. We conducted several controlled *in-vivo* evaluations of reflection using problets. In this paper, we will present results from the following evaluations:

- Selection problet in fall 2006, used by students of six instructors.
- Selection problet in spring 2007, used by students of twelve instructors.
- while loop problet in fall 2007, used by students of seventeen instructors.

In order to ensure that all the students of an instructor got the same treatment, we randomly divided instructors (rather than students) into control and test groups. We combined the data from all the students of all the instructors in each group.

We used the pre-test-practice-post-test protocol - practice was adaptive, and post-test was restricted to only the concepts on which students practiced, as mentioned earlier. The control group was never presented any reflection questions. The test group was presented a reflection question after each problem during the pre-test, practice and post-test. If the student solved the preceding problem incorrectly, the student was required to answer the reflection question correctly before going on to the next problem. If the student solved the preceding problem correctly, the student had the option of answering or not answering the reflection question before going on to the next problem. But, this optional nature did not affect the results of evaluation because of adaptive nature of the tutors, and the fact that we considered only "worked-through" concepts for analysis, as described in the next paragraph. In order to account for the additional time needed to answer reflection questions, the test group was allowed 10% more time than the control group (33 versus 30 minutes).

For each student, we considered data from only those concepts (henceforth referred to as *worked-through concepts*) on which the student solved problems during all three stages: pre-test, adaptive practice and post-test. Therefore, we did not consider data from any of the concepts on which students solved problems correctly during the pre-test – due to the adaptive nature of practice, students did not solve any problems on these concepts during practice or post-test. We also did not consider data from any concepts on which students did not solve problems during all three stages because they ran out of time, since the tutoring session was limited to a fixed duration of time. We tabulated the number of problems solved, the total score and the average score per problem on the pre-test, practice and post-test on the *worked-through concepts* for each student who had at least one worked-through concept.

Analysis

First, we compared the number of worked-through concepts for the two groups. When we combined data from all three evaluations for aggregate analysis, we found no significant difference between the two groups as shown in Table 1. In other words, our randomization of test subjects was effective - the number of concepts worked-through, and hence, learned by the control and test groups was the same. When we analyzed the data from each tutor/year individually, we found one exception – control group worked through significantly more concepts than test group on the selection tutor in spring 2007 (2.5 versus 1.89).

Table 1: Aggregate analysis – Number of worked-through concepts were the same for control and test groups except on the selection tutor in spring 2007

Practice Concepts	Without Reflection Group	With Reflection Group
Aggregate analysis	1.978	1.974
	$t(239) = 0.022$, $p = 0.982$	
Selection, Fall 06	1.857	2.486
	$t(54) = -1.318$, $p = 0.193$, $d = -0.370$	
Selection, Spring 07	2.500	1.890
	$t(140) = 2.940$, $p = 0.004$, $d = 0.515$	
While Loop, Fall 07	1.231	1.412
	$t(41) = -.908$, $p = 0.369$, $d = -0.2773$	

Next, we analyzed the score per problem on the pre-test and post-test on all the worked-through concepts. We started with aggregate analysis of the data combined from all three evaluations. We conducted a 2 X 2 X 3 mixed factor ANOVA analysis with pre-post as the within-

subjects factor and treatment (reflection versus no reflection), as well as topic/year (selection – fall 06, selection- spring 07, while loops – fall 07) as between-subjects factors.

We found significant main effect for time (pre-test versus post-test), $F(1,235) = 496.239$, $p < 0.001$ - subjects scored significantly higher on the post-test than on the pre-test. This indicates that the tutor is effective in helping students learn concepts. The effect of treatment (no reflection versus reflection) was not significant, $F(1,235) = 0.015$, $p = 0.902$. The interaction between treatment and time was *not* significant, $F(1,235) = 0.559$, $p = 0.455$. In other words, *the improvement in the learning of the students with reflection (measured as pre-post increase in the average score per problem) was no greater than that of the students without reflection* as shown in Table 2. The effect of topic/year (selection – fall 06, selection- spring 07, while loops – fall 07) was not significant, $F(1,235) = 2.184$, $p = 0.115$. The interaction between time and topic/year was significant, $F(1,235) = 6.825$, $p = 0.001$, suggesting difference in the experiences of different cohorts of students with the different tutors.

