On the Role and Effectiveness of Pop Quizzes in CS1

Vincent A. Cicirello
Computer Science and Information Systems
The Richard Stockton College
Pomona, NJ
cicirelv@stockton.edu

ABSTRACT

In this paper, we explore the potential important role that unannounced (or "pop") quizzes can play in CS-1. "Pop" quizzes generate continual feedback to both the student and the instructor of the course. They also encourage students to avoid missing class unless necessary. We present the results of a three year study on the effectiveness of "pop" quizzes in CS-1. Our results demonstrate that students who experience "pop" quizzes in CS-1 score higher on exams than do their counterparts who did not experience "pop" quizzes. This appears especially true for upperclassmen, particularly for juniors. The use of "pop" quizzes in CS-1 also improves performance on programming assignments for juniors. CS/IS majors and Math majors both receive a greater benefit from "pop" quizzes than do other non-majors.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education

General Terms

Experimentation, Human Factors

1. INTRODUCTION

In this paper, we explore the role that "pop" quizzes can play in CS-1 courses; and provide empirical evidence of their effectiveness on student learning. Our study of "pop" quizzes in CS-1 spanned three academic years and nine sections of CS-1, a course that we call "Programming and Problem Solving I". Three of those sections were taught without "pop" quizzes. The other six used "pop" quizzes. No other major pedagogical difference existed among the nine sections of the course, and all were taught by the same instructor. Total enrollment was 219 with 70 in the "quiz-free" sections and 149 in the sections where "pop" quizzes were used.

"Pop" quizzes serve several roles. First, the author does not like explicitly factoring attendance into semester grades. "Pop" quizzes are used to randomly sample attendance and indirectly factor it into course grades. Second (and more importantly), frequent quizzes provide students with continual feedback of their learning progress. Third, quizzes also provide the instructor with continual feedback of student progress, helping to guide the design of review sessions.

There is usually a general grumbling from a class of students when they discover a course will use "pop" quizzes. We doubt any student will ever claim aloud to like unannounced quizzes. However, out of 149 students, only 2 have mentioned the quizzes in the open-ended anonymous comments on our end of semester student evaluations. One of those wrote that the quizzes motivated them to review their notes on a regular basis. The other wrote the more expected yet clever, "quizzes.setYuck(true);". To assist students in realizing that the "pop" quizzes are beneficial, or at least to encourage them to tolerate them, 50% of their grade an any given quiz comes from putting their name on it, while the other 50% actually comes from performance on the quiz. The quizzes count for 10% of the semester grade. However, due to half the grade coming from taking the quiz, you can view this as 5% of a randomly sampled attendance record and 5% quiz performance. We do not mention this interpretation to the students. If a student misses a quiz for a documented legitimate excuse, the missed quiz does not count against them and they get a blank copy for review (no makeup quiz). An unexcused missed quiz counts as a 0.

There have been few studies of quiz effectiveness on student learning in programming courses—e.g., Woit and Mason compared using online weekly quizzes and unmarked lab assignments to using marked assignments without quizzes, and showed the benefit of quizzes [6]. Others use quizzes to prevent programming plagiarism (e.g., [2]). Our use of quizzes is not meant to replace graded lab assignments, but rather to augment them. Prior studies focus on scheduled quizzes, rather than "pop" quizzes. Although the concept of a pop quiz is certainly not new, as far as the author is aware, this is the first comprehensive study of "pop" quiz effectiveness in enhancing student learning in CS-1. Our objectives are to validate pop quiz effectiveness and to categorize efficacy by majors and academic years of CS-1 students.

We begin with an overview of our approach to CS-1 in Section 2 and a summary of the majors and academic years of the CS-1 students in Section 3. In Section 4 we analyze the effectiveness of "pop" quizzes in CS-1. We will see that "pop" quizzes lead to increased exam grades, especially for juniors and seniors, and reduce the use of exam curves on the more difficult topics. We conclude in Section 5.

SIGCSE'09, March 3–7, 2009, Chattanooga, Tennessee, USA.

