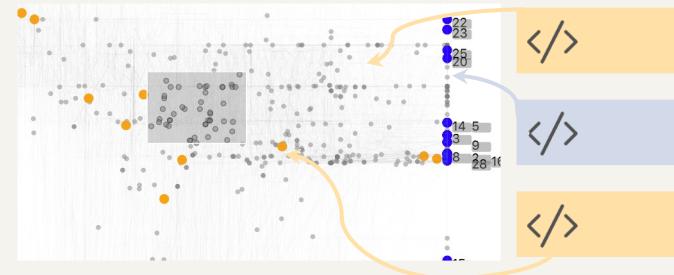


# VizProg: Identifying Misunderstandings by Visualizing Students' Coding Progress

Ashley Zhang<sup>1</sup>, Yan Chen<sup>2</sup>, Steve Oney<sup>1</sup>

<sup>1</sup> University of Michigan

<sup>2</sup> Virginia Tech



## Introductory programming courses



In-class programming  
exercises

## Introductory programming courses



In-class programming  
exercises



Practice coding skills

## Introductory programming courses



In-class programming  
exercises



Practice coding skills



Reinforce concepts

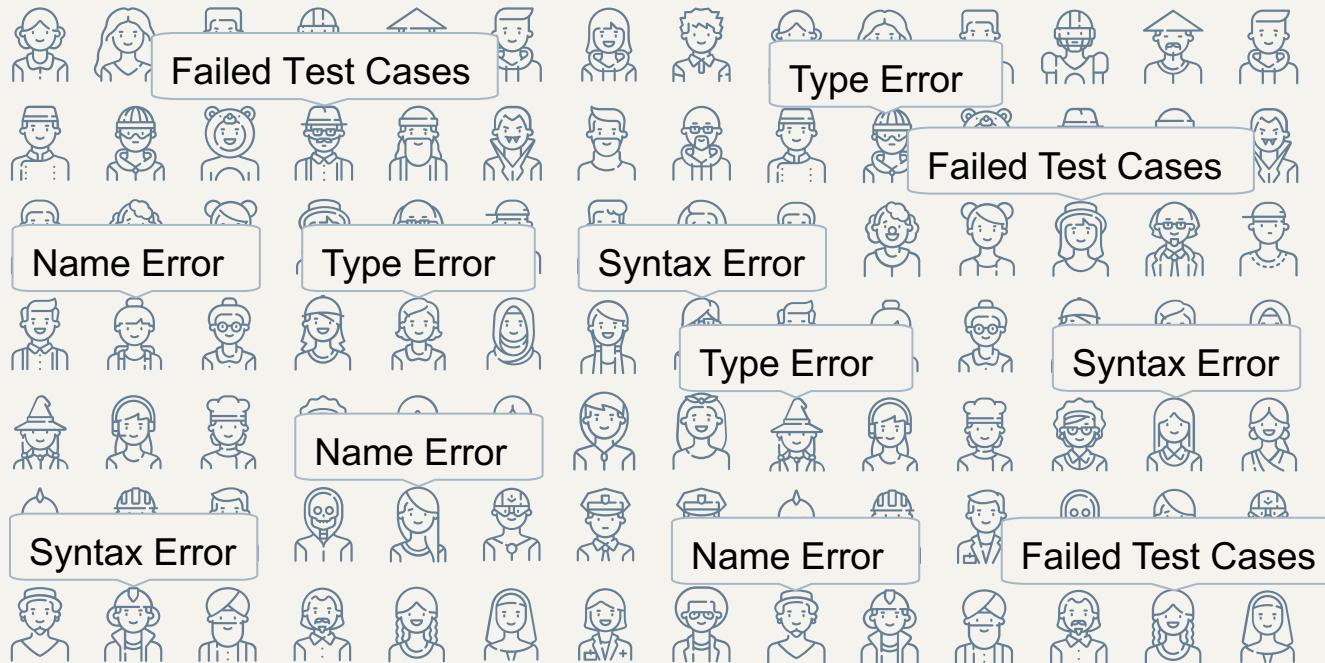
## Introductory programming courses



Monitoring students' progress is challenging

# Motivation

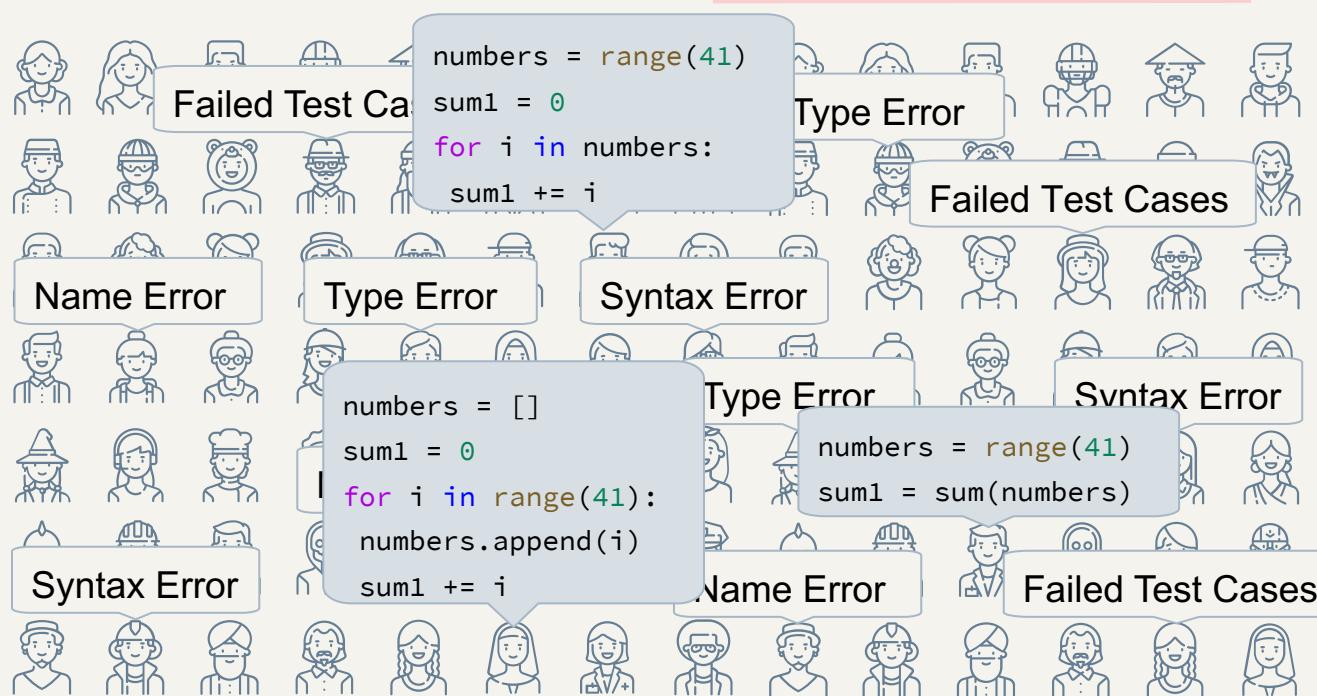
Exercise: Create a list from 0-40 and sum the list



