Dilsun Kaynar
Assistant Teaching Professor
Computer Science Department

TEACHING & LEARNING SUMMIT 2016

Norman Bier
Executive Director, Simon Initiative
Director, Open Learning Initiative (OLI)
Tom Cortina
Teaching Professor
Computer Science Department
SCS Assistant Dean for Undergraduate Education
Carnegie Mellon University

Teaching Principles of Computing with OLI

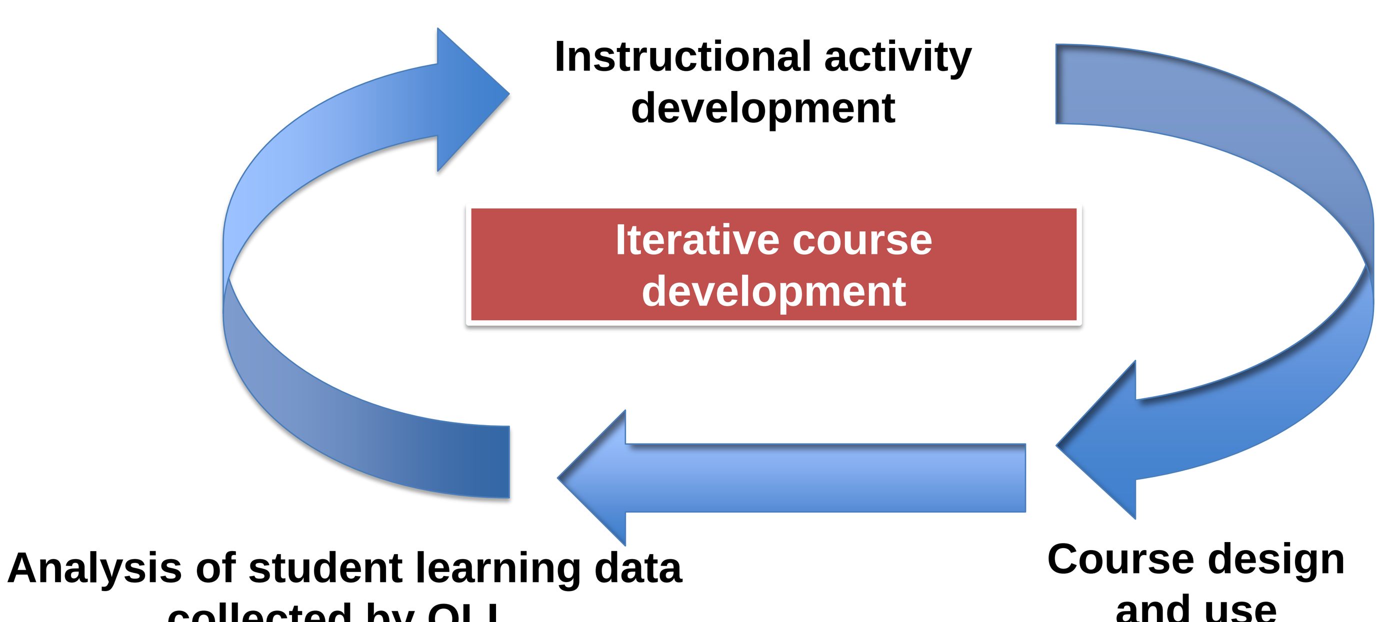
Project Goals

- Develop online course material where each instructional activity is tied to a measurable learning objective, assessment, and targeted feedback;
- Use the material in a blended teaching mode, collecting data through the OLI platform for iterative improvements to the content and delivery of the course material; and
- Measure student learning gain and identify the learning objectives that prove to be most difficult to achieve in introduction to programming.

