SLO Project – AP Computer Science Principles

Anthony Pieper

UMUC – EDTP 650

Dr. Pierczynski

December 3, 2017

Introduction

AP Computer Science Principles is a relatively new exam compared to other AP exams released by the College Board. Using feedback from the previous year, which was the first time it was offered, the College Board recently updated the exam content and scoring rubric (The College Board, n.d.). This can lead to a potential problem in certain classrooms because if teachers don't update their lessons to assess students based on the updated exam requirements, the students will be unprepared and unable to pass the AP Computer Science Principles exam. As a means to mitigate this risk, this Student Learning Objectives (SLO) project will exam the use of certain learning strategies that can be used within an AP Computer Science Principles classroom to guide the students and teacher to ensure that all learners are getting the most benefit from classroom instruction.

Conceptualization

This SLO project was carried out at a Magnet School located in Baltimore County, which specializes in getting students prepared for colleges or careers after high school. Despite 35% of the school population qualifying for free or reduced lunch, and a similar amount of the population having Individualized Education Programs (IEP) or 504 plans, the graduation rate for the school has remained at or above 90% for at least the past three years. There is heavy emphasis on Advanced Placement (AP) and vocational courses in the school, and the school's overall School Improvement Plan (SIP) involves ensuring that every student enrolled in an AP class is capable of taking and passing the AP exam for that specific class.

The population of this study was the 10th grade AP Computer Science Principles class, which is composed of 22 students from diverse backgrounds and is a good reflection of the overall diversity of the school. The baseline data collected was the Unit 1 assessment which was

given in a similar manner to how the students will take the AP exam. The assessment consisted of a multiple-choice exam followed by a performance task that required the students to perform critical thinking and analysis. 11 students scored 68% or lower on the multiple-choice exam (with 52% being the lowest score), and 7 students scored 60% or lower on the performance task (with 50% being the lowest score). Looking at student performance for both parts as a whole:

- 7 students scored 70% or better on both parts of the exam
- 12 students scored 68% or less on one part of the exam
- 3 students scored 68% or less on both parts of the exam

Throughout the AP Computer Science Principles course, students are required to take similar assessments at the end of every unit as a way of preparing them for the AP exam. Based on the results of the Unit 1 assessment, two growth targets were set and evaluated on the Unit 3 assessment to ensure success on the AP exam. A Growth to Mastery target of 90% was set on the performance task and a Common Growth target of 20% was set on the multiple-choice exam.

The period of the SLO project ran throughout Units 2 and 3 of the AP Computer Science Principles course, which took approximately 6 weeks to complete. The topics students learned during this period aligned to the following curriculum standards:

- CSTA K-12 Computer Science Standards – Networks & the Internet
- CSTA K-12 Computer Science Standards – Data & Analysis

Literature Review

As the two pieces to the AP Computer Science Principles assessments test different levels of student knowledge, it was best to go back to the basics and research Bloom's taxonomy, but with an emphasis on its relationship with Computer Science. The question was, what

instructional methods have been shown to have a direct impact on learning in courses related to computer science?

Using formative quizzes as a means of stimulating deeper knowledge can be quite useful when it comes to questions related to computer science. Cox and Clark (1998) and Machanick (2007) both discuss how careful wording of questions on programming quizzes can test students at multiple levels of Bloom's taxonomy, as well as provide rapid feedback on areas that need improvement. Questions that require students to evaluate whether or not a particular piece of code will function properly, along with carefully chosen distractor questions, can stimulate higher levels of knowledge. These styles of questions can also help students practice for the multiple-choice exam portion of the AP Computer Science Principles assessment, which can build confidence and hopefully lead to a higher score.

Adjusting the particular medium that students use for taking notes can also have an impact, possibly because it gives students more freedom to take what was presented and create something that is their own. Yen, Lee, and Chen (2012) compared students using image-based concept mapping to students who just used text-based concept mapping, and found that although there was no noticeable difference in the groups when it came to a summative assessment, the group that used image-based concept maps seemed to display more in-depth knowledge when it came to creating artifacts that emphasized their knowledge. This holds particular value for students in AP Computer Science Principles because the performance task requires students to create artifacts as part of the assessment.

