PCRS

PCRS is an application for bundling interactive programming exercises with video-based instruction that has been developed at the University of Toronto. Currently, PCRS supports linking to videos, program writing exercises with the option of starter code and fixed (unmodifiable) code, and multiple choice and short answer questions. The system is in use in courses at the University of Toronto, and some content packages are available upon request. The system is also an active research platform, with data from PCRS being used to explore novice programmer misconceptions in Python and C.

PCRS is open source, is available for download, and can be used and modified freely. While most contributors to date have been based at the University of Toronto, we welcome collaborations in development and research.

Available Domains

PCRS supports online exercises for Python, Java, C, R (in development), relational algebra, and SQL. A content package including programming problems is available for C, with videos hosted online for use outside of PCRS. Videos are also hosted online for Python.

PCRS Interface

PCRS is built around the idea of quests and challenges. Quests are collections of challenges, and each challenge is a collection of video and text resources with accompanying exercises.

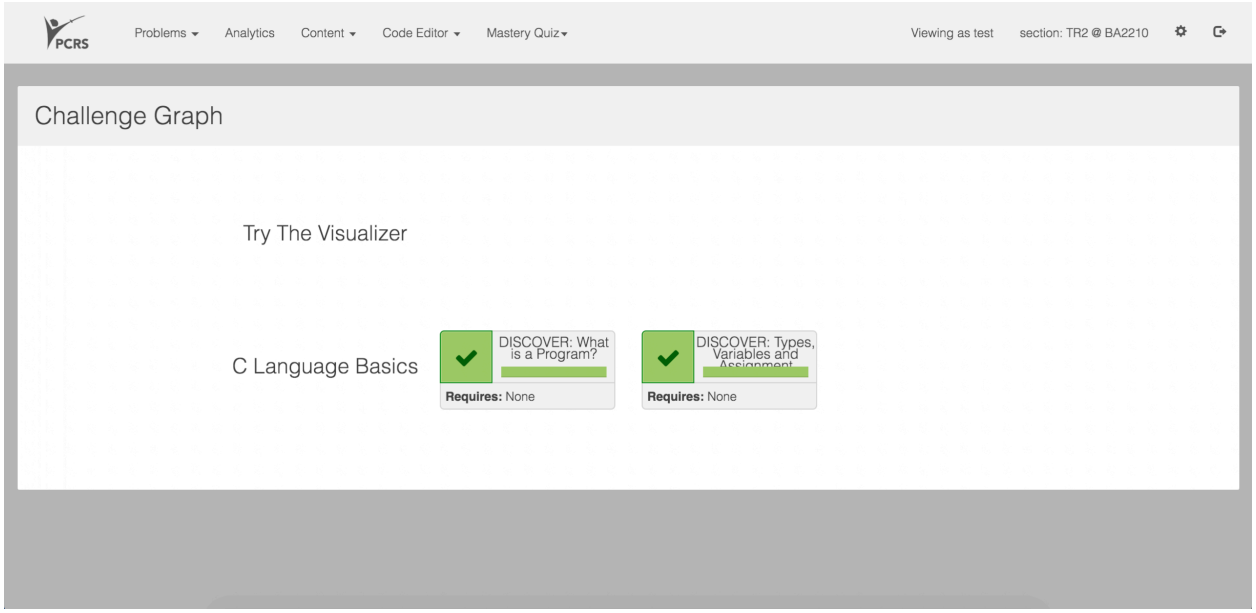

Figure: Each quest has its challenges listed by it. Each challenge has its own name, and the bar below shows your progress through the challenge. You can click on the challenge to enter it.

Inside a challenge, you'll see a progress bar in the top right and content on the main page. The progress bar contains icons -- for videos and exercises -- and these icons are red if you have not attempted the problem, yellow if you have attempted but not completed the problem, and green if you have solved the problem. Clicking on an icon takes you directly to the problem it represents.

