Weekly Programs in CS1: Experiences with Many Small Auto-Graded Programs

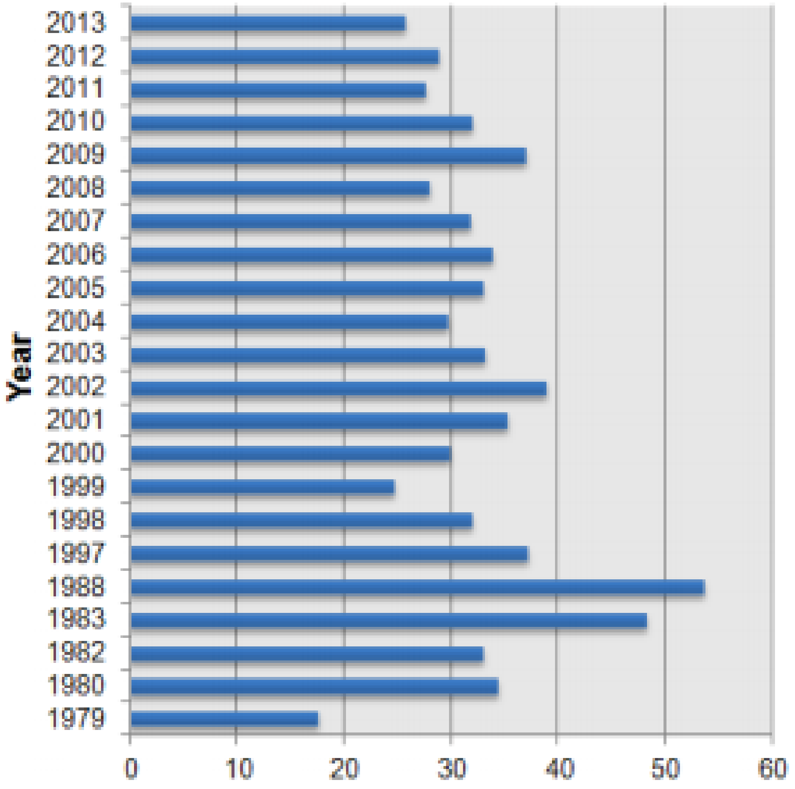
Joe Michael Allen

Advisor: Frank Vahid
Dept. of Computer Science and Engineering
University of California, Riverside

UNIVERSITY OF CALIFORNIA, RIVERSIDE

Problem

Mean Percentage of Non-Passing Students in CS1



~28%

~55%

~18%

Watson, C. and Li, F. “Failure Rates in Introductory Programming Revisited, ” iTiCSE, 2014
http://dro.dur.ac.uk/19223/1/19223.pdf%3FDDD10%2Bd74ks0%2Bdcs0lw

- CS1 issues:
  - High student stress
  - Student dissatisfaction
  - Academic dishonesty
  - Low grades
  - High non-passing rates

~ 30% non-passing rate over the past 30 years

Goal

- Improve the student experience
  - Improve satisfaction & happiness
  - Without worsening performance

- Problem: Weekly programming assignments
  - Large part of the students' experience
  - Key source of issues – student struggle/fear

Outline

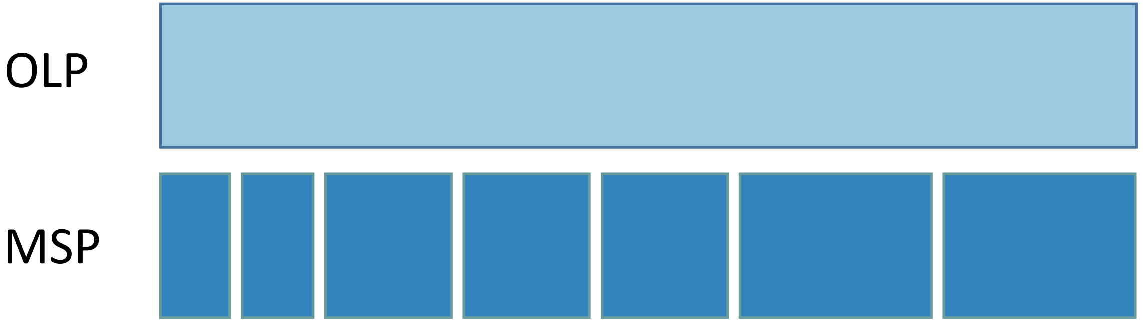
- OLPs vs. MSPs
- Study 1 – Satisfaction & grade performance
- Study 2 – MSP usage analysis - UCR
- Study 3 – MSP usage analysis - Other universities
- Future work
- Conclusion