**Misconceptions**  
Implicit and abstract

# Motivation

Exercise: Create a list from 0-40 and sum the list

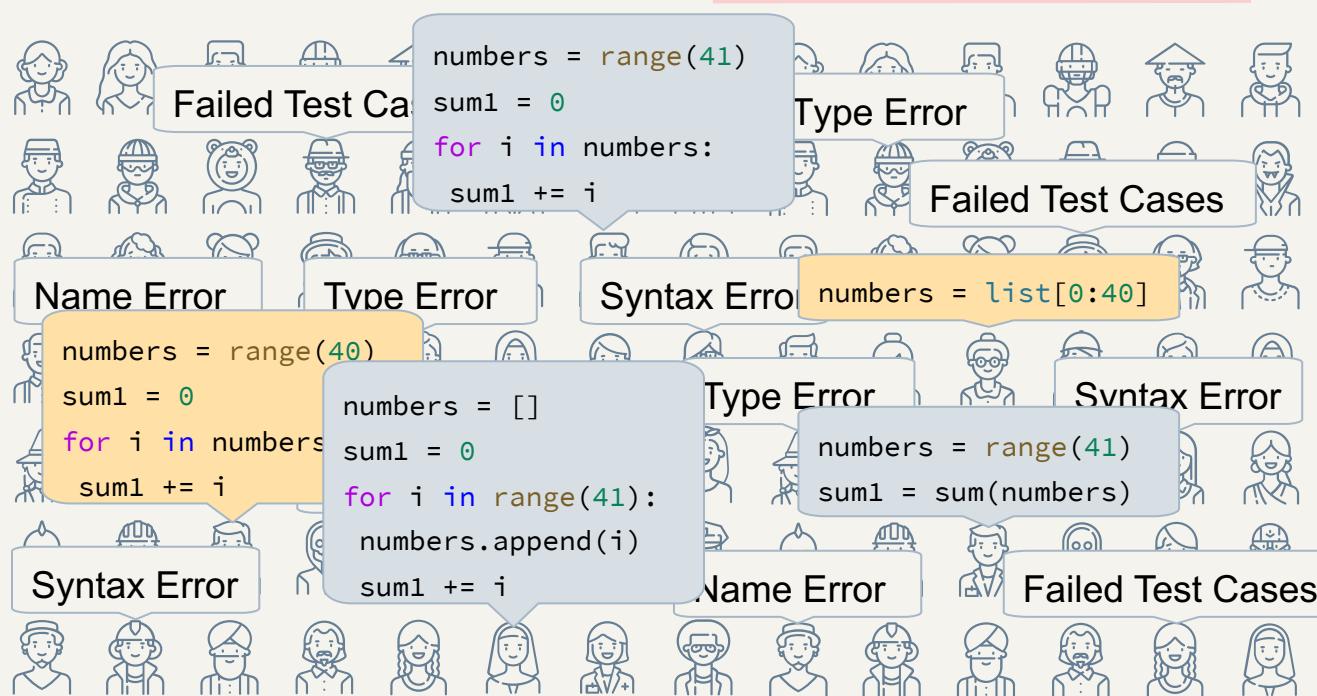


➊ Misconceptions  
Implicit and abstract

➋ Approaches  
Wide variation among solutions

# Motivation

Exercise: Create a list from 0-40 and sum the list



○ **Misconceptions**  
Implicit and abstract

○ **Approaches**  
Wide variation among solutions

○ **Progress**  
Different stages of solving a problem

# CS Instructor's Pain Point

(Mirhosseini et al., SIGCSE 23)



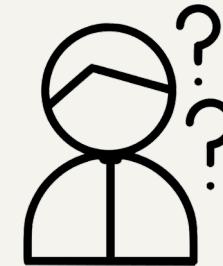
## Large amount of repetitive questions

Instructors frequently spend time on repetitive tasks, including answering questions and preparing quizzes.



## Scaling support

Instructors lack TAs or TAs are limited in what they are allowed to do.



## Surface student struggles

Instructors have low visibility into the student struggles

# Motivation

## OverCode

Glassman et al., TOCHI 15

The screenshot shows the OverCode interface with the following sections:

- Showing stacks**: 28 correct, 39 total.
- representing submissions**: 101 correct, 112 total.
- filtering by**: A slider for "lines that appear in at least 50 submissions".
- largest stack (matching filters)**: 28 lines.

```
def solution():
    numbers=range(41)
    sum1=0
    for num in numbers:
        sum1+=num
    return sum1
```
- remaining stacks (matching filters)**: 12 lines.

```
def solution():
    numbers=range(0,41)
    sum1=0
    for num in numbers:
        sum1+=num
    return sum1
```
- 11 lines**.

```
def solution():
    numbers=list(range(41))
    sum1=0
    for num in numbers:
        sum1+=num
    return sum1
```

- ✗ Dynamic visualization
- ✓ Concise user interface
- ✗ Represent all solutions
- ✗ Summarize history versions

# Motivation

## Codeopticon

Guo et al., UIST 15

The image displays a grid of ten code editor windows, each representing a different learner's history or a specific step in their development process. The editors are arranged in two rows of five. Each window includes a title bar, a code editor area, and a chat log below it.