Table 1: Distribution of students by major

Major	No Quiz		Pop Quiz		All	
	#	%	#	%	#	%
CS/IS	24	34.3	46	30.9	70	32.0
Mathematics	29	41.4	57	38.3	86	39.3
Business	3	4.3	10	6.7	13	5.9
Physics	3	4.3	8	5.4	11	5.0
Biology	1	1.4	3	2.0	4	1.8
Chemistry	1	1.4	0	0.0	1	0.5
Marine Science	0	0.0	1	0.7	1	0.5
Psychology	1	1.4	1	0.7	2	0.9
Communication	1	1.4	0	0.0	1	0.5
History	1	1.4	1	0.7	2	0.9
Literature	1	1.4	0	0.0	1	0.5
Undeclared	3	4.3	21	14.1	24	11.0
Non-Matric.	2	2.9	1	0.7	3	1.4
Total	70	100.0	149	100.0	219	100.0

2. OUR CS-1 COURSE

We introduce object-oriented concepts early in our CS-1/CS-2 sequence. We currently use Cay Horstmann's "Java Concepts" textbook[3]. The pre-requisite for our CS-1 is either Discrete Math or Calculus or either of those courses concurrently with CS-1. In CS-1, we provide a general overview of programming, and proceed by covering how to manipulate objects. We spend a significant amount of time on implementing classes. We then cover primitive data types, decision statements, loops and iteration, arrays, debugging. We integrate unit testing throughout. We do not cover any of the more advanced object-oriented concepts in CS-1 such as inheritance, interfaces, etc (we continue with those in CS-2).

The content and sequence of topics remained constant across all nine sections of CS-1 used in this study, and all were taught by the same instructor during a three year period. During that time, four other sections of the course were taught by other faculty members, but those sections were not used in this study. The pedagogical strategies used by the instructor include significant use of "live" programming [5] and the use of "pair" programming[4, 1]. These strategies were used consistently across all nine sections of the course. The only major pedagogical difference is the use of "pop" quizzes in six of the sections of the course. In the "quiz-free" sections, programming assignments accounted for 45% of the students' grades, exams accounted for 50%, and the remaining 5% came from participation. In the "pop-quiz" sections, quizzes accounted for 10%, programming assignments 40%, exams 45%, and 5% for participation.

3. STUDENT DEMOGRAPHICS

CS-1 is a core requirement for our "Computer Science and Information Systems" (CS/IS) major as well as for the Mathematics major. CS and IS are not separate majors. Instead, we have a single integrated major, and students choose a concentration in either CS or IS. Table 1 summarizes the distribution of majors for the "quiz-free" and the "pop-quiz" sections of the course. The two dominant groups are CS/IS majors and Math majors, together accounting for 71.3% of the students. Approximately 11% have not declared a major. The next two groups in order of significance are Business and Physics with 5.9% and 5.0%, respectively. Several other majors are represented by no more than 4 students each.

Table 2: Distribution of students by academic year

Major	No Quiz		Pop Quiz		All	
	#	%	#	%	#	%
Freshmen	23	32.9	60	40.3	83	37.9
Sophomores	22	31.4	40	26.8	62	28.3
Juniors	17	24.3	38	25.5	55	25.1
Seniors	8	11.4	11	7.4	19	8.7
Total	70	100.0	149	100.0	219	100.0

Table 3: Distribution of students by major and year for all nine sections of CS-1. FR = freshmen, SO = sophomores, JR = juniors, SR = seniors

Major	FR	SO	JR	SR
CS/IS	39	20	10	1
Mathematics	19	18	35	14
Business	5	3	4	1
Physics	2	4	3	2
Biology	3	1	0	0
Chemistry	0	1	0	0
Marine Science	1	0	0	0
Psychology	0	2	0	0
Communication	0	1	0	0
History	1	1	0	0
Literature	1	0	0	0
Undeclared	12	10	2	0
Non-Matric.	0	1	1	1