Traditional: One Large Program (OLP) each week

- Solution 50-200 lines
- Long spec

Many Small Programs (MSPs) each week

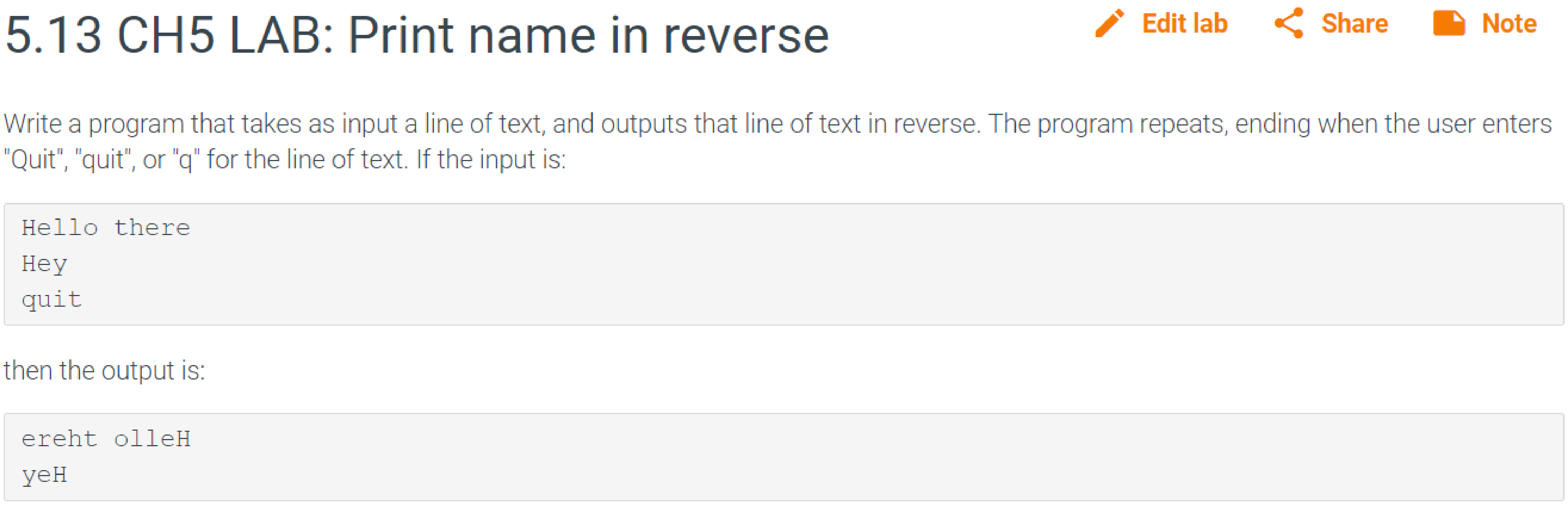
- Our approach: 5-7 MSPs
  - Solution 10-50 lines each
  - Short & concise spec



- Benefits
  - Less intimidating
  - Pivot if stuck
  - Build confidence, more practice

- Enabled by new auto-graders
  - Easy to create / Instant feedback
  - zyLabs (zyBooks): ~30 min create lab

MSPs - prompt



5.13 CH5 LAB: Print name in reverse

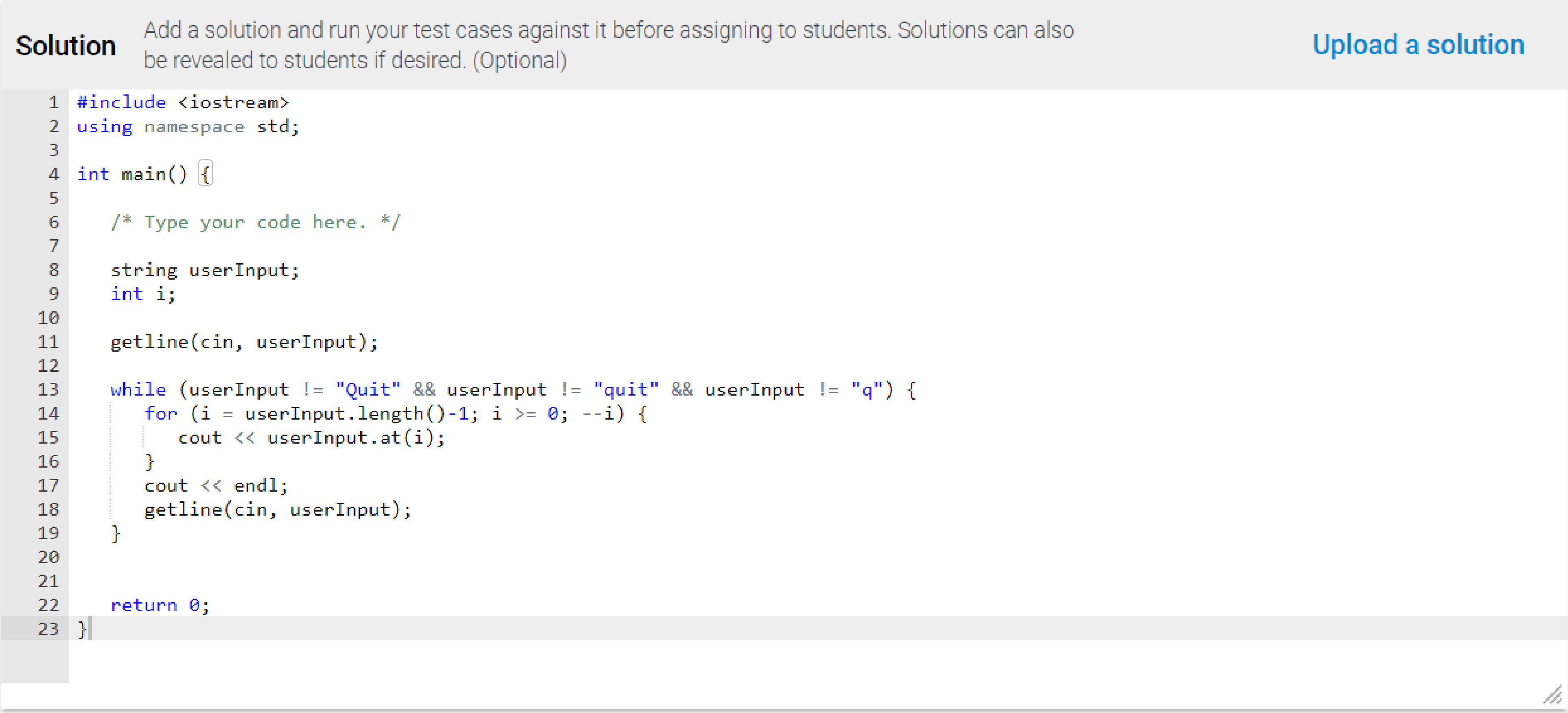
Write a program that takes as input a line of text, and outputs that line of text in reverse. The program repeats, ending when the user enters "Quit", "quit", or "q" for the line of text. If the input is:

```
Hello there
Hey
quit
```