- Learner 12 [untrack]**: Python 2. Shows a function to find minimum votes from a list of lists.
- Learner 49 [untrack]**: JavaScript. Shows a script for a game involving a robot and a korgus.
- Learner 45 [untrack]**: JavaScript. Shows a script for a game involving a robot and a korgus.
- Learner 27 [untrack]**: Python 2. Shows a function to print a grid of tiles.
- Learner 43 [untrack]**: Python 2. Shows a function to print a grid of tiles.
- Learner 57 [untrack]**: Python 2. Shows a function to get robot position.
- Learner 39 [untrack]**: Python 2. Shows a function to get robot direction.
- Learner 10 [untrack]**: Java. Shows Java code with a compilation error: "Error: <identifier> expected". The message "you mistyped 'public'" is shown in the chat.
- Learner 36 [untrack]**: Python 3. Shows a function to calculate the area of a rectangle.
- Learner 46 [untrack]**: Python 3. Shows a function to calculate the area of a rectangle.

Annotations labeled 'a.)' through 'e.)' point to specific windows:

- a.)** Points to Learner 49 [untrack].
- b.)** Points to Learner 45 [untrack].
- c.)** Points to Learner 57 [untrack].
- d.)** Points to Learner 10 [untrack].
- e.)** Points to Learner 36 [untrack].

- ✓ Dynamic visualization
- ✗ Concise user interface
- ✓ Represent all solutions
- ✗ Summarize history versions

RQ: What are the needs and challenges instructors have when conducting in-class coding exercises?

## Semi-structured interviews

---

- Six instructors
- Taught in classes that had more than 150 students

## RQ: What are the needs and challenges instructors have when conducting in-class coding exercises?

### Semi-structured interviews

- Six instructors
- Taught in classes that had more than 150 students

### Findings

- Need
  - Monitoring students' coding progress in real time
- Challenge
  - Visualizing students' progress at different granularity
  - Validating students' progress at scale
  - Providing tailored feedback on progress at scale

Students' coding progress → 2D map view

Students' coding progress → 2D map view



Start point  
(empty editors)

Students' coding progress → 2D map view

Destination  
(correct solutions)



Start point  
(empty editors)

```
numbers = range(41)  
sum1 = 0  
for i in numbers:  
    sum1 += i
```



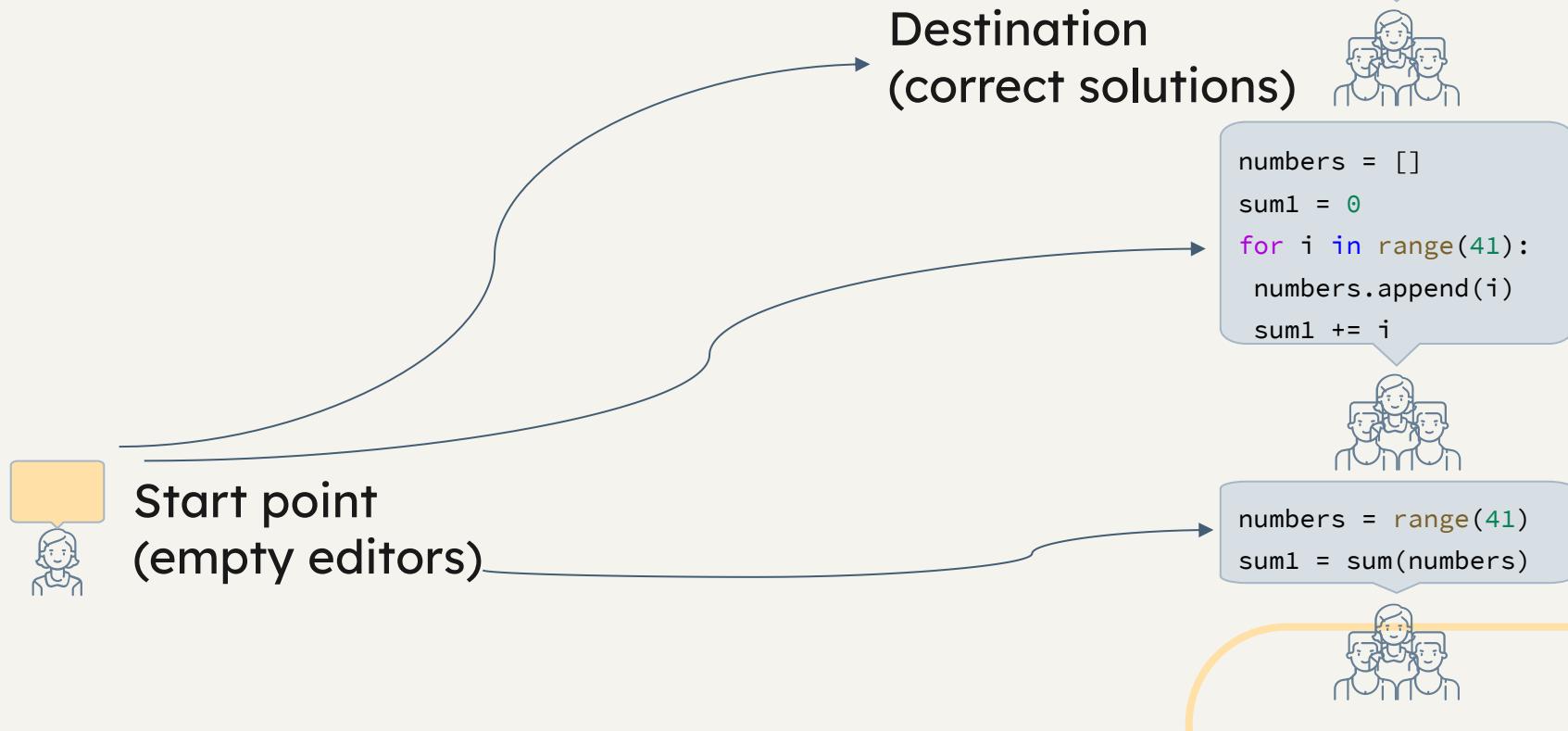
```
numbers = []  
sum1 = 0  
for i in range(41):  
    numbers.append(i)  
    sum1 += i
```