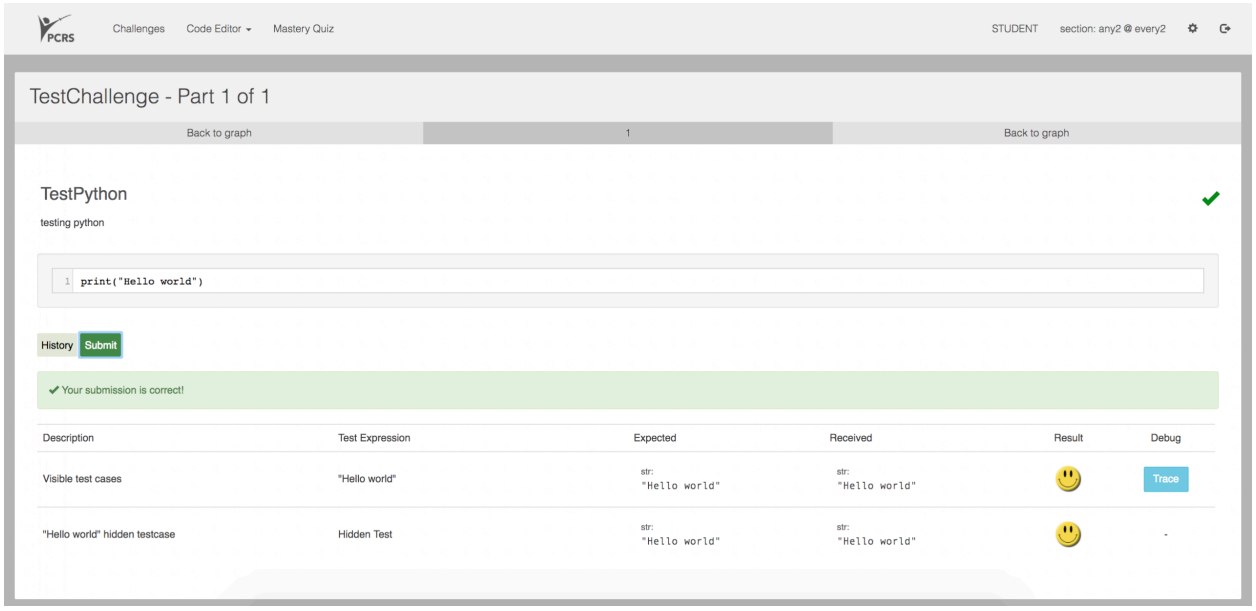

Figure: A programming exercise that has been completed.

While a challenge may contain multiple items -- videos, multiple choice problems, or programming exercises -- each item can be completed separately. When a submission is made to an exercise, feedback -- often compiler or interpreter output, but also output from a static analysis tool (for Python) or more customized feedback -- is presented in a ribbon, and for programming exercises, the results of unit tests are displayed. A check mark appears on the problems that a student has solved.

Instructors can create programming exercises within PCRS by providing a name and description of a problem and then defining starter code and tests. The contents of the "Starter Code" field are displayed to the student when they view the problem, and the student adds to the field containing this code and then submits their solution for testing using the tests provided by the instructor.

While the code the student submits must be complete (executable), instructors can control what code students see. Tags are provided to set the visibility and editability of code. “Student_code” tags define a section where students can view and insert code freely. “Blocked” tags define code that is visible but not editable; this is useful when a particular solution is desired or when the instructor wishes a student to insert small sections of code into a larger block. Finally, the

“hidden” tags can be used to make code invisible, to reduce the amount a student must interpret or to hide code used for testing.

