Praxly

An Online IDE for the Praxis CS Test Pseudocode

Benjamin Saupp
James Madison University

Ellona Macmillan
James Madison University

Chris Mayfield
James Madison University

Chris Johnson
James Madison University

Michael Stewart
James Madison University

Shelita Hodges
Richmond Public Schools

Praxis Prep Course

- Taught online by CodeVA
- By teachers, for teachers
- Synchronous sessions
- Asynchronous modules
- About 4 months long

Research Study

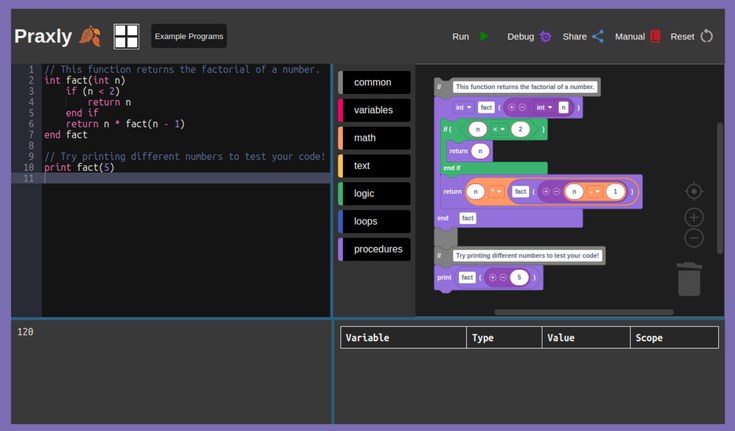
CS for CTE

- Fall 2023 cohort: 14 teachers with little to no use of Praxly
- Fall 2024 cohort: 12 teachers with extensive use of Praxly
- Analyzing surveys, interviews, and pre/post test scores

Try Praxly Today!

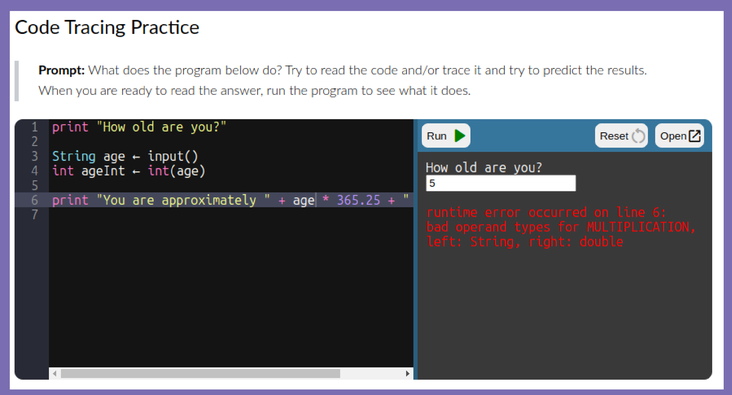
How do **high school** teachers prepare for a **standardized test** in a **language** that doesn't exist?

Praxly turns the Praxis pseudocode into real programs that can be run and debugged.



- Text Editor and Block Editor
- Bidirectional synchronization
- Easy-to-use step debugger
- Share source code via URL

- Embed in LMS via `<iframe>`
- Customizable user interface
- Additional language features
- Free and open source



Praxly makes the pseudocode more concrete and hands-on!

To learn more about Praxly, visit praxly.cs.jmu.edu.
To learn more about CodeVA's Praxis Prep course, visit codevirginia.org.
This work was supported by National Science Foundation award 2219770.

Praxis CS Test

- Pathway for K-12 CS endorsement
- Offered by Educational Testing Service
- 3 hours, 100 multiple-choice questions
- Many questions based on *pseudocode*

	Content Categories	Approx # of Questions
I.	Impacts of Computing	15
II.	Algorithms and Computational Thinking	25
III.	Programming	30
IV.	Data	15
V.	Computing Systems and Networks	15

Sample Question

Consider the following pseudocode procedure, which sorts an integer array `arr` of length `len`. The first element of `arr` is at index `0`. A call `swap ( arr, i , j )` swaps the values of `arr[i]` and `arr[j]`.

```
void sort ( int[] arr, int len )
    int pos ← 0
    while ( pos < len )
        if ( pos == 0 )
            pos ← pos + 1
        else
            if ( arr[pos] > arr[pos - 1] )
                pos ← pos + 1
            else
                swap ( arr, pos, pos - 1 )
                pos ← pos - 1
            end if
        end if
    end while
end sort
```

If `arr` originally contains the values `{2, 1, 5, 3, 4}`, what will the values in `arr` be after 6 iterations of the **`while`** loop?

(A) `{1, 2, 3, 4, 5}`
(B) `{1, 2, 3, 5, 4}`
(C) `{1, 2, 5, 3, 4}`
(D) `{2, 1, 5, 3, 4}`

Source: ets.org/pdfs/praxis/5652.pdf