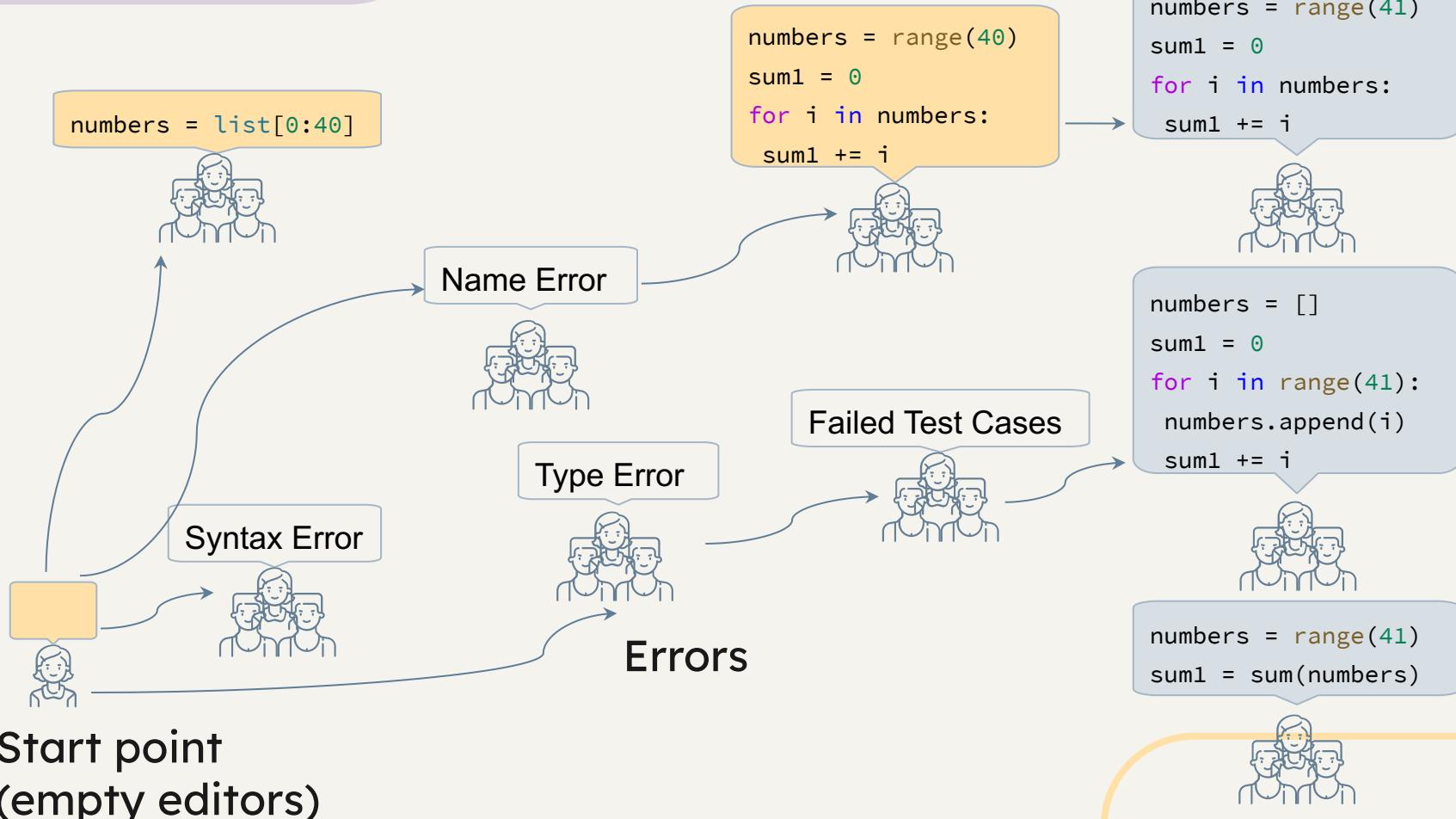
```
numbers = range(41)  
sum1 = sum(numbers)
```



Students' coding progress → 2D map view



# Design of VizProg



Start point  
(empty editors)

Hide History Versions

Reset Map



The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
    sum1+=x

assert sum1==820
del sum1
```

Search User:

Submit

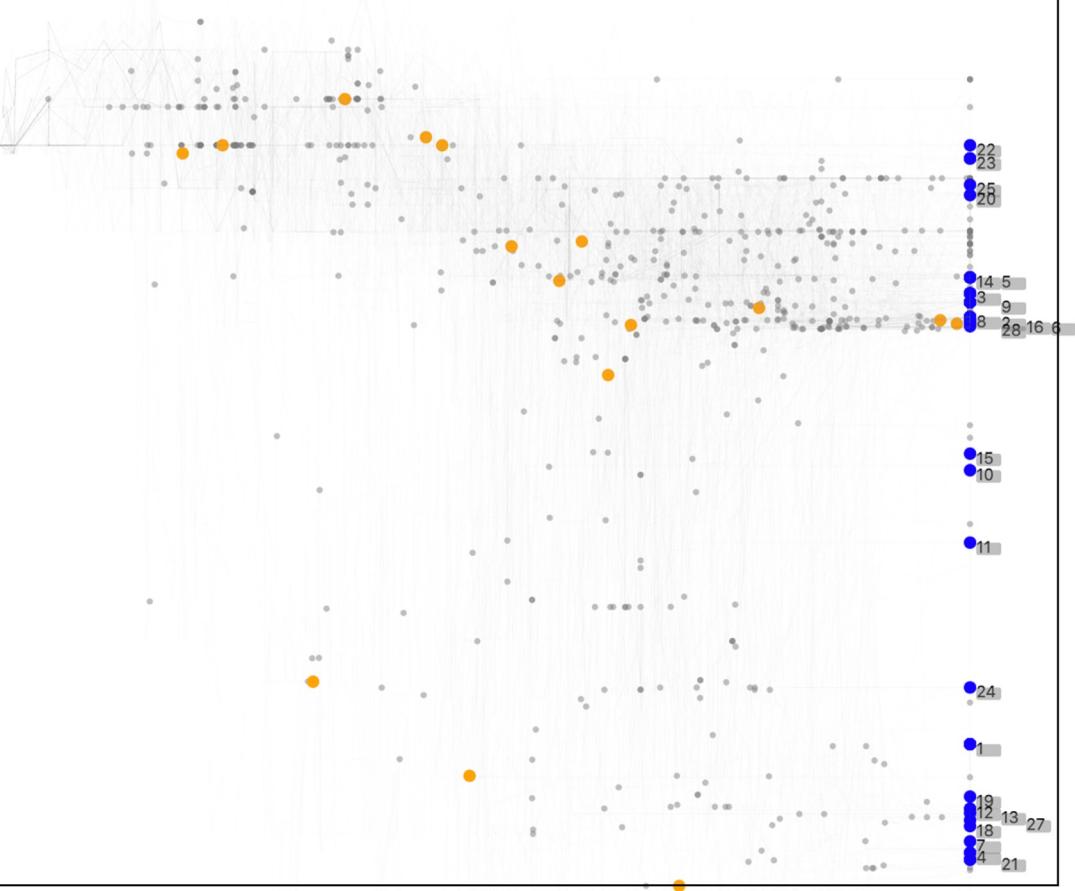
Feedback:

Submit

Hide History Versions

Reset Map

## Overall class progress 2D map view



The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
    sum1+=x

assert sum1==820
del sum1
```