Student debate can also stimulate deeper learning through the stimulation of student emotions, especially if the topic is important to the student. Jagger (2013) looked at the affective domain and its relationship with the cognitive domain, showing examples of using classroom

debates and role-play to stimulate student emotions, improving engagement and also increasing cognitive development in the subject being debated. The debate examples also involved topics specifically related to Computer Science, so it would be very easy to work similar debate topics into AP Computer Science Principles lessons.

Ensuring that teachers have the resources to support gaps in content knowledge is also important in ensuring deeper learning is continuously stimulated. Goel and Sharda (2004), Mathumbu, Rauscher, and Braun (2014), and Wankhede, Gandhi, and Kiwelekar (2016) all point to repeated cases of teachers in Computer Science and Engineering courses focusing on the more basic levels of Bloom's taxonomy, indicating that these teachers possibly didn't start out teaching in those fields, and lack knowledge in that subject, so they are unable to assess students at higher levels. Fortunately, Kausar, Choudhry, and Gujjar (2008) indicate there is an increasing push to include Computer Aided Instruction (CAI) in many new technology courses, which can provide much-needed support to overwhelmed teachers. CAI also has the added benefit of providing instant feedback to the student, so it can function as an effective formative assessment similar to the quizzes mentioned by Cox and Clark (1998).

Procedures

The data that formed the baseline for this SLO project came from the Unit 1 assessments that occurred at the end of the first month of the AP Computer Science Principles class, and the actual SLO project itself took place during Unit 2 and Unit 3, which covered a period of approximately 6 weeks. There was a multiple-choice unit exam and performance task at the end of week 3, and another at the end of week 6. Following the recommendations by Cox and Clark (1998) and Machanick (2007) in the literature review, formative assessments called *Quick Quizzes* were administered as lesson warm-ups in week 2 and week 5. These were meant to

provide students with an idea of the types of questions they would see on the multiple-choice exams, as well as provide feedback to the teacher on what gaps were present in student knowledge. And for the multiple-choice exams that were administered in weeks 3 and 6, many of the questions were rewritten to ensure they aligned with the lesson content that was taught, as well following the recommendations of Wankhede, Gandhi, and Kiwelekar (2016) to ensure that they tested students at higher-levels.

To help students prepare for the performance tasks in weeks 3 and 6, clear objectives for those assessments were given to students in weeks 1 and 4. Following the recommendation of Mathumbu, Rauscher, and Braun (2014), each lesson leading up to the performance tasks had higher-level learning objectives written on the whiteboard, and students were reminded of how those learning objectives related to the performance task objectives. To reinforce the learning objectives, students were directed to use a specific Computer Aided Instruction (CAI) system at home to complete homework assignments, which follows the recommendations of Kausar, Choudhry, and Gujjar (2008). This specific CAI (https://studio.code.org) maps to the curriculum of the AP Computer Science Principles course, and is also continuously being updated to reflect changes in the AP exam (Code.org, n.d.). Any questions that the students had regarding the CAI or the results of the homework was addressed by the teacher in class the following lesson.

Results

Data was collected from the *Quick Quizzes* in weeks 2 and 4, as well as the multiple-choice and performance tasks completed in weeks 3 and 6. Compared to the baseline data collected from the Unit 1 multiple-choice exam and performance task, the goal was to have an improvement of 20% per student on the Unit 3 multiple-choice exam, and an overall score of 90% for all students on the Unit 3 performance task. Based on the data collected, these goals

were not met. Only 8 students achieved the Common Growth target of 20% increase on the multiple-choice exam, and only 12 students achieved the Growth to Mastery target of 90% or higher on the performance task. However, even though the goals were not met, there was still evidence of significant learning occurring:

- 15 students improved their multiple-choice exam scores, and 15 students improved their performance task scores, with 12 students improving both
- Students who improved their multiple-choice exams scores averaged 19% improvement, and students who improved their performance task scores averaged 25% improvement
- Only 2 students scored less than 70% on the multiple-choice exam, with none scoring less than 60%

See Table 1 in Appendix A for the specific data that was collected.