then the output is:

```
ereht olleH
yeH
```

MSPs - solution



Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

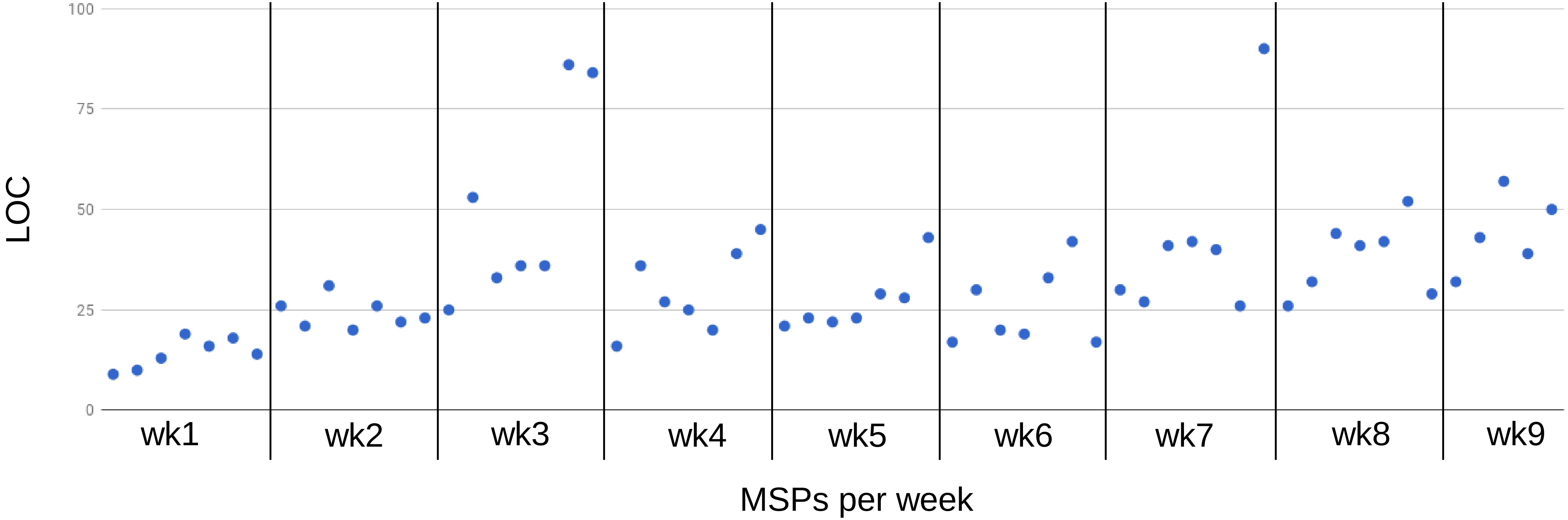
```
#include <iostream>
using namespace std;

int main() {

   /* Type your code here. */

   string userInput;
   int i;

   getline(cin, userInput);

   while (userInput != "Quit" && userInput != "quit" && userInput != "q") {
      for (i = userInput.length()-1; i >= 0; --i) {
         cout << userInput.at(i);
      }
      cout << endl;
      getline(cin, userInput);
   }


   return 0;
}
```

MSPs – lines of code (LOC)