Table 2 summarizes the distribution of students in our CS-1 course by academic year for the "no quiz" and "pop quiz" sections. Additionally, Table 3 shows the number of students in each academic year for each major. The dominant group are freshmen (37.9%) since most of our CS/IS majors who intend to major in CS/IS from the start take CS-1 in the first year (nearly 56% of CS/IS majors in the study). There is also a significant group (22%) of Math majors who take CS-1 in the freshmen year. This is also the most common time for undeclared students to take CS-1 (half of the undeclared students in the study). The next largest group are sophomores (28.3%) including CS/IS majors who either decided their major after beginning their college career or who were not ready for CS-1 in the first year (e.g., perhaps needed to increase their mathematics skills). A significant number of Math majors also take CS-1 in their sophomore year, as well as undeclared students. Juniors (25.1%) make up another significant group, largely because junior year is the most common time for Math majors at our institution to take CS-1 (41% of the Math majors in the study). The remaining 8.7% are seniors (almost entirely Math majors).

4. EMPIRICAL ANALYSIS

We now present an analysis of our empirical data. We begin by exploring differences in the performance of students in the "quiz-free" sections of the course with those who were given "pop quizzes". In Table 4, we summarize this comparison for several performance metrics, including programming assignments, exams, and students' overall course grades.

Exam coverage is consistent across all nine sections of the course. Exam 1 covers introductory programming concepts, variable declarations, assignment statements, manipulating objects (e.g., construction, method calls, object references),

Table 4: Student performance (ALL students)

Course Element	No Quizzes ($N = 62$)	Pop Quizzes ($N = 138$)	P
Programs	87.4 ± 4.2	88.4 ± 2.3	0.351
Exam 1 (raw)	80.9 ± 3.2	81.7 ± 2.4	0.336
Exam 2 (raw)	65.9 ± 3.8	73.5 ± 2.8	0.001
Exam 3 (raw)	55.1 ± 4.6	62.1 ± 2.8	0.006
Semester grade	80.8 ± 3.3	81.6 ± 1.9	0.332

Table 5: Student performance (Freshmen only)

Course Element	No Quizzes ($N = 20$)	Pop Quizzes ($N = 56$)	P
Programs	86.6 ± 7.7	86.9 ± 4.0	0.475
Exam 1 (raw)	78.7 ± 6.1	79.7 ± 4.0	0.389
Exam 2 (raw)	63.8 ± 6.8	69.9 ± 4.4	0.076
Exam 3 (raw)	52.0 ± 7.7	59.2 ± 4.4	0.059
Semester grade	78.9 ± 6.6	79.7 ± 3.2	0.413

Table 6: Student performance (Sophomores only)

Course Element	No Quizzes ($N = 19$)	Pop Quizzes ($N = 36$)	P
Programs	86.5 ± 8.4	84.2 ± 5.3	0.329
Exam 1 (raw)	83.7 ± 4.7	82.4 ± 5.1	0.359
Exam 2 (raw)	67.0 ± 8.2	75.2 ± 5.8	0.058
Exam 3 (raw)	56.7 ± 8.3	61.3 ± 6.6	0.200
Semester grade	81.5 ± 5.9	79.7 ± 4.6	0.324

Table 7: Student performance (Underclassmen)

Course Element	No Quizzes ($N = 39$)	Pop Quizzes ($N = 92$)	P
Programs	86.5 ± 5.6	85.8 ± 3.2	0.416
Exam 1 (raw)	81.1 ± 3.9	80.8 ± 3.2	0.447
Exam 2 (raw)	65.4 ± 5.2	72.0 ± 3.5	0.022
Exam 3 (raw)	54.3 ± 5.6	60.0 ± 3.7	0.049
Semester grade	80.1 ± 4.4	79.7 ± 2.7	0.435

and general object-oriented concepts. Exam 2 covers implementation of classes, primitive data types, and decision statements. Exam 3 covers loops, iteration, arrays (and ArrayLists), and topics related to designing classes.