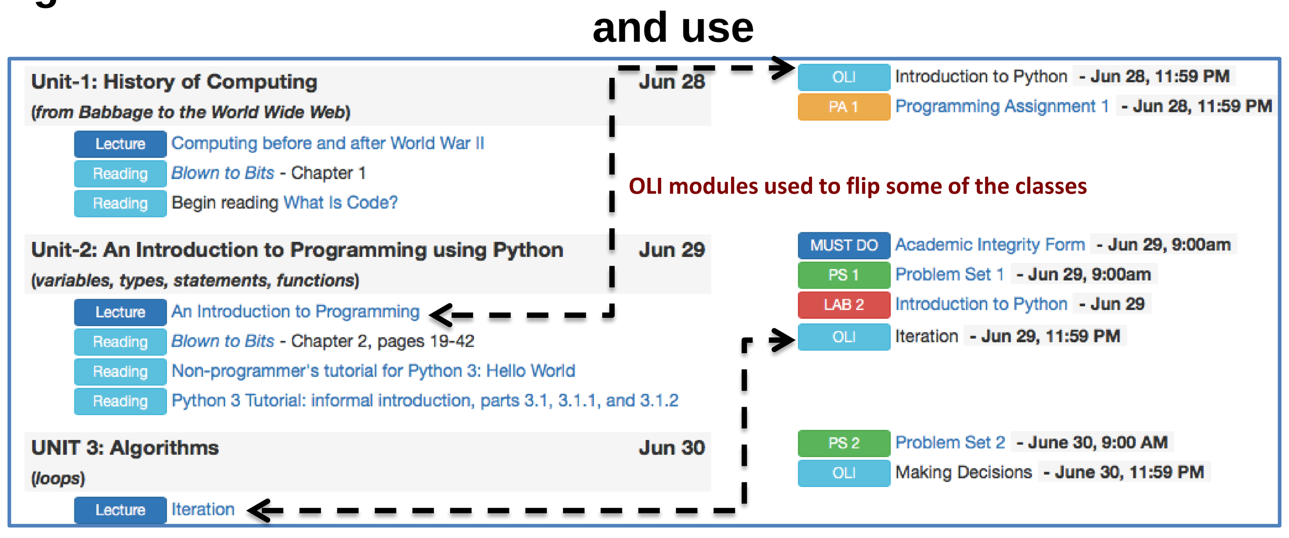
Project Design

Principles of Computing (15-110) is CMU's computer science course for non-majors that serves 200-300 students every semester. In Summer 2016, the course was taught using a blended teaching mode, which was the first time the course departed significantly from the traditional lecture model.

Modules available on OLI : Introduction to Programming with Python, Iteration, Making Decisions, Putting Iterations and Decisions Together, Recursion, Data Structures, Data Representation, Encryption, Computability, Cellular Automata. Modules with programming components include support for code execution and autograding for feedback.



Excerpt from Course Schedule for Summer 2016



Lessons Learned

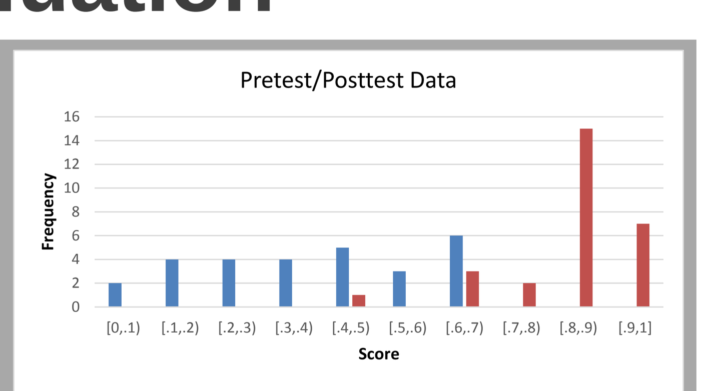
- As instructors developing course content, we found that the course development approach promoted by OLI guided us in a direction that let us combine our domain expertise with sound pedagogical principles.
- The effectiveness of the course delivered in a blended mode met our basic criterion of leading to significant student learning gain. A comparison to other teaching methods by controlled experiments remains as future work.
- Our course traditionally uses two kinds of exams: lab exams assess basic programming skills while regular exams assess conceptual understanding of the material in addition to programming skills. The correlation between Pretest and Exam 1 scores was expected since they assess similar skills. We expected Quiz 1 scores to have a correlation with Lab Exam 1 scores for the same reason. However, Quiz 1 data did not have enough variation to warrant further analysis. We are planning to redesign the quiz to improve the diagnostic quality of questions and the overall predictive quality of the quiz score.
- We identified the following concepts to be the most difficult for beginning programmers to grasp: (1) The difference between "an expression evaluating to a value" and "a command causing a side effect"; and (2) The concept of "calling a function" to obtain a value and using that value in the computation of another value. We recognize the need to redesign the related instructional activities in the next iteration of course development.