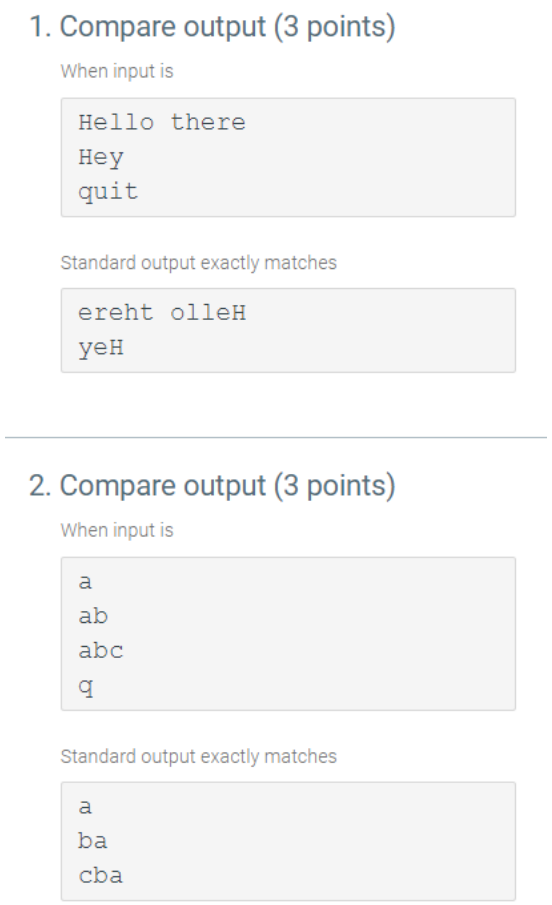
Min LOC: 9

Average LOC: 34

Max LOC: 90



MSPs – test cases



1. Compare output (3 points)

When input is

```
Hello there
Hey
quit
```

Standard output exactly matches

```
ereht olleH
yeH
```

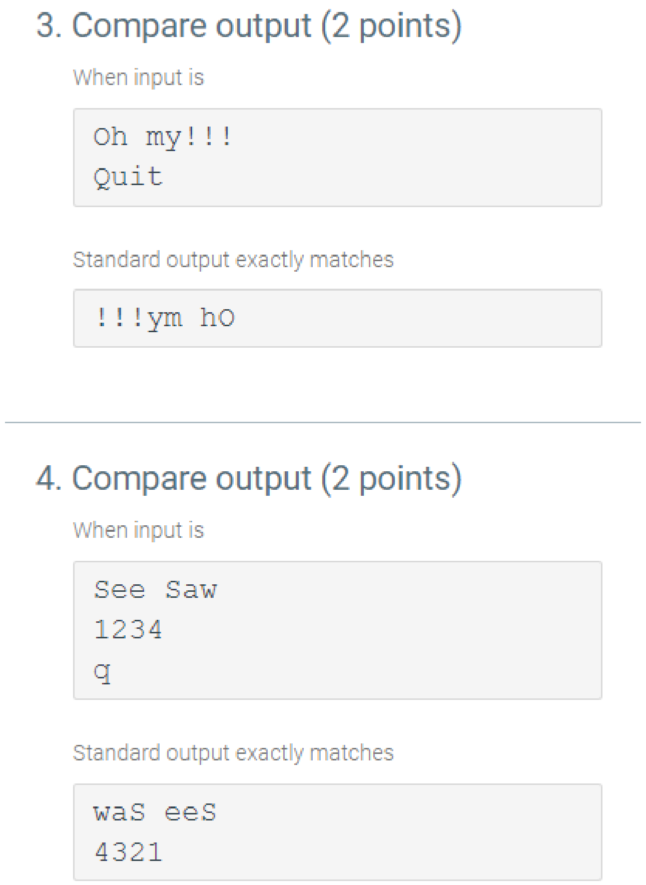
2. Compare output (3 points)

When input is

```
a
ab
abc
q
```

Standard output exactly matches

```
a
ba
cba
```



3. Compare output (2 points)

When input is

```
Oh my!!!
Quit
```

Standard output exactly matches

```
!!!ym hO
```

4. Compare output (2 points)

When input is

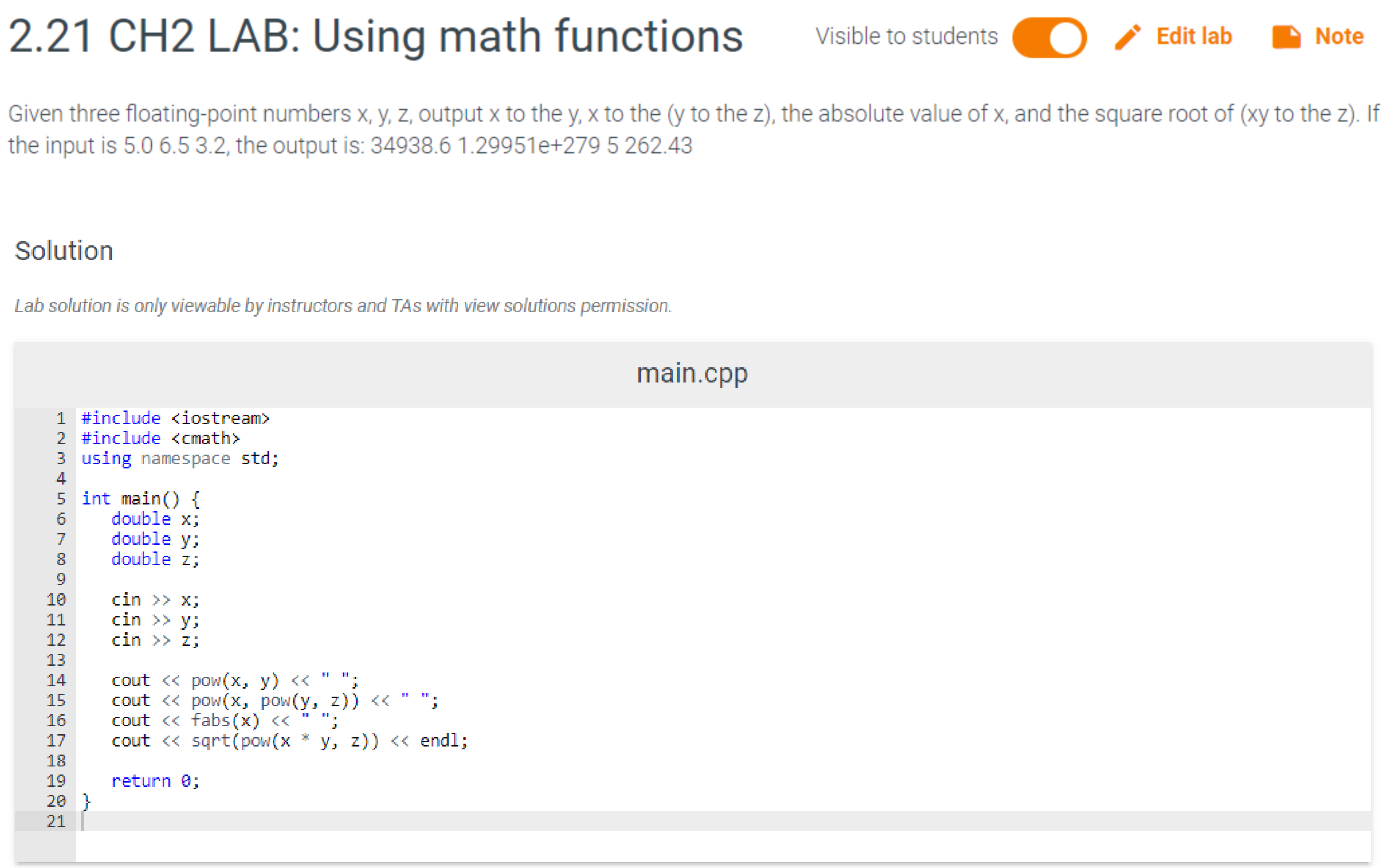
```
See Saw
1234
q
```

Standard output exactly matches

```
waS eeS
4321
```

- Test cases:
  - 10 points per MSP
  - Input/output tests
  - Unit tests

MSP sample 1



2.21 CH2 LAB: Using math functions

Given three floating-point numbers x, y, z, output x to the y, x to the (y to the z), the absolute value of x, and the square root of (xy to the z). If the input is 5.0 6.5 3.2, the output is: 34938.6 1.29951e+279 5 262.43