Search User:

Submit

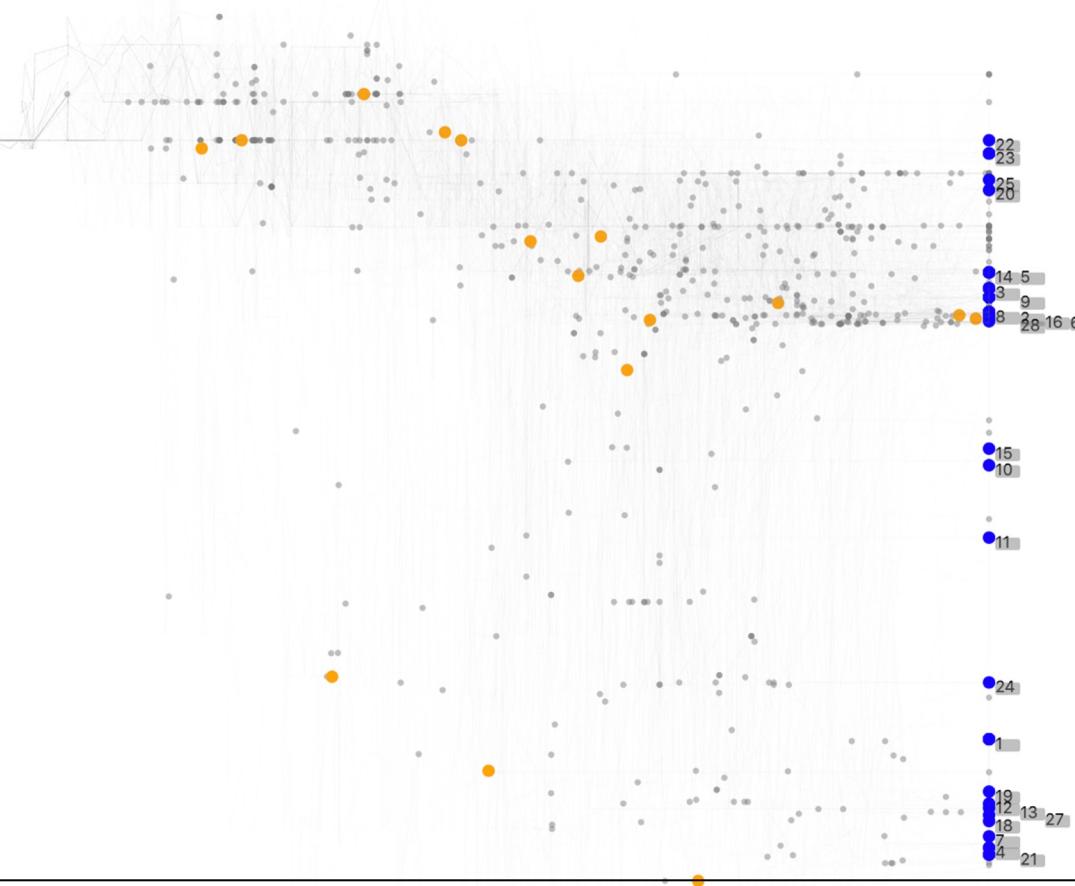
Feedback:

Submit

Hide History Versions

Reset Map

## Overall class progress 2D map view



The whole class has 117 students.  
Solution 10  
0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
sum1+=x

assert sum1==820
del sum1
```

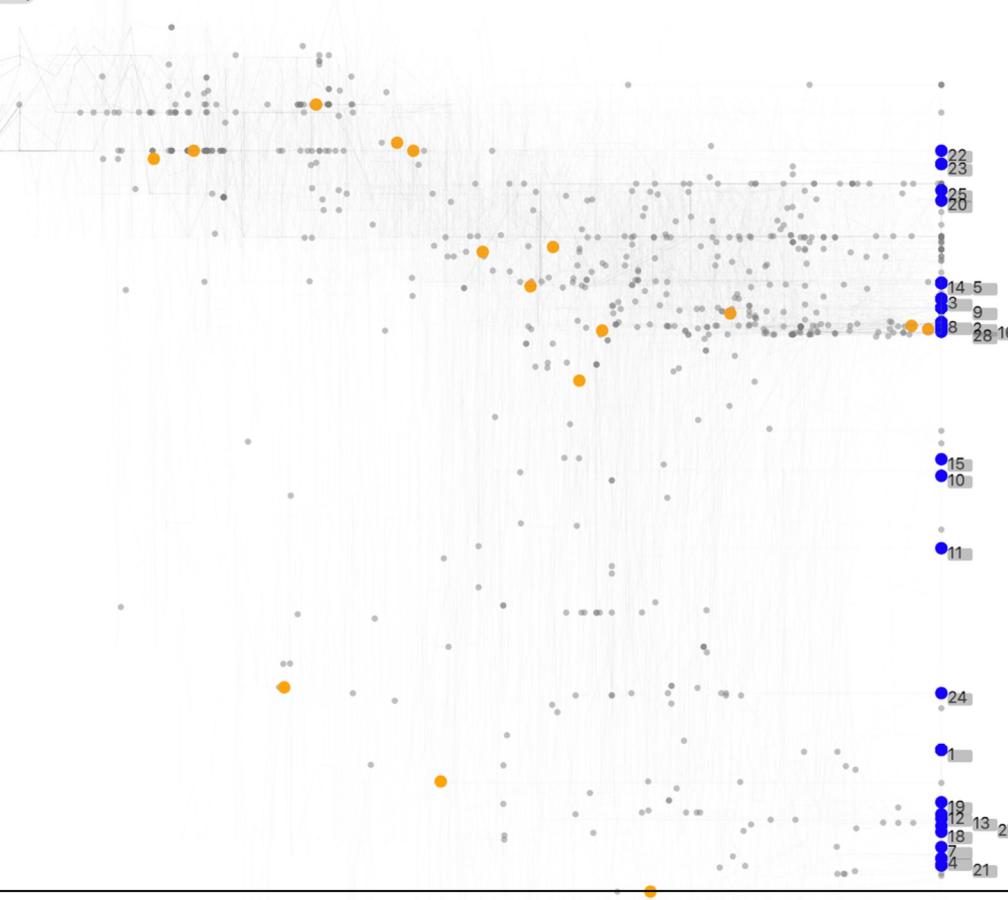
## Solution-centered view

Search User:  Submit  
Feedback:  Submit

Hide History Versions

Reset Map

## Overall class progress 2D map view



The whole class has 117 students.  
Solution 10  
0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
    sum1+=x

assert sum1==820
del sum1
```