The exam data shown is for "raw" exam scores, prior to any applied curve. The overall grade is shown for informational purposes only, as it includes factors not in the Table (e.g., participation) and were computed using curved exam scores in some cases—e.g., for all 3 "quiz-free" sections, Exam 3 was curved and for 2 of those sections Exam 2 was also curved; while for the "pop" quiz sections only Exam 3 was ever curved and for only 2 of the 6 sections.

Table 4 shows averages for various grade categories, with 95% confidence intervals, and the P-values from T-tests. Although the performance of students that experienced the "pop" quizzes is better for all categories, the difference is significant in only two cases. The raw exam averages for Exam 2 and Exam 3 are approximately 7.5 and 7.0 points higher, respectively, when students take "pop" quizzes than when they do not. T-tests show these differences to be highly significant (P values of 0.001 and 0.006, respectively).

Next, we refine our analysis to explore the potential benefits of "pop" quizzes by academic year (Section 4.1) and by major (Section 4.2). We then examine the effect of "pop" quizzes on the overall end of semester grades in Section 4.3.

4.1 Performance by Academic Year

Freshmen: In Table 5, we summarize performance with and without "pop" quizzes for all students with freshmen status (less than 32 credits completed). Students in the "pop" quiz sections averaged approximately 6 and 7 points higher on Exam 2 and Exam 3, respectively, than did their counterparts who did not experience the quizzes. These results, however, are only marginally significant by a T-test. Differences on other performance metrics were not significant.

Sophomores: Sophomores (at least 32 credits and less than 64 credits) in the "pop" quiz sections scored 8.2 points higher on Exam 2 than those who did not experience the "pop" quizzes (see Table 6), but this result is also only marginally significant. Differences on all other performance metrics for sophomores are not significant.

All Underclassmen: Table 7 compares the performance of freshmen and sophomores combined in "pop" quiz sections with those in quiz-free sections. Underclassmen score nearly 7 and 6 points higher on Exams 2 and 3, respectively, when "pop" quizzes are used (with statistical significance).

Juniors: The students who seem to benefit the most from the use of "pop" quizzes are upperclassmen. This is especially true specifically of students in their junior year. Table 8 summarizes results for students with junior status (at least 64 credits completed and less than 96). Juniors score over 13 points higher on Exam 2 when they experience "pop" quizzes. This difference is highly significant (P value of 0.0017). Additionally, they score 17.5 points higher on average on Exam 3 when they are given "pop" quizzes (difference is extremely significant with a P value of 0.0008).

Perhaps most interesting is the comparison of programming assignments for juniors in the "pop" quiz sections and those who were not given quizzes. Although only marginally significant with a P value of 0.072, juniors who experienced "pop" quizzes scored over 7 points higher on programming assignments than their counterparts in the sections without quizzes. We were surprised to find this difference.

Juniors also scored nearly 7 points higher on their overall semester average when "pop" quizzes were used (statistically significant with a P value of 0.0316). This is extremely promising when you consider that the overall semester grades of students in the quiz-free sections used curved exam grades. We have been able to eliminate exam curves through the introduction of "pop" quizzes, while retaining the difficulty level of exams, implying that students are learning more.

All Upperclassmen: Our study included too few students with senior status (96 credits or more completed) to examine seniors independently. Table 9 summarizes the results if we combine juniors and seniors. Upperclassmen score an average of nearly 10 points higher on both Exam 2 and Exam 3 when "pop" quizzes are used (significant with P values of 0.004 and 0.019, respectively). Upperclassmen who are given "pop" quizzes also perform better on programming assignments and in their overall semester grade, although these differences are only marginally significant.