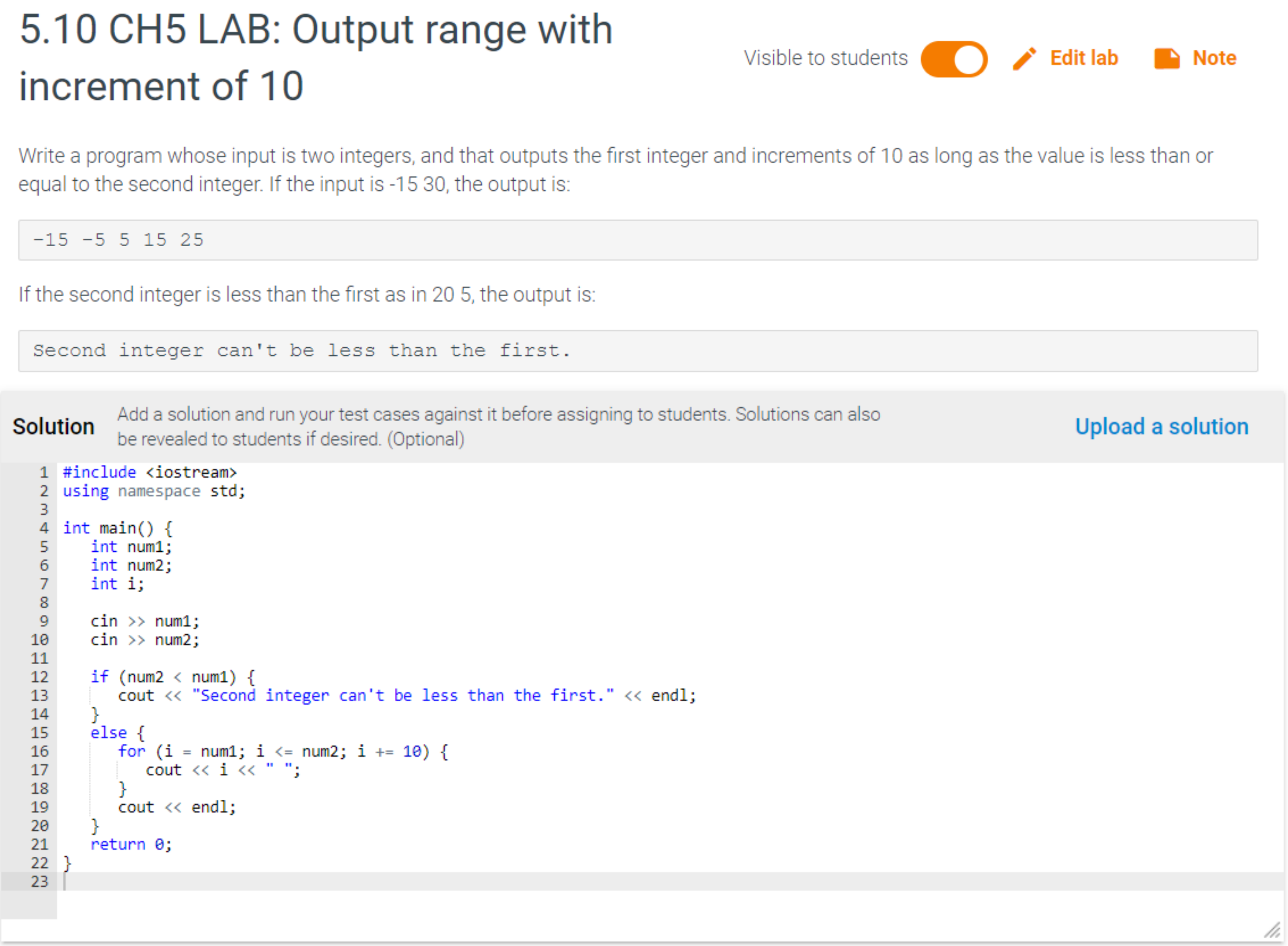
Solution

Lab solution is only viewable by instructors and TAs with view solutions permission.

main.cpp

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
   double x;
   double y;
   double z;

   cin >> x;
   cin >> y;
   cin >> z;

   cout << pow(x, y) << " ";
   cout << pow(x, pow(y, z)) << " ";
   cout << fabs(x) << " ";
   cout << sqrt(pow(x * y, z)) << endl;

   return 0;
}

```

MSP sample 2



5.10 CH5 LAB: Output range with increment of 10

Write a program whose input is two integers, and that outputs the first integer and increments of 10 as long as the value is less than or equal to the second integer. If the input is -15 30, the output is:

```
-15 -5 5 15 25
```

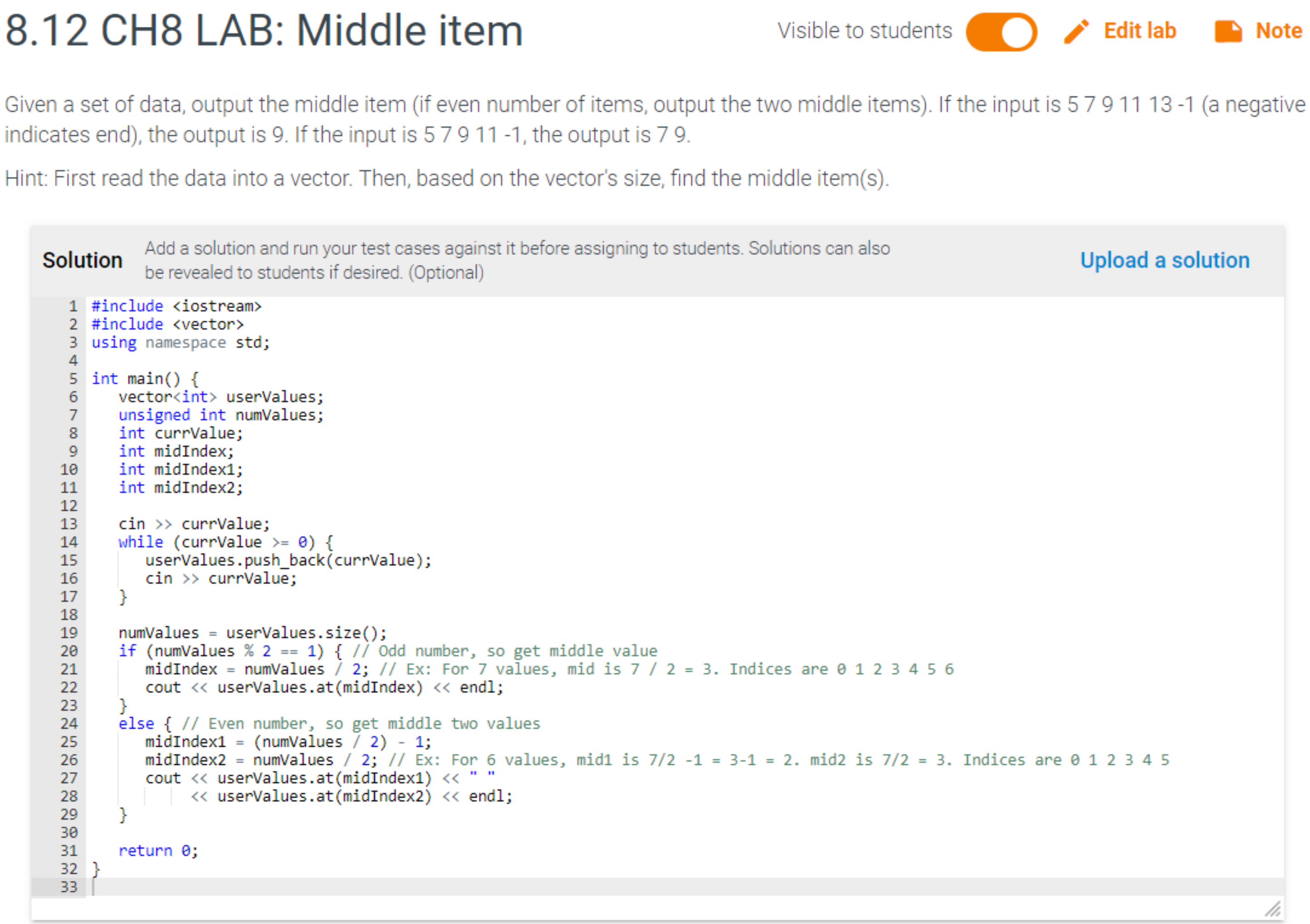
If the second integer is less than the first as in 20 5, the output is:

```
Second integer can't be less than the first.
```

Solution Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

```cpp
#include <iostream>
using namespace std;

int main() {
   int num1;
   int num2;
   int i;

   cin >> num1;
   cin >> num2;

   if (num2 < num1) {
      cout << "Second integer can't be less than the first." << endl;
   }
   else {
      for (i = num1; i <= num2; i += 10) {
         cout << i << " ";
      }
      cout << endl;
   }
   return 0;
}
```

MSP sample 3



8.12 CH8 LAB: Middle item

Given a set of data, output the middle item (if even number of items, output the two middle items). If the input is 5 7 9 11 13 -1 (a negative indicates end), the output is 9. If the input is 5 7 9 11 -1, the output is 7 9.

Hint: First read the data into a vector. Then, based on the vector's size, find the middle item(s).

Solution Add a solution and run your test cases against it before assigning to students. Solutions can also be revealed to students if desired. (Optional)

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
   vector<int> userValues;
   unsigned int numValues;
   int currValue;
   int midIndex;
   int midIndex1;
   int midIndex2;

   cin >> currValue;
   while (currValue >= 0) {
      userValues.push_back(currValue);
      cin >> currValue;
   }

   numValues = userValues.size();
   if (numValues % 2 == 1) { // Odd number, so get middle value
      midIndex = numValues / 2; // Ex: For 7 values, mid is 7 / 2 = 3. Indices are 0 1 2 3 4 5 6
      cout << userValues.at(midIndex) << endl;
   }
   else { // Even number, so get middle two values
      midIndex1 = (numValues / 2) - 1;
      midIndex2 = numValues / 2; // Ex: For 6 values, mid1 is 7/2 -1 = 3-1 = 2. mid2 is 7/2 = 3. Indices are 0 1 2 3 4 5
      cout << userValues.at(midIndex1) << " "
           << userValues.at(midIndex2) << endl;
   }

   return 0;
}

```

Study 1 – Satisfaction & grade performance

CS1 course at UCR during Spring 2017; 10 week quarter

Same online textbook
Same topics taught each week
Same midterm & final

7 MSPs; 50-point full-credit threshold

OLP

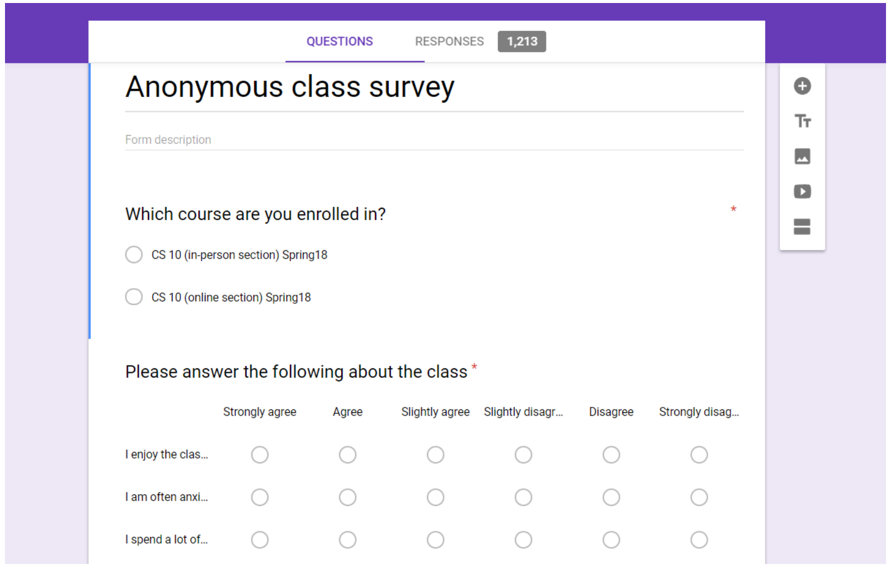
MSP

2 in-person sections; 166 students
Instructor 1
No collaboration
Prog assignments: 25%, Midterm: 20%