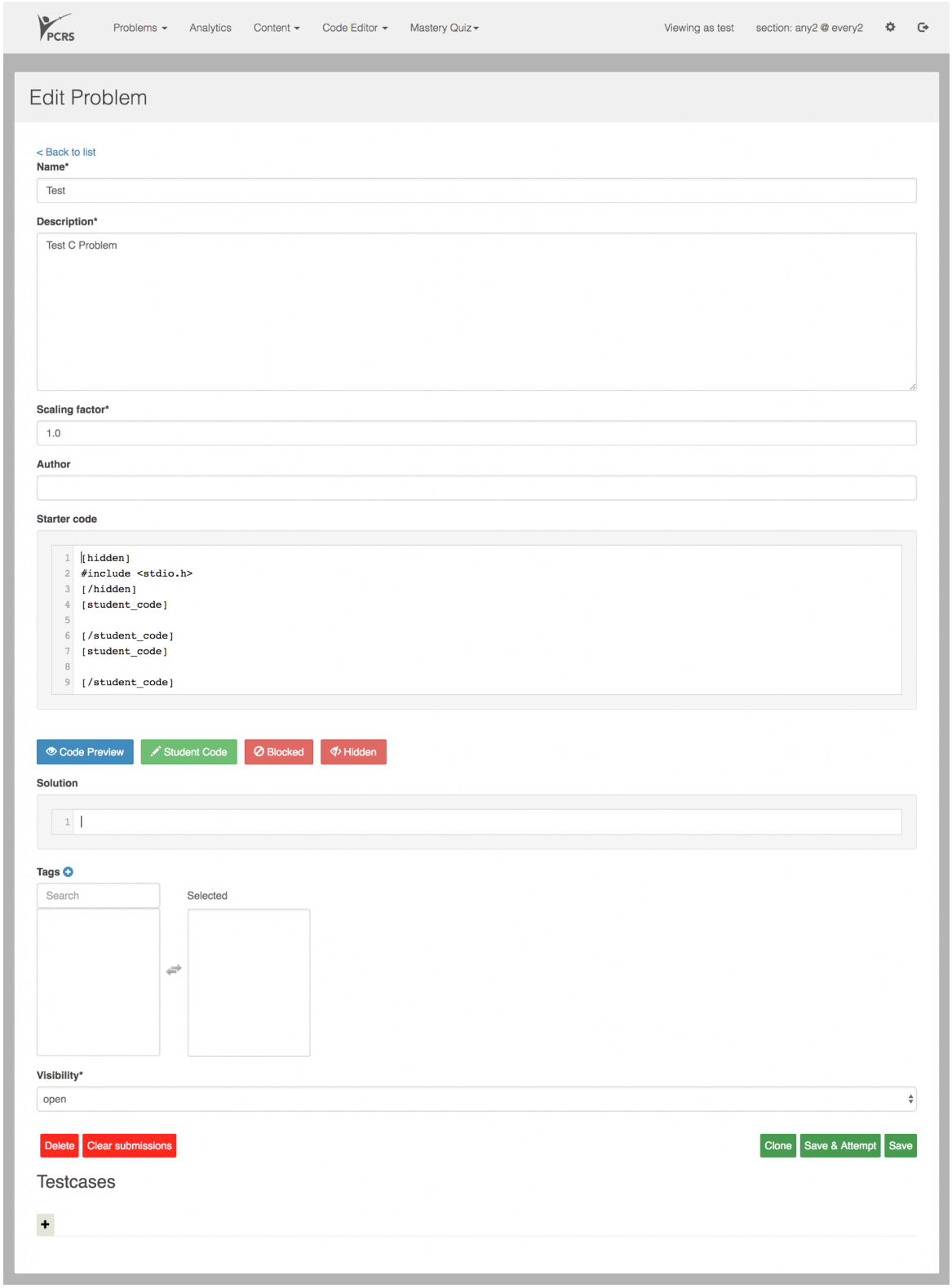

Figure: The problem creation interface, including example tags in the starter code window to define user editable and hidden code sections.

In addition to programming exercises, both multiple choice and short answer options are available. Both are also marked automatically, with the instructor defining correct answers for multiple choice and providing sets of required words or phrases for short answer.

PCRS Publications

Daniel Marchena Parreira, Andrew Petersen and Michelle Craig. "PCRS-C: Helping Students Learn C." Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (2015): 347.

Daniel Zingaro, Yuliya Cherenkova, Olessia Karpova and Andrew Petersen. "Facilitating code-writing in PI classes." *The 44th ACM Technical Symposium on Computer Science Education, SIGCSE '13, Denver, CO, USA, March 6-9, 2013* (2013): 585-590.

Links

https://mcs.utm.utoronto.ca/~pcrs/pcrs/ - user guides
https://bitbucket.org/utmandrew/pcrs/src/master/ - source repository