Implications

Formative assessments used as a means of identifying gaps was a crucial part of this SLO project. There was a specific instance in Unit 2 where students didn't perform well on the *Quick Quiz*, and as a result it was identified that many students were failing to make conceptual connections between what was taught in lesson activities and what was going to be assessed on the Unit 2 exam. Handouts were created to fill in some of the conceptual gaps, and part of a lesson prior to the exam was used to ensure that students really did understand the concepts that were going to be covered on the exam. This can be seen in Table 1 in Appendix A, as there is a reduction from 8 to 4 students who scored less than 70% on the *Quick Quiz* compared to the Unit 2 multiple-choice exam. There is also a similar drop in the number of students who scored less then 80% on the second *Quick Quiz* compared to the Unit 3 multiple-choice exam.

Clear lesson objectives and how they relate to assessments also was critical to this SLO project, and is probably the main reason that there were no failures on the Unit 2 or Unit 3 performance tasks compared to the 6 failures on the Unit 1 performance task. The main objective of the performance tasks for Unit 2 and Unit 3 were reiterated throughout the lessons in those units, and every time a new example was used, the students were reminded of the performance task objectives, and how they related to the current lesson's objectives. This allowed the students to be exposed to several different examples of the concept before they were assessed on their performance, providing them multiple ways of demonstrating mastery of the objective. One caveat to the results of the performance tasks is that there were 3 students who failed to submit the Unit 2 performance task, and 4 students how failed to submit the Unit 3 performance task, which could have potentially skewed some of the data for the study.

Although this study failed to achieve its goals, one thing that could potentially improve results in the future is spreading the unit exams out further, at least 4 weeks or longer, as opposed to the 3-week spacing that occurred in this study. The formative assessments in the week prior provided some valuable insights, but there wasn't time to tailor or adjust lessons in-between the assessments and the unit exams. This may explain why there was improvement shown in most students, but not all. If there was an additional week between the assessments and unit exams, this could allow more time to differentiate instruction to allow all students to increase their performance.

References

Code.org (n.d.). Computer Science Principles. Retrieved from https://code.org/educate/csp

Cox, K., & Clark, D. (1998). The use of formative quizzes for deep learning. *Computers & Education*, *30*(3), 157-167. Retrieved from https://doi-org.ezproxy.umuc.edu/10.1016/S0360-1315(97)00054-7

Goel, S., & Sharda, N. (2004, September 27). What do engineers want? Examining engineering education through Bloom's taxonomy. Presentation from AAEE, *The 15th Annual Conference for the Australasian Association for Engineering Education*. Retrieved from http://files.eric.ed.gov/fulltext/ED524509.pdf

Jagger, S. (2013). Affective learning and the classroom debate. *Innovations in Education and Teaching International*, *50*(1), 38-50. Retrieved from http://eds.b.ebscohost.com.ezproxy.umuc.edu/eds/pdfviewer/pdfviewer?vid=0&sid=fb84a720-b5c7-4485-bd2b-01714f7cf8d6%40sessionmgr104

Kausar, T, Choudhry, B. N., & Gujjar, A. A. (2008). A comparative study to evaluate the effectiveness of computer assisted instruction (CAI) versus class room lecture (CRL) for computer science at ICS level. *Turkish Online Journal of Educational Technology*, *7*(4), 19-28. Retrieved from http://files.eric.ed.gov/fulltext/EJ1102933.pdf

Loria-Saenz, C. (2009). On requirements for programming exercises from an e-learning perspective. *Cornell University Library: ArXiv E-Prints.* Retrieved from https://arxiv.org/pdf/0903.0786

Machanick, P. (2007). Teaching Java backwards. *Computers & Education*, *48*(3), 396-408. Retrieved from https://doi-org.ezproxy.umuc.edu/10.1016/j.compedu.2005.01.009

