

# System clusters similar student programs together, so instructors can identify broad trends

March 30 2015, by Larry Hardesty

---



MIT graduate students Elena Glassman and Jeremy Scott. Credit: Jose-Luis Olivares/MIT

In computer-science classes, homework assignments consist of writing programs. It's easy to create automated tests that determine whether a given program yields the right outputs to a series of inputs. But those tests say nothing about whether the program code is clear or confusing,

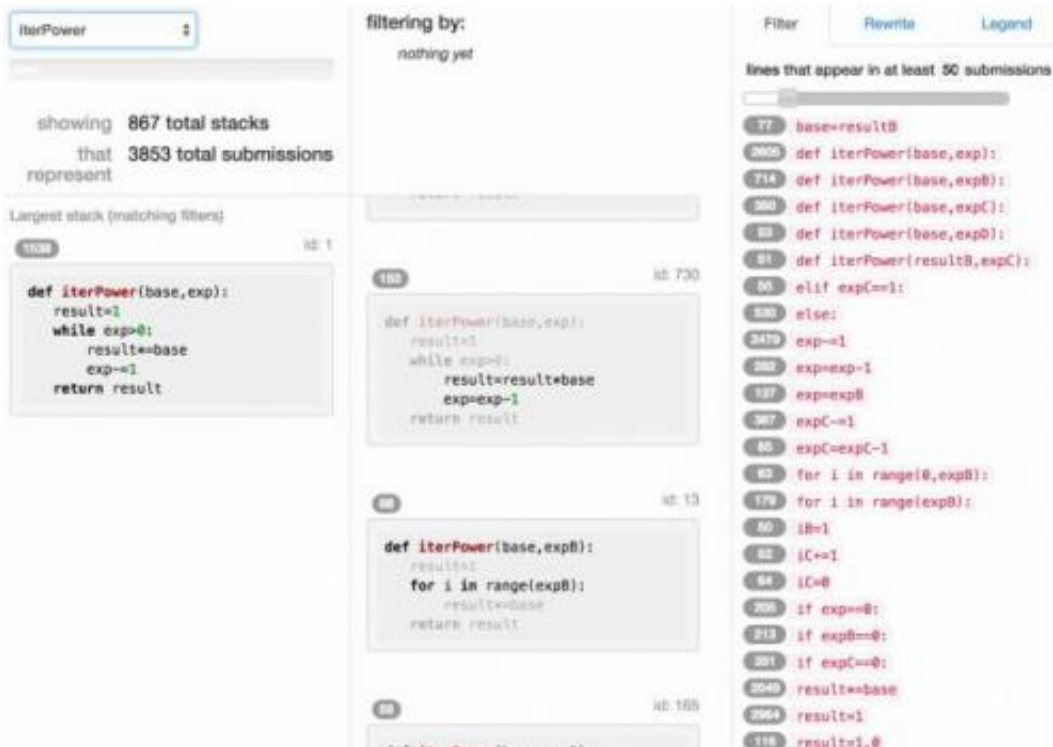
whether it includes unnecessary computation, and whether it meets the terms of the assignment.

Professors and teaching assistants review students' code to try to flag obvious mistakes, but even in undergraduate lecture courses, they usually don't have time for exhaustive analysis. And that problem is much worse in online courses, with thousands of students, each of whom might have approached a problem in a slightly different way.

In April, at the Association for Computing Machinery's Conference on Human Factors in Computing Systems, MIT researchers will present a new system that automatically compares students' solutions to programming assignments, lumping together those that use the same techniques.

For each approach, the system—called OverCode—creates a program template, using variable names that a preponderance of students happen to have converged on. It then displays templates side-by-side, graying out the code they share, so the differences stand out in relief. And from any template, instructors can, if they choose, pull up a list of the student programs that accord with it.

Instructors who notice variations across templates that make no difference in practice can also write rules establishing the equivalence of alternatives. In some instances, for example, " $y*x$ " might yield a different result than " $x*y$ ", but—depending on the ways in which  $x$  and  $y$  are defined—in other instances, it won't. When it doesn't, an instructor could further winnow down the number of templates by creating the rule " $y*x = x*y$ ".



A screenshot of the OverCode user interface. The top left panel shows the number of clusters, called stacks, and the total number of solutions visualized. The next panel down in the first column shows the largest stack; the second column shows the remaining stacks. The third column shows the lines of code occurring in the cleaned solutions of the stacks together with their frequencies.

The system could allow instructors of online courses to provide generalized feedback that addresses a broader swath of their students. But it could also provide information on how computer-science courses—both online and on campus—could be better designed.

With online courses, "in a few months, you can have many orders of magnitude of students go through the same material and find all the interesting alternative solutions or make the same errors," says Elena Glassman, an MIT graduate student in computer science and engineering and first author on the new paper. "Then it's taking all those records of

what people did and making sense of it so that when we run the course again, it's better, and when we run the course residentially, we're better able to handle the particular 200 students that we're meeting with on a regular basis."

Two programs that perform the same computation may have code that looks somewhat different. The programmers may have chosen different variable names—"total," say, in one case, versus "result" in the other. Subfunctions may be executed in different orders.

So in addition to comparing programs' code, OverCode observes the values that variables take on as the programs execute. Two programs with variables that take on the same values in the same order are judged to be identical.

In their new paper, Glassman and her collaborators—her thesis advisor, professor of computer science and engineering Rob Miller; her fellow graduate student Jeremy Scott; Rishabh Singh, who completed his PhD at MIT last year and is now at Microsoft Research; and Philip Guo, an assistant professor of [computer science](#) at the University of Rochester—also report the results of two usability studies that evaluated OverCode.

In the studies, 24 experienced programmers reviewed thousands of students' solutions to three introductory programming assignments, using both OverCode and a standard tool that displays solutions one at a time. For each assignment, the subjects were given 15 minutes to assess the strategies [students](#) most commonly used to design a particular function and to provide general feedback on each, complete with example code.

Remarkably, when assessing the simplest of the three assignments, the subjects analyzing raw code performed as well those using OverCode: In both cases, the five strategies they identified covered about half of the

student responses.

For the most difficult of the three assignments, however, the OverCode users covered about 45 percent of student responses, while the subjects analyzing raw data covered only about 9 percent. "The strategy starts to shine on more-complicated programs," Glassman says.

**More information:** "OverCode: Visualizing Variation in Student Solutions to Programming Problems at Scale."

[people.csail.mit.edu/elg/papers/glassman-tochi.pdf](http://people.csail.mit.edu/elg/papers/glassman-tochi.pdf)

*This story is republished courtesy of MIT News ([web.mit.edu/newsoffice/](http://web.mit.edu/newsoffice/)), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: System clusters similar student programs together, so instructors can identify broad trends (2015, March 30) retrieved 26 April 2024 from <https://phys.org/news/2015-03-clusters-similar-student-instructors-broad.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--