So why do "pop" quizzes seem to help juniors and seniors with the learning process more so than freshmen and

Table 8: Student performance (Juniors only)

Course Element	No Quizzes ($N = 16$)	Pop Quizzes ($N = 36$)	P
Programs	85.6 ± 8.8	92.8 ± 2.7	0.0720
Exam 1 (raw)	80.3 ± 6.0	83.6 ± 4.4	0.1994
Exam 2 (raw)	64.7 ± 6.8	77.9 ± 4.4	0.0017
Exam 3 (raw)	50.1 ± 8.7	67.6 ± 3.6	0.0008
Semester grade	78.7 ± 6.2	85.4 ± 2.4	0.0316

Table 9: Student performance (Upperclassmen)

Course Element	No Quizzes ($N = 23$)	Pop Quizzes ($N = 46$)	P
Programs	88.9 ± 6.5	93.4 ± 2.2	0.099
Exam 1 (raw)	80.4 ± 5.5	83.6 ± 3.7	0.174
Exam 2 (raw)	66.9 ± 5.4	76.7 ± 4.2	0.004
Exam 3 (raw)	56.5 ± 8.0	66.2 ± 3.7	0.019
Semester grade	81.8 ± 5.0	85.4 ± 2.0	0.098

Table 10: Student performance (CS and IS majors)

Course Element	No Quizzes ($N = 21$)	Pop Quizzes ($N = 42$)	P
Programs	85.6 ± 8.8	88.4 ± 4.2	0.291
Exam 1 (raw)	76.4 ± 6.4	81.9 ± 4.7	0.089
Exam 2 (raw)	60.8 ± 6.9	72.4 ± 5.1	0.006
Exam 3 (raw)	51.6 ± 8.9	63.0 ± 5.4	0.019
Semester grade	78.0 ± 7.2	81.6 ± 3.6	0.192

Table 11: Student performance (Math majors)

Course Element	No Quizzes ($N = 27$)	Pop Quizzes ($N = 55$)	P
Programs	89.1 ± 5.6	90.9 ± 3.1	0.294
Exam 1 (raw)	83.1 ± 4.5	83.1 ± 3.5	0.487
Exam 2 (raw)	69.7 ± 5.6	77.0 ± 4.2	0.024
Exam 3 (raw)	58.6 ± 7.0	63.9 ± 3.9	0.099
Semester grade	83.1 ± 4.4	83.7 ± 2.6	0.398

sophomores? One hypothesis here is that as they have matured into the role of a college student, juniors and seniors have perhaps developed more sophisticated studying and review strategies. Perhaps knowing of the possibility of a "pop" quiz, they may be more likely to review continuously throughout the semester to be prepared rather than cramming shortly prior to exams. Or perhaps their learning process has matured in such a way that they are better able to use the early feedback that "pop" quizzes provide.

4.2 Performance by Declared Major

CS and IS Majors: Table 10 summarizes the results for CS/IS majors. Although students in our integrated CS/IS program declare a concentration in either CS or IS, no data on selected concentrations is available for students in the study. We encourage CS/IS students to complete the core requirements in common to both concentrations prior to selection. CS-1 is one of seven courses in this common set of core requirements, and thus many CS/IS students may not have chosen their concentration by the time they take CS-1.

CS/IS majors scored nearly 12 points higher on Exams 2 and 3 when "pop" quizzes are used (statistically significant with P values 0.006 and 0.019, respectively). Additionally, although only marginally significant (P value 0.089), CS/IS majors scored over 5 points higher on Exam 1 when "pop" quizzes were used. The benefit of "pop" quizzes for CS/IS majors, therefore, is not limited to the more difficult topics.

Math Majors: The dominant group of students in our CS-1 course are mathematics majors. 39.3% of the students who enroll in CS-1 are Math majors, while 32% are CS/IS majors (Table 1). As seen in Table 11, students majoring in Math also benefit from "pop" quizzes in CS-1. They score over 7 points higher on Exam 2 when "pop" quizzes are used (statistically significant with P value 0.024). Math majors also scored over 5 points higher on Exam 3 when "pop" quizzes were used (although only marginally significant).

Why do CS/IS majors experience a more significant increase in performance with "pop" quizzes than do Math majors? If you compare the data in Table 11 (Math majors) with that of Table 10 (CS/IS majors), you will see that Math majors in sections without quizzes were already performing (on Exam 1, programming assignments, and overall semester) at or above the level that the "pop" quizzes brought our CS/IS majors to. On Exams 2 and 3, in sections without quizzes, Math majors were scoring nearly a letter grade better (9 and 7 points, respectively) than CS/IS majors. Therefore, there was less room for further significant improvement caused by quizzes for Math majors.