## Solution-centered view

Search User:  Submit  
Feedback:  Submit

## Progress detailed view

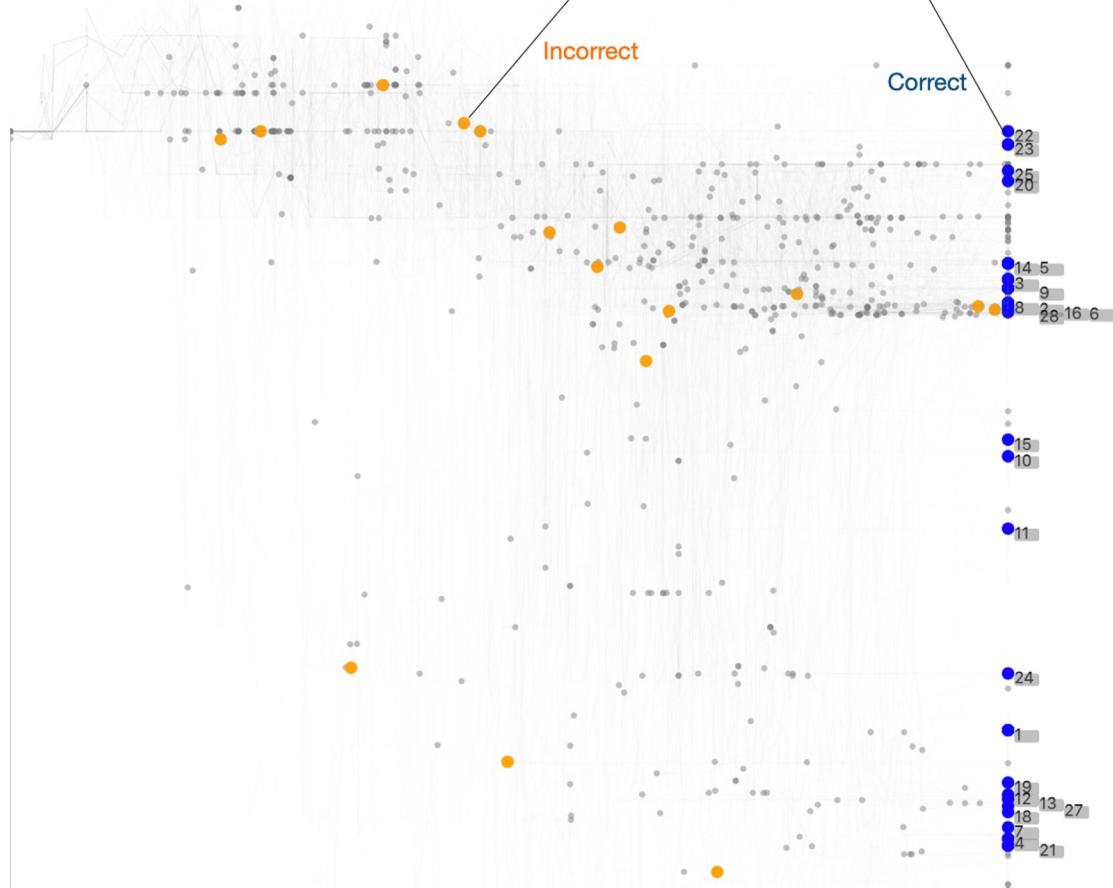
Hide History Versions

Reset Map

Large colored dots: where students are

Incorrect

Correct



The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
    sum1+=x

assert sum1==820
del sum1
```

Search User:

Submit

Feedback:

Submit

- Where students are (correct submission)
- Where students are (incorrect submission)
- Versions of code that students have tried
- Trajectories between two consecutive versions of code

Hide History Versions

Reset Map

Large colored dots: where students are

Incorrect

Correct



Small gray dots: versions of code that students have tried

The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

```
sum1=0
numbers=[]
for x in range(41):
    numbers.append(x)
    sum1+=x

assert sum1==820
del sum1
```

Search User:

Submit

Feedback:

Submit

- Where students are (correct submission)
- Where students are (incorrect submission)
- Versions of code that students have tried
- Trajectories between two consecutive versions of code

Hide History Versions

Reset Map

Large colored dots: where students are

Incorrect

Correct

Gray lines: trajectories between two consecutive versions of code

Small gray dots: versions of code that students have tried

The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

Search User:

Submit

Feedback:

Submit

```
sum1=0  
numbers=[]  
for x in range(41):  
    numbers.append(x)  
    sum1+=x  
  
assert sum1==820  
del sum1
```

- Where students are (correct submission)
  - Where students are (incorrect submission)
  - Versions of code that students have tried
- Trajectories between two consecutive code

Hide History Versions

Reset Map

Large colored dots: where students are

Incorrect

Correct

Gray lines: trajectories between two consecutive versions of code

Small gray dots: versions of code that students have tried

The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

Search User:

Submit

Feedback:

Submit