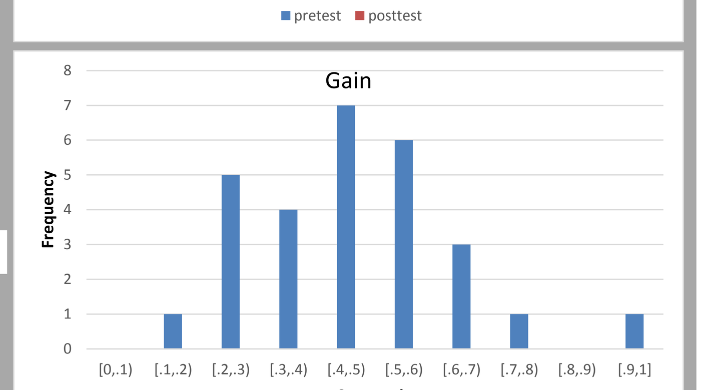
Project Evaluation

Summary Statistics

	pretest	posttest	gain
average	0.39	0.85	0.46
min	0.00	0.47	0.16
max	0.68	1.00	1.00
median	0.39	0.87	0.47
25th percentile	0.25	0.84	0.32
75th percentile	0.58	0.91	0.54

gain = posttest score – pretest score

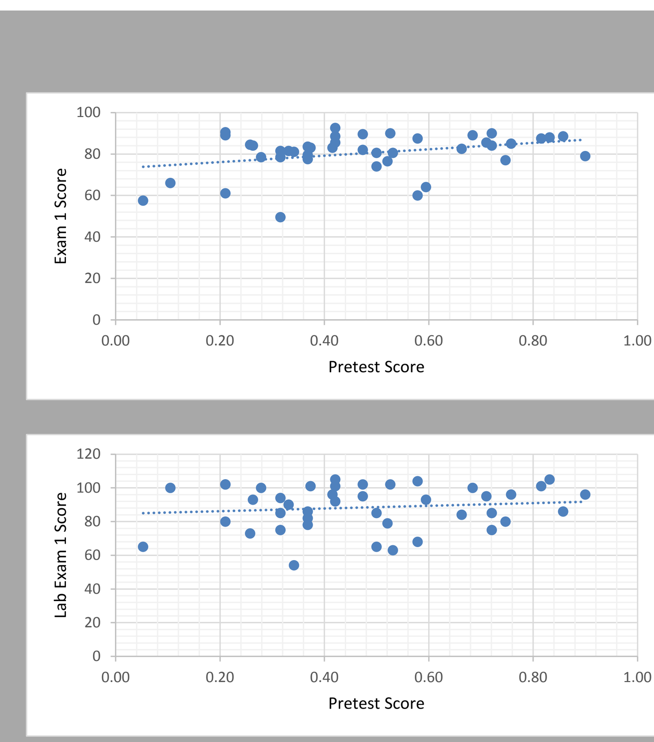




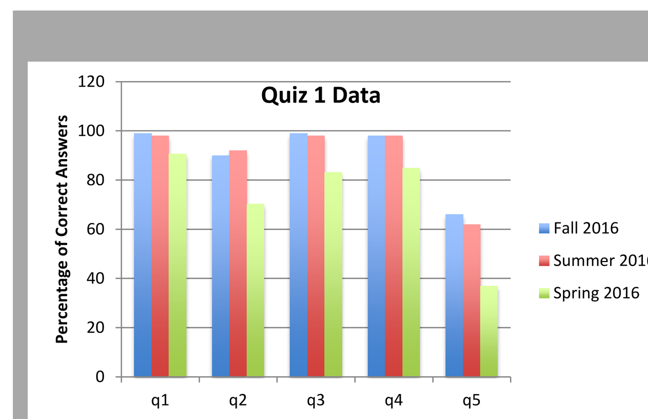
PRETEST/POSTTEST ANALYSIS. The test that was given before and after taking the course consisted of 19 questions. The Pretest/Posttest chart shows the number of students for each score interval . The Gain chart shows the number of students for per score gain interval. The learning gain was found to be statistically significant: one-sample $t_{(27)}=13$, $p< .001$. *(BASED ON SUMMER 2016 DATA)*

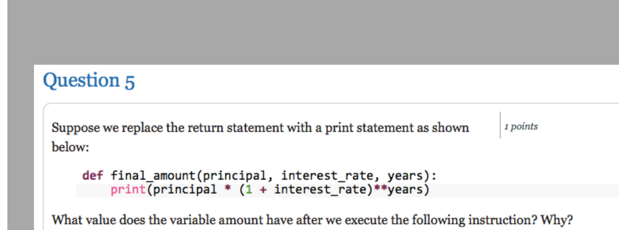
Pretest_Q#	DI	% correct
q01	0.36	0.50
q02	0.54	0.46
q03	0.46	0.73
q04	0.34	0.44
q05	0.31	0.13
q06	0.39	0.53
q07	0.29	0.64
q08	0.58	0.42
q09	0.61	0.53
q10	0.60	0.72
q11	0.48	0.50
q12	0.62	0.46
q13	0.34	0.41
q14	0.18	0.15
q15	0.38	0.65
q16	0.52	0.48
q17	0.17	0.54
q18	0.61	0.51
q19	0.62	0.28