Possible explanations for why Math majors perform better in CS-1 than CS/IS majors are related to when Math majors take the course. As indicated earlier in Table 3, Math majors are more likely to take CS-1 in the junior year than in any other academic year. By the time a student is in their junior year, their study practices may have matured significantly. Additionally, a junior Math major has a far more sophisticated math background than freshmen and sophomores of any major. This strong math background can help them more effectively learn programming concepts. Junior Math majors are also very likely to have encountered programming concepts through courses that use systems such as MatLab or Maple (e.g., variable declarations, iteration, loops, decision statements, and possibly even arrays).

Other Non-Majors: No significant differences in performance were seen for non-majors (majors other than CS/IS and Math) due to the use of "pop" quizzes. A few point increase on Exams 2 and 3 were seen, but the differences are not significant statistically. More peculiar (but also not statistically significant) is a slight decrease on Exam 1 which covers "easier" material. The author is not sure why "pop" quizzes do not benefit non-majors in the way they do majors.

4.3 Effect on Semester Grade

Table 13 shows the percent of students in the "pop" quiz and the "quiz-free" sections that achieved a grade of A, and the percent of students who remained in the course until the end and achieved a B or greater, C or greater, and D or greater. It also lists the percent of students who failed and the percent who withdrew prior to completion. The data is organized by academic status and major. Table 13(a) summarizes the data for all students in the study.

"Pop" quizzes have reduced the withdraw rate from 11.4% to 7.4% after "pop" quizzes were introduced. This reduction in the rate of withdraws is seen for both underclassmen and upperclassmen, as well as for CS/IS majors, Math majors,

Table 12: Student performance (Other non-majors)

Course Element	No Quizzes ($N = 14$)	Pop Quizzes ($N = 41$)	P
Programs	86.8 ± 8.4	84.9 ± 5.0	0.352
Exam 1 (raw)	83.1 ± 5.1	79.8 ± 4.9	0.177
Exam 2 (raw)	66.3 ± 7.3	70.1 ± 5.0	0.205
Exam 3 (raw)	53.7 ± 7.6	58.8 ± 5.6	0.150
Semester grade	80.4 ± 5.2	78.7 ± 4.1	0.311

Table 13: Earned letter grades with and without the use of "pop" quizzes. Shows percentage of students earning: an A, at least a B, at least a C, at least a D, an F, and percentage of withdraws. Table is organized by academic year and by major.

ALL	A	≥B	≥C	≥D	F	W
Pop Quiz	33.6%	60.4%	79.2%	87.3%	5.4%	7.4%
No Quiz	25.7%	58.6%	78.6%	84.3%	4.3%	11.4%

(a) all 219 students in the study

FR & SO	A	≥B	≥C	≥D	F	W
Pop Quiz	30.0%	53.0%	72.0%	84.0%	8.0%	8.0%
No Quiz	24.4%	57.8%	77.8%	82.2%	4.4%	13.3%

(b) the 145 underclassmen (freshmen and sophomores)

JR & SR	A	≥B	≥C	≥D	F	W
Pop Quiz	40.8%	75.5%	93.9%	93.9%	0.0%	6.1%
No Quiz	28.0%	60.0%	80.0%	88.0%	4.0%	8.0%

(c) the 74 upperclassmen (juniors and seniors)

CS/IS	A	≥B	≥C	≥D	F	W
Pop Quiz	34.8%	63.0%	80.4%	84.8%	6.5%	8.7%
No Quiz	12.5%	58.3%	75.0%	79.2%	8.3%	12.5%

(d) the 46 CS/IS majors

Math	A	≥B	≥C	≥D	F	W
Pop Quiz	36.8%	68.4%	87.7%	94.7%	1.8%	3.5%
No Quiz	34.5%	62.1%	86.2%	89.7%	3.5%	6.9%

(e) the 57 Math majors

Other	A	≥B	≥C	≥D	F	W
Pop Quiz	28.3%	47.8%	67.4%	80.4%	8.7%	10.9%
No Quiz	29.4%	52.9%	70.6%	82.4%	0.0%	17.7%

(f) the 46 students with majors other than CS/IS and Math

and other non-majors as seen in the Table. For some groups, the reduction in the rate of withdraws is greater than for others. For example, the rate of course withdraws for Math majors was cut in half by the introduction of "pop" quizzes. The failure rate for Math majors was also cut in half.