Table 2: Aggregate analysis - Both the groups improved significantly from pre-test to post-test; the difference between the two groups was not significant on either the pre-test or the post-test

Score per problem	Pre-Test	Post-Test	Pre-post *p*
Control Group (Without Reflection) (*N* =89)			
Average	0.118	0.736	< 0.001
Std-Dev	0.177	0.353	
Test Group (With Reflection) (*N* =152)			
Average	0.144	0.787	< 0.001
Std-Dev	0.183	0.319	
Between groups *p*	0.283	0.266	

Next, we did a 2 X 2 mixed factor ANOVA analysis of the data from individual evaluation of each tutor/semester, with pre-post as the within-subjects factor and treatment as between-subjects factor.

Selection Tutor, Fall 2006: We found a significant main effect for time, $F(1,54) = 104.53$, $p < 0.001$ - subjects scored significantly higher on the post-test than on the pre-test as shown in Table 3. The effect of treatment was not significant, $F(1,54) = 0.148$, $p = 0.702$. Finally, the interaction between treatment and time was *not* significant, $F(1,54) = 0.347$, $p = 0.558$, i.e., the improvement in learning of both the groups was similar.

Table 3: Selection Tutor, Fall 2006 – Same pattern observed as on the aggregate analysis.

Score per problem	Pre-Test	Post-Test	Pre-post *p*
Control Group (Without Reflection) (*N* = 21)			
Average	0.121	0.662	< 0.001
Std-Dev	0.200	0.417	
Test Group (With Reflection) (*N* = 35)			
Average	0.176	0.658	< 0.001
Std-Dev	0.185	0.355	
Between groups *p*	0.313	0.970	

Selection Tutor, Spring 2007: We found a significant main effect for time, $F(1,140) = 568.578$, $p < 0.001$ - subjects scored significantly higher on the post-test than on the pre-test. The main effect of treatment was significant, $F(1,140) = 5.791$, $p = 0.017$ – test group scored better than control group on both the pre-test and post-test as shown in Table 4. In contrast, control group worked through significantly more concepts than test group (2.5 versus 1.89). Finally, the interaction between treatment and time was *not* significant, $F(1,140) = 0.930$, $p = 0.336$, i.e., the improvement in learning was similar for both the groups.

Table 4: Selection Tutor, Spring 2007 - Same pattern observed as on the aggregate analysis.

Score per problem	Pre-Test	Post-Test	Pre-post *p*
Control Group (Without Reflection) (*N* =42)			
Average	0.094	0.764	< 0.001
Std-Dev	0.147	0.295	
Test Group (With Reflection) (*N* =100)			
Average	0.135	0.862	< 0.001
Std-Dev	0.185	0.249	
Between groups *p*	0.164	0.064	

While Loop Tutor, Fall 2007: We found a significant main effect for time, $F(1,41) = 73.751$, $p < 0.0001$ - subjects scored significantly higher on the post-test than on the pre-test as shown in Table 5. The main effect of treatment was not significant, $F(1,41) = 1.152$, $p = 0.289$. Finally, the interaction between treatment and time was *not* significant, $F(1,41) = 0.840$, $p = 0.365$.

Table 5: While Loop Tutor, Fall 2007 - Same pattern observed as on the aggregate analysis.

Score per problem	Pre-Test	Post-Test	Pre-post *p*
Control Group (Without Reflection) (*N* =26)			
Average	0.154	0.750	< 0.001
Std-Dev	0.200	0.387	
Test Group (With Reflection) (*N* =17)			
Average	0.128	0.609	< 0.001
Std-Dev	0.170	0.450	
Between groups *p*	0.648	0.296	

In summary, we found the same pattern in both aggregate analysis and the three individual analyses – significant pre-post improvement, attesting to the effectiveness of the problet, but no significant interaction between treatment and pre-post improvement, indicating that the improvement in learning was not affected one way or the other by post-practice reflection on the concept underlying each problem.

Problems Solved During Practice: Next, we considered the number of problems solved by the two groups during practice, as well as the number of practice problems solved

per worked-through concept. As shown in Table 6, on aggregate analysis, there was no significant difference between the two groups on either the number of practice problems solved, or the number of practice problems per worked-through concept. On analysis of the data of individual tutors, we observed the same pattern, except in spring 2007, when there was a significant difference between the two groups on the number of practice problems solved. This was to be expected considering that the control group worked through significantly more concepts than the test group, as shown in Table 6. Even so, there was no significant difference between the two groups on the number of practice problems per worked-through concept. *In other words, there was no significant difference in the number of problems needed to learn each concept, with or without reflection on the concept underlying each problem.*

Table 6: Both groups solved the same number of problems during practice (except in spring 2007). Both groups solved the same number of problems per worked-through concept.