Mathumbu, D., Rauscher, W., & Braun, M. (2014). Knowledge and cognitive process dimensions of technology teachers' lesson objectives. *South African Journal of Education, 34*(3), 1-9. http://dx.doi.org/10.15700/201409161053

The College Board (n.d.). AP computer science principles: The exam. *AP Central*. Retrived from https://apcentral.collegeboard.org/courses/ap-computer-science-principles/exam?course=ap-computer-science-principles

Wankhede, H. S., Gandhi, S. S., & Kiwelekar, A. W. (2016). On which skills do Indian universities evaluate software engineering students? *Cornell University Library: ArXiv E-Prints.* Retrieved from https://arxiv.org/pdf/1601.01796

Yen, J., Lee, C., & Chen, I. (2012). The effects of image-based concept mapping on the learning outcomes and cognitive processes of mobile learners. *British Journal of Educational Technology, 43*(2), 307-320. Retrieved from http://eds.b.ebscohost.com.ezproxy.umuc.edu/eds/pdfviewer/pdfviewer?vid=0&sid=7ef9cf19-d88d-4583-8ef0-b45678900afd%40sessionmgr102

Appendix A

Assessment Results

Table 1

Student	Unit 1 MC	Unit 1 PT	Quick Quiz	Unit 2 MC	Unit 2 PT	Quick Quiz	Unit 3 MC	Unit 3 PT	Common Growth - MC +20%	Growth to Mastery - PT 90%	PT 3 Improvement +20%
1	84%	80%	89%	80%	89%	71%	70%	91%	-14%	91%	11%
2	60%	70%	100%	80%	94%	86%	83%	91%	23%	91%	21%
3	84%	60%	33%	76%		71%	67%	73%	-17%	73%	13%
4	80%	70%	22%	68%	72%	57%	90%	91%	10%	91%	21%
5	100%	60%	89%	76%	83%	86%	80%		-20%		
6	64%	70%	89%	92%	89%	71%	93%	82%	29%	82%	12%
7	72%	50%	67%	72%	72%	100%	73%	73%	1%	73%	23%
8	84%	70%	100%	92%	94%	100%	80%	91%	-4%	91%	21%
9	76%	100%	67%	96%	100%	100%	100%	100%	24%	100%	0%
10	88%	70%	89%	84%	100%	100%	80%	91%	-8%	91%	21%
11	68%	60%	44%	64%		86%	87%	73%	19%	73%	13%
12	84%		100%	80%	89%	100%	83%	100%	-1%	100%	100%
13	60%	60%	67%	52%	83%	71%	83%	73%	23%	73%	13%
14	64%	80%	100%		94%	100%	90%		26%		
15	56%	60%	78%	60%	94%	100%	60%	91%	4%	91%	31%
16	52%	70%	67%	80%	100%	71%	97%	100%	45%	100%	30%
17	100%	100%	100%	96%	100%	100%	100%	100%	0%	100%	0%
18	68%	80%	100%	76%	89%	71%	70%		2%		
19	68%	100%	89%	92%	100%	100%	90%	100%	22%	100%	0%
20	60%	70%	89%	76%	89%	100%	90%	82%	30%	82%	12%
21	64%	70%	44%	84%	89%	71%	77%	100%	13%	100%	30%
22	76%	70%	78%	84%		86%	90%		14%		
80% +	7	3	12	8	17	14	13	14			
70%	3	9	2	5	2	7	4	4			
60% -	11	6	8	4	0	1	2	0			

Common Growth (MC +20%)	8
Growth to Mastery (PT 90%)	12
Both Targets	4
MC Improve	15
MC Worse	6
PT Improve	15
PT Worse	0
Both Improve	12
Average Growth* - MC	19%
Average Loss* - MC	-11%
Average Growth* - PT	25%

* only those that experienced gain or loss

Please note in the above table that improvements of 100% or 0% are indicated in orange to identify them as outliers. This results from either no change between assessments or complete change due to a student not participating in a previous assessment. Purple indicates that a student failed to participate in the assessment.