```
sum1=0  
numbers=[]  
for x in range(41):  
    numbers.append(x)  
    sum1+=x  
  
assert sum1==820  
del sum1
```

Different approaches

- Where students are (correct submission)
- Where students are (incorrect submission)
- Versions of code that students have tried
- Trajectories between two consecutive versions of code

Hide History Versions

Reset Map

Large colored dots: where students are

Incorrect

Correct

Gray lines: trajectories between two consecutive versions of code

Small gray dots: versions of code that students have tried

Far away from a correct solution

Close to a correct solution

The whole class has 117 students.

Solution 10

0 students have correct solutions, 0 students have incorrect solutions.

Search User:

Submit

Feedback:

Submit

```
sum1=0  
numbers=[]  
for x in range(41):  
    numbers.append(x)  
    sum1+=x  
  
assert sum1==820  
del sum1
```

Different approaches

- Where students are (correct submission)
- Where students are (incorrect submission)
- Versions of code that students have tried
- Trajectories between two consecutive versions of code

[ ]:

[Hide History Versions](#)[Reset Map](#)

The whole class has 117 students.  
Solution

Search User:  Submit  
Feedback:  Submit

22  
2325  
20

14 5  
3 9  
8 28 16 6 26

15  
10

11

24

1

19  
12 13 27  
18  
17

ac6\_8\_9.ipynb X

Real Time View Observe Code Deep Learning View Scatter Plot View VizPro Example OverCode Python 3 (ipykernel)

Hide History Versions

Reset Map

A  
numb

B  
for

C  
nu

22  
23  
25  
26

+ 14 5  
3 9  
8 28 16 6 26

15  
10

11

24

1

19  
12 13 27  
18  
7  
4 21

Hide History Versions

Reset Map

The whole class has 117 students.  
Solution 23  
3 students have correct solutions, 2 students  
have incorrect solutions.  
user\_47 user\_113 user\_89 user\_183 user\_23

Search User:  Submit  
Feedback:  Submit

```
numbers = range(0, 41)
sum1 = sum(numbers)
```

```
assert sum1==820
del sum1
```

```
in range(41):
```

```
assert sum1==820
del sum1
```

```
numbers = list(range(40))
print (numbers)
```

```
assert sum1==820
del sum1
```

15  
10

11

24

1

19  
12  
18  
13  
27  
7  
4  
21

Hide History Versions

Reset Map

## Statistics

The whole class has 117 students.  
Solution 8  
28 students have correct solutions, 3 students have incorrect solutions.

Search User:  Submit  
Feedback:  Submit

user\_75 user\_177 user\_123 user\_212 user\_33  
user\_110 user\_133 user\_65 user\_37 user\_64  
user\_216 user\_10 user\_159 user\_154 user\_155  
user\_49 user\_8 user\_32 user\_158 user\_130  
user\_29 user\_36 user\_142 user\_102 user\_136  
user\_55 user\_17 user\_27 user\_60 user\_16 user\_15

```
numbers = range(41)
sum1 = 0
for num in numbers:
    sum1 += num
assert sum1==820
del sum1
```

```
numbers = range(0, 41)
sum1 = 0
for x in numbers:
    if x > sum1:
        sum1 = sum1 + x
print(sum1)

assert sum1==820
del sum1
```

```
numbers= [0:41]
sum1=0
for num in numbers:
    num += sum1

assert sum1==820
```



X

Code Real Time View Observe Code Deep Learning View Scatter Plot View VizPro Example OverCode

Python 3 (ipykernel)

Hide History Versions

Reset Map

The whole class has 117 students.  
Solution 8  
28 students have correct solutions, 3 students have incorrect solutions.

Search User:  Submit

Feedback:  Submit

Color code student IDs

numbers = range(41)  
sum1 = 0  
for num in numbers:  
 sum1 += num  
  
assert sum1==820  
del sum1

numbers = range(0, 41)  
sum1 = 0  
for x in numbers:  
 if x > sum1:  
 sum1 = sum1 + x  
print(sum1)  
  
assert sum1==820  
del sum1

numbers= [0:41]  
sum1=0  
for num in numbers:  
 num += sum1  
  
assert sum1==820

11  
10  
15  
18  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117



+ X Code

Real Time View Observe Code Deep Learning View Scatter Plot View VizPro Example OverCode

Python 3 (ipykernel)

Hide History Versions

Reset Map



The whole class has 117 students.

Solution 8

28 students have correct solutions, 3 students have incorrect solutions.

```
user_75 user_177 user_123 user_212 user_33
user_110 user_133 user_65 user_37 user_64
user_216 user_10 user_159 user_154 user_155
user_49 user_8 user_32 user_158 user_130
user_29 user_36 user_142 user_102 user_136
user_55 user_17 user_27 user_60 user_16 user_15
```

```
numbers = range(41)
sum1 = 0
for num in numbers:
    sum1 += num
assert sum1==820
del sum1
```

```
numbers = range(0, 41)
sum1 = 0
for x in numbers:
    if x > sum1:
        sum1 = sum1 + x
print(sum1)
```

```
assert sum1==820
del sum1
```

```
numbers= [0:41]
sum1=0
for num in numbers:
    num += sum1
```

```
assert sum1==820
```

Search User:

Submit

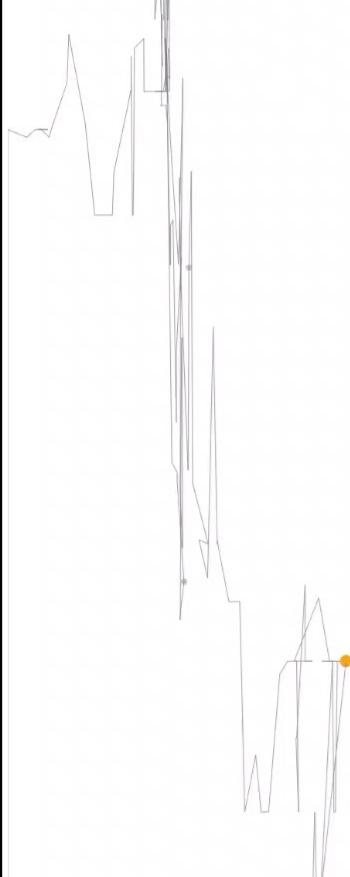
Feedback:

Submit

A list of submissions made by the students above

Hide History Versions

Reset Map



The whole class has 117 students.

Solution 22

1 students have correct solutions, 3 students have incorrect solutions.

user\_46 user\_127 user\_180 user\_199