Practice	Without Reflection	With Reflection	*p*-value
Aggregate analysis			
Problems	7.674	6.684	0.282
Problems per concept	3.567	3.585	0.952
Selection Tutor, Fall 2006			
Problems	4.619	7.314	0.148
Problems per concept	2.282	3.300	0.136
Selection Tutor, Spring 2007			
Problems	10.952	6.650	0.003
Problems per concept	4.1480	3.618	0.244
While Loop Tutor, Fall 2007			
Problems	4.846	5.588	0.560
Problems per concept	3.667	3.980	0.525

Discussion

Our data supports neither that reflection on the concept underlying each problem promotes greater learning, measured as pre-post increase in the score per problem, nor that it promotes faster learning, measured as the number of practice problems solved per concept. This result is counter-intuitive.

It could be that the *multiple-choice format that we use to provide reflection is not effective*. It would seem (to us) that listing high-level domain principles as choices, and mandating that the learner identify the correct principle underlying a problem, before proceeding to the next problem would force the learner to reflect on the principles underlying the problem. Moreover, our prior analysis has shown that correctly solving problems is closely associated with the number of attempts needed to identify the underlying concept during post-practice reflection [16]. Therefore, there is no conceptual disconnect between the problems and the reflection questions. This brings us back to the format we used, viz., multiple-choice. In the future, we will evaluate whether a different format may be better suited to promoting reflection on the concept underlying each problem.

The focus in problets is on helping the learner construct the proper mental model underlying a problem. This is reflected in the design of the user interface, feedback and problems generated by problets:

- Reified User Interface [4]: The user interface forces the learner to step through a model of the program when solving problems. E.g., in order to debug a program, the learner must first select the line of code where she/he thinks the bug exists (e.g., line 9), followed by the program object on that line to which the bug applies (e.g., pointer ptr), followed by the actual nature of the bug (e.g., Dangling pointer because the variable to which ptr points has already gone out of scope). A student cannot correctly solve problems without a deep understanding of the programming domain.
- Feedback: Problets explain the step-by-step execution of the program as part of the feedback [2]. The explanation includes any semantic/run-time errors in the program, any output generated by the program, any changes in the state of the program, etc. This helps the learner build the correct mental model of a program, or correct the notional machine that the learner has already built.
- Problem generation: Problets generate problems as instances of parameterized templates. Each template can be "instantiated" modulo identifiers, data types and literal constants to generate an infinite number of similar, but un-identical problems. Therefore, no matter how many problems a problet generates from the same template (such as during pre-test and post-test), the learner must use the mental model of the domain to solve each problem, and cannot simply memorize and transfer the solution from one instance of a template to another.

Reflection helps the learner identify the concept underlying a problem and inter-relate concepts in the domain. Problets promote these activities by forcing the learner to use a mental model of the domain to solve problems. Therefore, adding reflection exercises to problets may not accrue additional benefits to the learner. In other words, the *benefits of reflecting on the concept underlying each problem may be limited if a tutor already promotes deep understanding of the domain*.

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

References

1. Kumar, A.N. *A Scalable Solution for Adaptive Problem Sequencing and its Evaluation.* in *International Conference on Adaptive Hypermedia*