"Pop" quizzes have also increased the rate of students who pass CS-1. The increase is larger for some groups than for others. For example, upperclassmen seem to benefit much more from "pop" quizzes than do underclassmen. Table 13(c) shows that nearly 94% of juniors and seniors completed and passed CS-1 when "pop" quizzes are used compared to 88% when they are not used. Furthermore, every single junior and senior in the "pop" quiz sections of the course, who remained in the course until the end passed with a grade of at least a C. No juniors or seniors, in the "pop" quiz sections, failed or received grades less than C.

Nearly 95% of Math majors continued in the course until the end and passed when "pop" quizzes were used, compared to just under 90% before the introduction of quizzes (Table 13(e)). The rate of CS/IS majors completing the course and passing it also increased from approximately 79% to around 85% (Table 13(d)). However, for CS/IS majors, the key number to consider here is the rate at which students passed with at least a C since we require all of our CS/IS majors to pass each of the core CS/IS courses with a minimum grade of C. This rate increased from around 75% before the "pop" quizzes to 80% after "pop" quizzes were introduced.

More students are able to excel in the course when "pop" quizzes are used. Before the introduction of "pop" quizzes, approximately a quarter of the students in CS-1 received grades of A. This rate increased to around a third of the students in the "pop" quiz sections of CS-1. This increase in the rate of A grades was seen for every group except for students majoring in something other than CS/IS and Math. The biggest increase in the rate of A grades was seen for CS/IS majors. Prior to the introduction of "pop" quizzes, only 12.5% of CS/IS majors received grades of A in CS-1. "Pop" quizzes helped increase the percent of CS/IS majors who excel in CS-1 (grade of A) to nearly 35%. The increase in semester grades has occurred even with the elimination of exam curves, in favor of "pop" quizzes, while retaining the difficulty level of the exams themselves.

5. CONCLUSIONS

In this paper, we have seen that "pop" quizzes can serve an important role in CS-1. They provide a mechanism for continual feedback on performance both to the students as well as to the instructor of the course. Additionally, they help in improving student performance on exams and programming assignments. This improved performance appears most significant for CS/IS majors than for Math majors and for other majors. Our results do show that Math majors performance is enhanced with the use of "pop" quizzes, though less significantly, possibly due to Math majors (at least at our institution) performing at a level above others in the course to begin with, or perhaps due to the large percentage of Math majors that wait until their junior year to take CS-1 after developing stronger study skills. Juniors and seniors also seem to benefit from the use of "pop" quizzes more so than freshmen and sophomores. This study has provided evidence that students' learning is enhanced through the continual feedback mechanism that "pop" quizzes offer both the student and the instructor.

6. REFERENCES

[1] G. Braught, L. M. Eby, and T. Wahls. The effects of pair programming on individual programming skill. *SIGCSE Bulletin*, 40(1):200–204, 2008.

[2] D. T. Dadamo. The correlation quiz: an aid in curbing cheating in programming assignments. *ACM SIGCSE Bulletin*, 22(2):52–54, 1990.

[3] C. Horstmann. *Java Concepts*. Wiley, 5th edition, 2008.

[4] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald. Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8):90–95, 2006.

[5] J. Paxton. Live programming as a lecture technique. *Journal of Computing Sciences in Colleges*, 18(2):51–56, 2002.

[6] D. Woit and D. Mason. Enhancing student learning through on-line quizzes. *ACM SIGCSE Bulletin*, 32(1):367–371, 2000.