1 online section; 76 students
Instructor 2
Yes collaboration
Prog assignments: 15%, Midterm: 30%

Methods

- Student “stress” survey
  - Given week 8 of the quarter
  - Ask students about their experience
  - 18 questions: Strongly agree (6) to Strongly disagree (0)
  - Bonferroni correction: Conservative interpretation of p-value



QUESTIONS RESPONSES 1,213

Anonymous class survey

Form description

Which course are you enrolled in? *

○ CS 10 (in-person section) Spring18

○ CS 10 (online section) Spring18

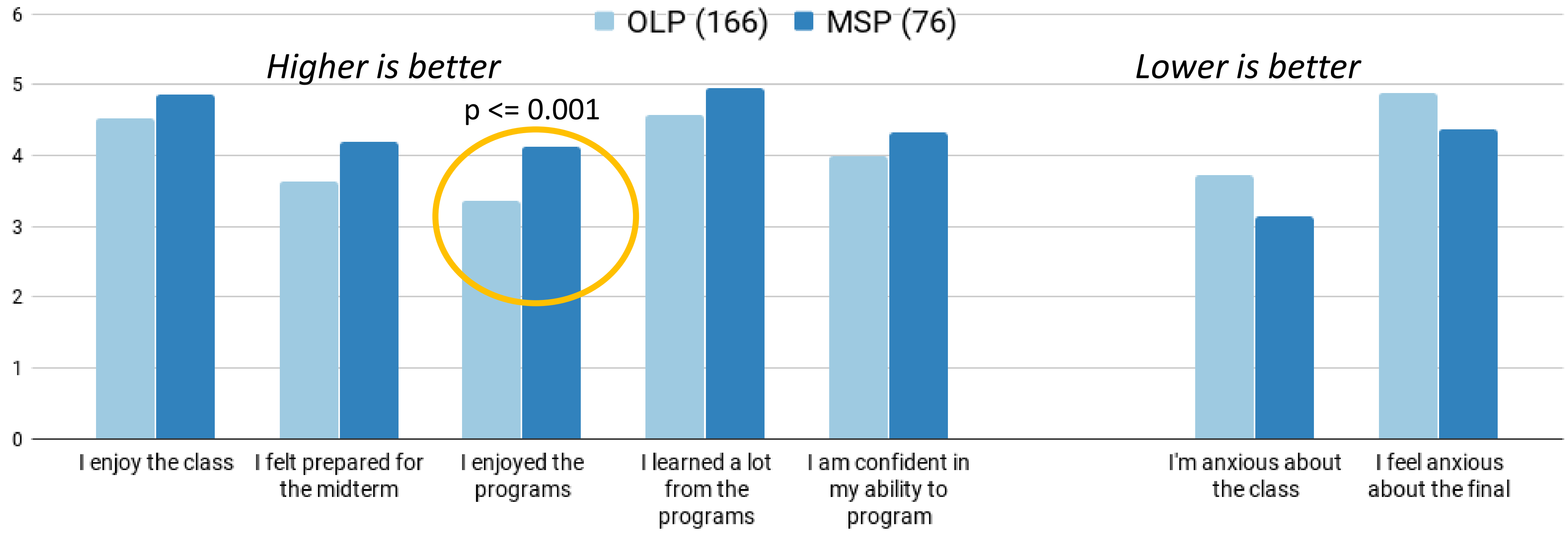
Please answer the following about the class *

	Strongly agree	Agree	Slightly agree	Slightly disagr...	Disagree	Strongly disag...
I enjoy the clas...	○	○	○	○	○	○
I am often anxi...	○	○	○	○	○	○
I spend a lot of...	○	○	○	○	○	○

- Student outcomes
  - Participation, Challenge, and Programming Activities, Midterm, Final, Total grade
  - Bonferroni correction