and Adaptive Web-Based Systems (AH 2006). 2006. Dublin, Ireland.
2. Kumar, A.N., *Explanation of step-by-step execution as feedback for problems on program analysis, and its generation in model-based problem-solving tutors.* Technology, Instruction, Cognition and Learning (TICL) Journal, 2006. **4**(1).
3. Kumar, A.N. *Using Enhanced Concept Map for Student Modeling in a Model-Based Programming Tutor*. in *International FLAIRS conference on Artificial Intelligence*. 2006. Melbourne Beach, FL.
4. Kumar, A.N. *A Reified Interface for a Tutor on Program Debugging*. in *3rd IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*. 2003. Athens, Greece.
5. Collins, A., *Cognitive Apprenticeship and Instructional Technology*, in *Educational Values and Cognitive Instruction*, L. Idol and B.F. Jones, Editors. 1990, Lawrence Erlbaum: Hillsdale, NJ.
6. Frederiksen, J.R. and B.Y. White. *Cognitive Facilitation: A Method for Promoting Reflective Collaboration*. in *2nd International Conference on Computer Support for Collaborative Learning*. 1997. Toronto, Canada.
7. Collins, A. and J.S. Brown, *The computer as a tool for learning through reflection*, in *Learning Issues for Intelligent Tutoring Systems*. 1988, Springer-Verlag, Inc: New York. p. 1-18.
8. Katz, S. and A. Lesgold, *The Role of the Tutor in Computer-Based Collaborative Learning Situations*, in *Computers as Cognitive Tools*, S.P. Lajoie and S.J. Derry, Editors. 1993, Lawrence Erlbaum: Hillsdale, NJ. p. 289-317.
9. Katz, S., et al., *Sherlock 2: An intelligent tutoring system built upon the LRDC Tutor Framework*, in *Facilitating the Development and Use of Interactive Learning Environments*, C.P. Bloom and R.B. Loftin, Editors. 1998, Lawrence Erlbaum Associates. p. 227-258.
10. Lesgold, A., et al., *Extensions of intelligent tutoring paradigms to support collaborative learning*, in *Instructional Models in Computer-Based Learning Environments*, S. Dijkstra, H. Krammer, and J.V. Merrienboer, Editors. 1992, Springer-Verlag: Berlin. p. 291-311.
11. Chi, M.T.H. and K. VanLehn, *The content of physics self-explanations.* Journal of the Learning Sciences, 1991. **1**: p. 69-105.
12. VanLehn, K., R.M. Jones, and M.T.H. Chi, *A model of the self-explanation effect.* Journal of the Learning Sciences, 1992. **2**: p. 1-59.
13. Katz, S., G. O'Donnell, and H. Kay, *An Approach to Analyzing the Role and Structure of Reflective Dialogue.* International Journal of Artificial Intelligence in Education, 2000. **11**: p. 320-343.
14. Goodman, B., et al., *Encouraging Student Reflection and Articulation using a Learning Companion.* International Journal of Artificial Intelligence in Education, 1998. **9**: p. 237-255.
15. Hartley, D. and A. Mitrovic. *Supporting Learning by Opening the Student Model*. in *6th International Conference on Intelligent Tutoring Systems*. 2002. Biarritz, France: Springer Verlag.
16. Kumar, A.N. and P. Rutigliano. *Analyzing the Data Collected by Programming Tutors that Provide Post-Practice Reflection*. in *7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007) Workshop on Educational Data Mining*. 2007. Niigata, Japan.