min	0.17	0.13
max	0.62	0.73
average	0.44	0.48

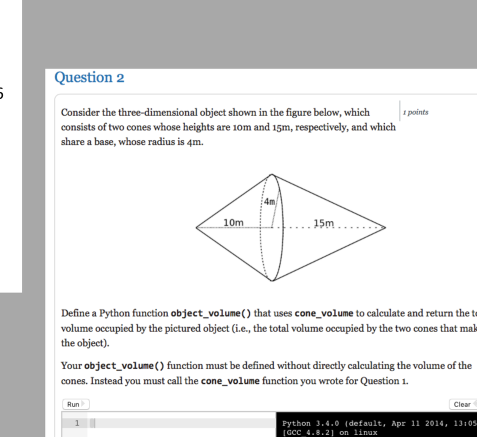
ITEM DISCRIMINATION FOR PRETEST. Discrimination Index (DI) , indicates Questions 2, 8, 9, 10, 12, 16, 18, 19 had a reasonable diagnostic quality whereas Questions 4 and 17 could be improved. Correctness rates indicate students had room to grow.



PRETEST AS A PREDICTOR OF EXAM PERFORMANCE. The Pretest was significantly correlated with the Exam 1 score (r=.33, p < .05) but not with the Lab Exam 1 score (r=.13, n.s.)
(BASED ON SUMMER 2016 DATA)







ITEM DIFFICULTY FOR QUIZ 1. This quiz is an assessment made after the first module on programming basics. Question 5 has the lowest correctness rate, followed by Question 4 . *(BASED ON SPRING-SUMMER-FALL 2016 DATA)*

Carnegie Mellon University
Computer Science Department

With support from:
Carnegie Corporation of New York
The Kresge Foundation
Eberly Center Teaching Excellence & Educational Innovation
Open Learning Initiative Transforming higher education through the science of learning.

Special thanks to
- Judy Brooks
- Lauren Herckis
- Kim Law
- Marsha Lovett