```
numbers = range(41)
sum1 = sum(numbers)
```

```
assert sum1==820
del sum1
```

```
for val in range(41):
    numbers = val
    print(numbers)
```

```
assert sum1==820
del sum1
```

```
numbers = range(0:40)
print(numbers)
```

```
assert sum1==820
del sum1
```

```
numbers = list(range(0:41))
print(numbers)
```

```
assert sum1==820
del sum1
```

```
1
```

```
19
```

```
12
```

```
13
```

```
27
```

```
18
```

```
7
```

```
4
```

```
21
```

Search User:  SubmitFeedback:  Submit

NameError: name 'accum' is not defined on line 2

```
1 numbers=[0,1,2,3,4,5,6,7,8,9,10,
2 sum1=accum(numbers)
3
4
5 assert sum1==820
6 del sum1
7
```

Failed the test case

```
1 numbers=[0,1,2,3,4,5,6,7,8,9,10,
2 acccum=0
3 sum1=numbers
4
5
6 assert sum1==820
7 del sum1
8
```

NameError: name 'nums' is not defined on line 3

```
1 numbers=[0,1,2,3,4,5,6,7,8,9,10,
2 accum = 0
3 for w in nums:
4     accum = accum + w
5     print(accum)
6
7
8 assert sum1==820
9 del sum1
10
```

[ ]:

Hide History Versions

Reset Map



```
2 print(numbers)
3
4
5
6 assert sum1==820
7 del sum1
8
```

numbers = range(0, 41)
sum1 = sum(numbers)

```
assert sum1==820
del sum1
```

```
in range(41):
```

```
assert sum1==820
del sum1
```

```
numbers = list(range(0,41))
print (numbers)
```

```
assert sum1==820
del sum1
```

```
Failed the test case
```

```
1 numbers = list(range(0,41))
2 print(numbers)
3
4
5
6 assert sum1==820
7 del sum1
8
```

```
1 numbers = list(range(0,41))
2 print(numbers)
3
4
5
6 assert sum1==820
7 del sum1
8
```

```
1 numbers = list(range(0,41))
2 print(numbers)
3 sum1 = 0
4 for n in numbers:
5     sum1 += n
6
7
8 assert sum1==820
9 del sum1
10
```

Failed the test case

```
1 numbers = list(range(40))
2 print (numbers)
3
4
5 assert sum1==820
6 del sum1
7
```

[ ]:

Hide History Versions

Reset Map

Search User:  SubmitFeedback:  Submit

```
numbers = range(0, 41)
sum1 = sum(numbers)
```

```
assert sum1==820
del sum1
```

```
in range(41):
```

```
    assert sum1==820
    del sum1
```

```
numbers = list(range(40))
print (numbers)
```

```
assert sum1==820
del sum1
```

[ ]:

Hide History Versions

Reset Map



The whole class has 117 students.

Solution 23

3 students have correct solutions, 2 students have incorrect solutions.

user\_47 user\_113 user\_89 user\_183 user\_23

Search User:  SubmitFeedback:  Submit

```
numbers = range(0, 41)
sum1 = sum(numbers)
```

```
assert sum1==820
del sum1
```

```
in range(41):
```

```
    assert sum1==820
    del sum1
```

```
numbers = list(range(40))
print (numbers)
```

```
assert sum1==820
del sum1
```

## VizProg

16 teaching assistants

- Two 60 minute lab sessions

## OverCode

Adapted for real-time

A replay of approximately 110 students solving a programming exercise (15 min)

Quiz questions to evaluate their understanding of students' progress and approaches (20 min)

Survey on user experience

Reflective interview for comparing the systems  
(This part only happened after participants used the two systems)

## Understand students' problems more accurately

---

- Accuracy of quiz questions
  - VizProg:  $\mu = 79.6\%$ ,  $\sigma = 0.1$
  - OverCode:  $\mu = 51.4\%$ ,  $\sigma = 0.2$
  - $p < 0.0001$

Understand students' problems more accurately

---

Understand issues faster in live settings

---

- Participants started finding errors significantly earlier with VizProg than with OverCode
  - VizProg:  $\mu = 500.1\text{s}$ ,  $\sigma = 468.2$
  - OverCCode:  $\mu = 1069.9\text{s}$ ,  $\sigma = 360.2$
  - $p < 0.05$

Understand students' problems more accurately

---

Understand issues faster in live settings

---

Understand in live settings with less context switching

---

Understand students' problems more accurately

---

Understand issues faster in live settings

---

Understand in live settings with less context switching

---

Less effort to validate students' progress

---

Understand students' problems more accurately

---

Understand issues faster in live settings

---

Understand in live settings with less context switching

---

Less effort to validate students' progress

---

Quickly form a strategy to decide who to give feedback to

---

## VizProg is intuitive to use

---

- The movement from left to right tells that students are making progress
- Examining progress at different levels on map is straightforward

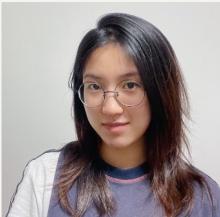
## Trajectories in VizProg ease the reasoning process

---

- The dynamics and paths provide visual guide to reasoning about students' behavior
- Normal behavior
  - Approaches to solve the exercises
  - Make progress toward a final solution
- Abnormal behavior
  - Have a correct solution but change to incorrect solutions later on
  - Wander in the middle of the map and never reach the right side of it

# Visualizing Students' Progress to Facilitate Real-time Pedagogy in In-Class Programming Exercises At Scale

- A better understanding of the needs and challenges that instructors face when monitoring students' in-class coding exercise
- VizProg, a system that encodes students' approaches and progress on a 2D map view to facilitate monitoring students' progress in real-time
- Evidence showing that VizProg can help identify more misunderstandings and important behaviors in coding exercises than a baseline system



**Ashley Zhang**  
University of Michigan  
<https://gezhangrp.com/>



**Yan Chen**  
Virginia Tech



**Steve Oney**  
University of Michigan