Improving Student Computer Programming Understanding and Engagement Through Computer-Monitored Problem-Solving Discussions

Jung Hee Kim, Michael Glass, Kelvin Bryant, Taehee Kim

Dept. of Computer science & Dept. of Human Development and Services
North Carolina A&T State University & Valparaiso University

Introduction

This project introduces small-group collaborative discussion problems into a CS2 Java programming class. Over 10 years from 2003 to 2013 approximately 1/3 of students did not succeed enough to proceed. We aim to improve learning of Java concepts and increase student interest. Students keyboard-chat via a web page in groups of 3 or 4, with an instructor or TA overseeing each group. Four exercises were administered during the Fall, 2015 semester, covering the following topic areas:

- Classes, objects, and references
- Inheritance
- Swing Graphical User Interface (now updated to Java FX)
- Exceptions and exception handling

The project has three thrusts:

- Curriculum that fosters productive collaborative discussion
- Student motivation and self-efficacy at the level of the entire semester and in individual exercises,
- Technology to monitor the exercises in real-time as an aid to the instructor.

Here we report on experience in all three thrusts after the first semester of applying these labs.

Example Problem

```
Problem 2: What output is produced by the
following code sequence?

Scanner scan = new Scanner( System.in );
int number = 0;
try
{
  // User enters 12, ENTER
  number = scan.nextInt( );
  System.out.println( "1: " + number );

  // User enters 56, hits ENTER
  number = scan.nextInt( );
  System.out.println( "2: " +  number );

  // User enters 99ABC, hits ENTER
  number = scan.nextInt( );
  System.out.println( "3: " +  number );
}
catch( Exception e)
{
  String s = scan.nextLine( );
  System.out.println( "4: " + number );
}
finally
{ // User enters 100, hits ENTER
  number = scan.nextInt( );
  System.out.println( "5: " + number );
}
Answer (4 lines):
1: 12  2: 56  4: 56   5: 100
```

Script: reach agreement, then validate answer with the instructor, continue if needed. Repeat the process for each question.

Following the script engages all students, prompts them to articulate their reasoning, and reduces shallow reasoning and guessing.

Example Dialogue

Who	Dialogue Turn
A:	99ABC isn't compatible with an int type, so what type of exception would that be
B:	InputMisMatchException
B:	or NumberFormatException ?
A:	wouldn't it be numberFormatException because I thought that InputMismatch was when let's say you're checking for more things than there are in the file
B:	what do you mean more things than there are in the file?
B:	Input mismatch is if you're supposed to be taking in a int like using scan.nextInt but you're reading in a strin ginstead
C:	The type of exception does not really matter in this problem because the catch block covers all types of exceptions
A:	so would it be 4
B:	don't we still need to go through the finally block?
C:	I think the program outputs the first couple of lines before the code crashed and then goes to the catch and finally blocks
A:	ok so 1:12
A:	2: 56, 4: ,5:100?
C:	I agree with you, but I think 56 will be in front of 4
B:	I think it also prints 4: 56
A:	why wouldn't 56 go with 2
B:	it goes with both
C:	I agree with B
A:	oh so 1:12,2:56,4:56,5:100
B:	yes

Learning and Happiness

We analyzed learning gains according to the relative position of each student in a discussion group (highest, middle, lowest), based on pre-test score. The high student in each discussion lost ground on average, the other two gained.

Table 1. Average Learning Gains

	Highest Stratum	Middle	Lowest	Class average
Lab3 (n=32)	-0.32	0.45	0.34	0.23
Lab4 (n=24)	0.02	0.34	0.47	0.32

However the post-test scores for high, mid, and low students tended to become more alike.

Table 2. Lab 3 Pre-test and Post-test Scores

	Pre-Test Mean / SD		Post- Test Mean / SD		t	Sig
High	12.25	3.41	9.75	6.06	t (11)= 1.94	p =.078
Mid	5.92	5.05	12.25	4.41	t (11)= -3.74	p =.003
Low	3.75	3.1	9.25	4.56	t (7)= -4.72	p = .002

Students rated the last two labs more highly than the first two labs, possibly as they became more accustomed to the group exercises.

Table 3: After lab surveys

	Effectiveness of group work Mean / SD		Understanding of concept Mean / SD		Interest in lab Mean /SD	
Lab1	3.17	0.68	3.45	0.96	3.19	0.94
Lab2	3.08	0.93	3.42	1.05	3.08	0.93
Lab3	3.47	0.71	4.03	1.06	3.65	0.76
Lab4	3.40	0.61	3.78	0.85	3.17	0.89

No noticeable change in student interest.

Table 4. Beginning and End of Semester Surveys

Time	Interest	Efficacy
beginning of sem.	4.33 / 5	2.83 / 5
ending of sem.	4.32 / 5	3.81 / 5

Future Work

Fostering more productive discussions:

- Have the pre-test question overlap the discussion exercise (the students will be primed).
- Add a separate text box in the computer interface to compose a written answer + explanation.

Motivation:

- Relate semester-level student enthusiasm to situational interest in the labs.
- Study how to match students into groups based on their interest levels and scores.

Technology:

- Machine-identification of features to feed a dashboard evaluation of the discussion.

Experiment in Monitoring

Who	Dialogue Turn
A:	f and foo are the refernece variables
A:	so those together make 16? for the refrence types
B:	yup yup
A:	16 bytes
C:	2a = 20
B:	:D
B:	there ya go lol
D:	Wait where did you get 16?
D:	wouldnt it be 48 at least for the main method
D:	because the array creates 5 object
A:	oh yeah i looked over that was just countingm f and foo
C:	those are on the heap not the stack
D:	So the objects created by an array are on the heap

Can the computer recognize the gender of participants?

- Transcripts were annotated by machine for features such as discourse markers, smilies, participation, certain pronouns.
- Transcripts were annotated by hand for several affective features.
- Unable to train classifier to recognize gender. based on individual turns.
- Using aggregated turns of each participant, participation (% of turns of whole dialogue) and apologetic affect were weakly predictive of gender.

Affective features were chosen for:

- Potentially indicative of state of discourse
- Potentially measurable by text algorithms

Affective features key:

excited apologetic confused humorous

Acknowledgement

Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award No. 1504917. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.