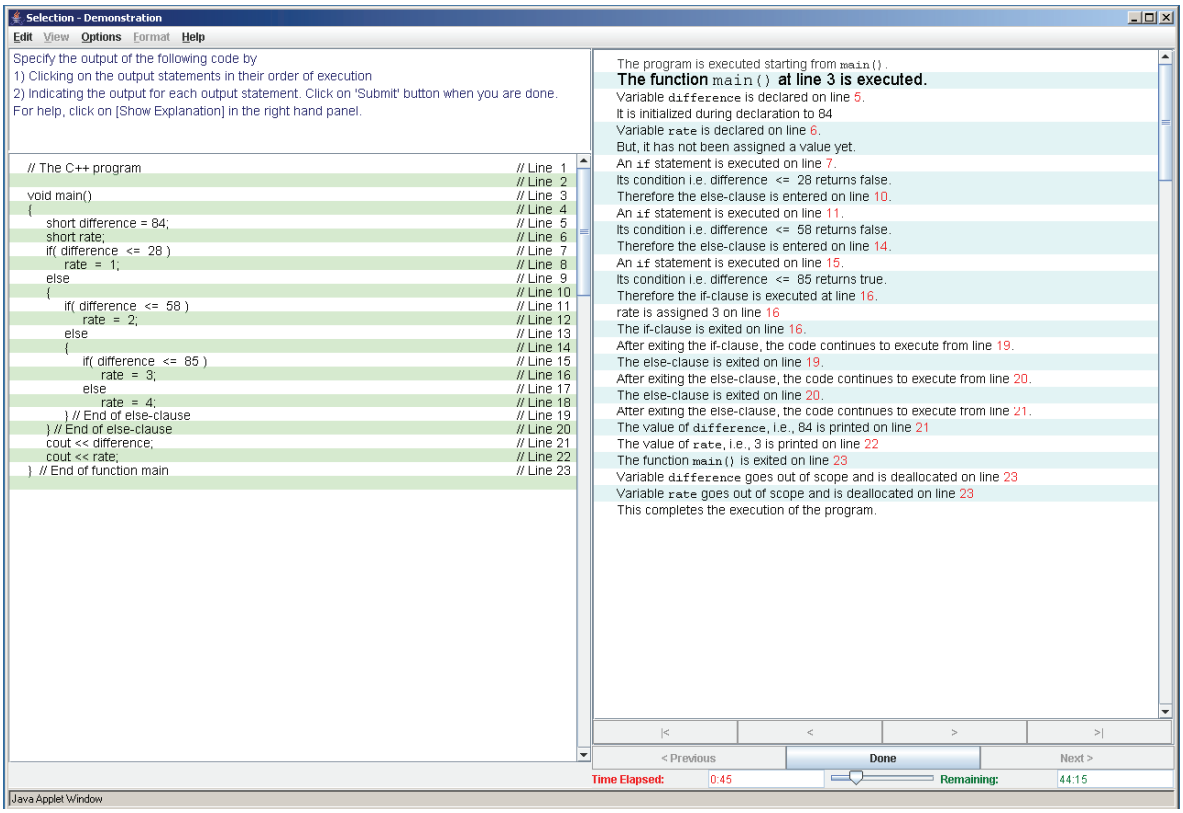

Figure 1: Snapshot of the problet on selection statements – program in the left panel, explanation of the step-by-step execution in the right panel

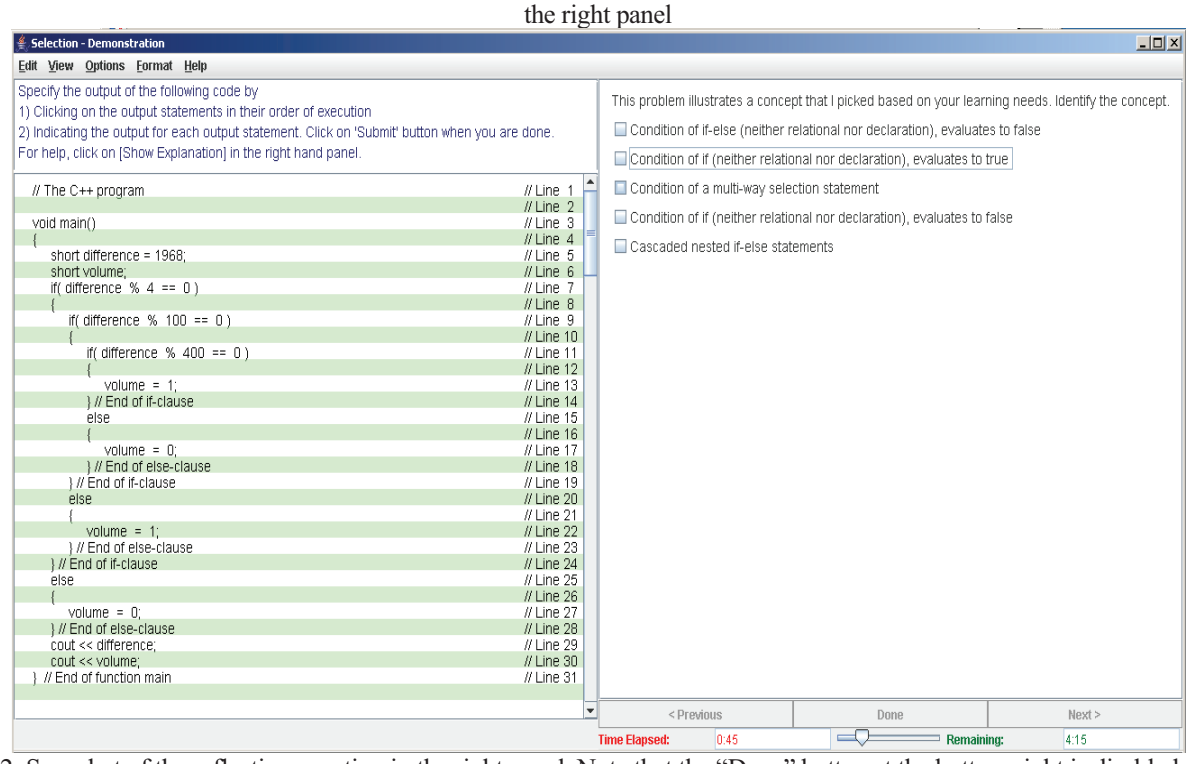

Figure 2: Snapshot of the reflection question in the right panel. Note that the "Done" button at the bottom right is disabled until the student identifies the correct concept underlying the problem in the left panel.