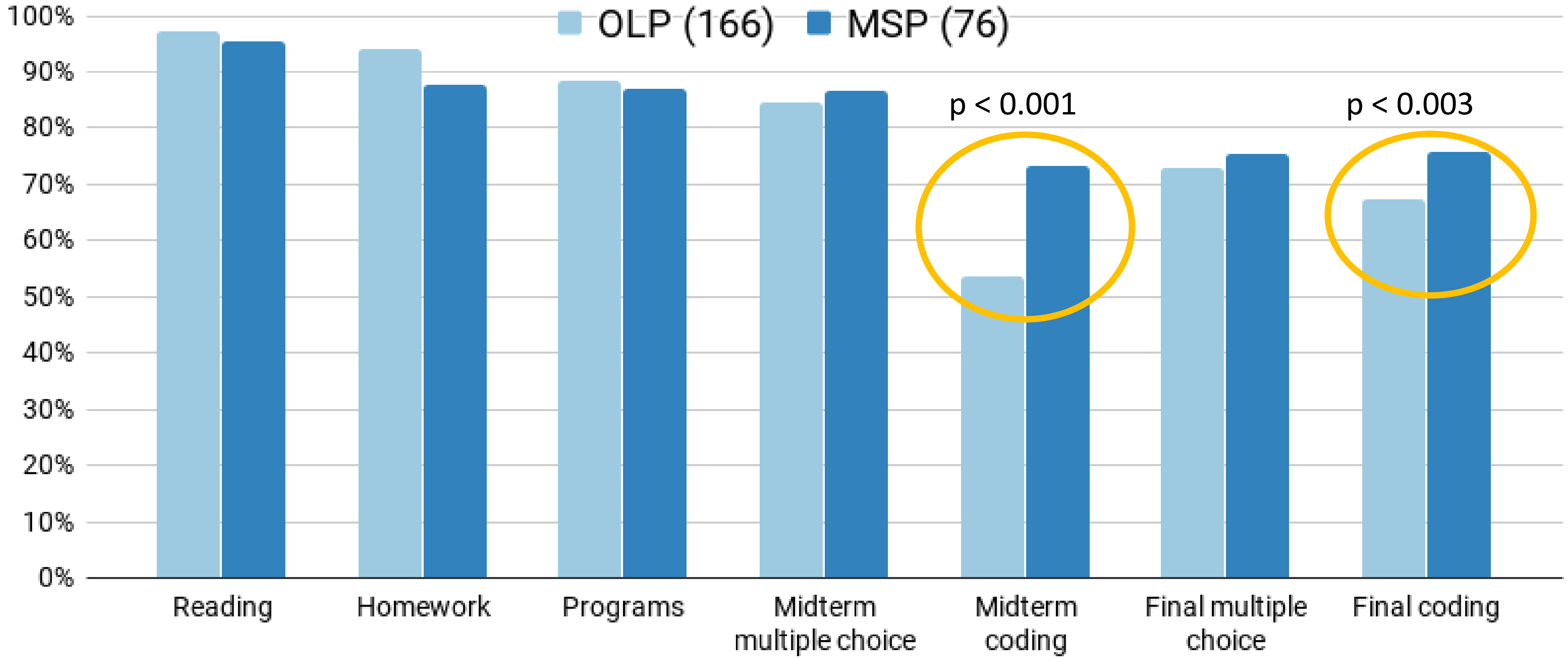
Student satisfaction

- MSP group had more favorable responses for almost all questions
- MSP group preferred the class 9% more than OLP group



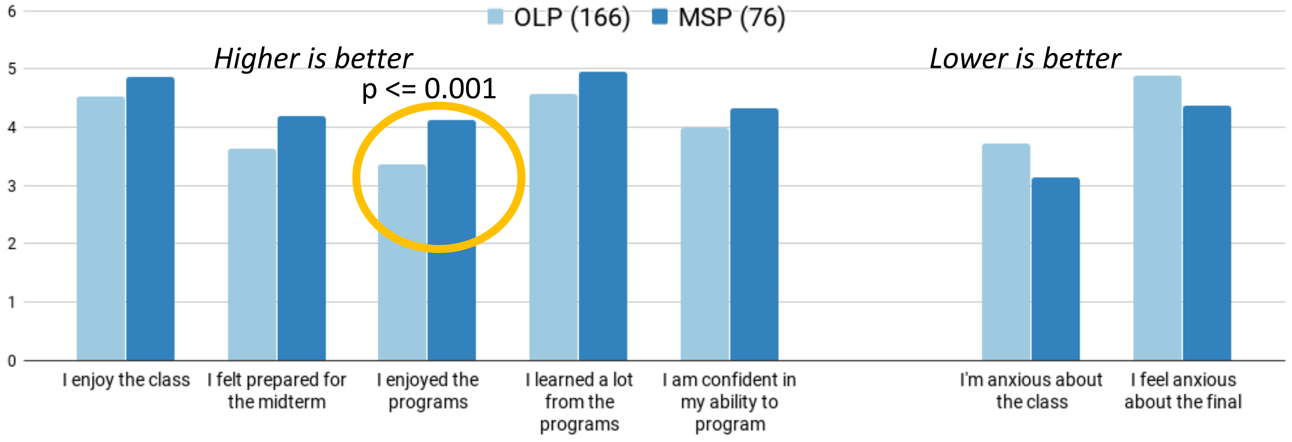
Student outcomes

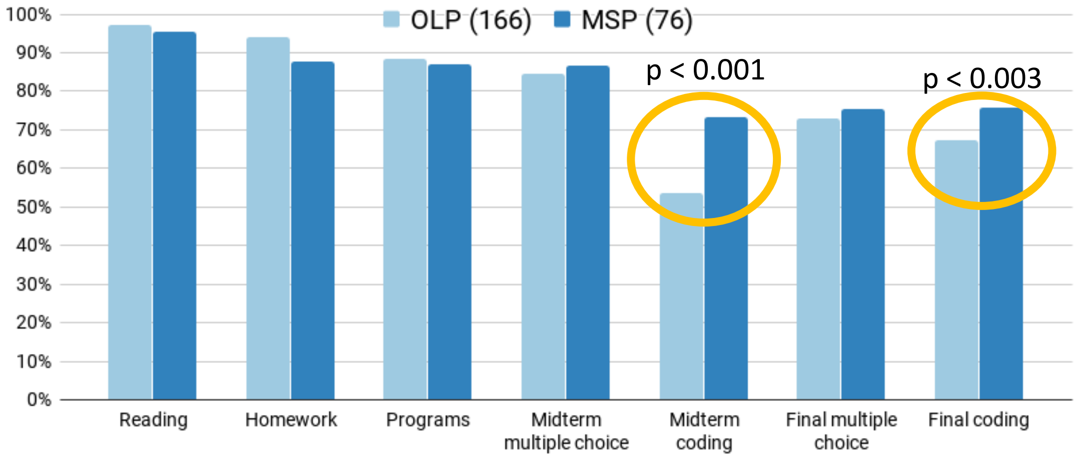
- MSP group performed better on coding portions of exams



Study 1 - Conclusion

- MSP students preferred the class more than OLP students
- MSP student grade performance did not worsen
  - Performed better on coding portion of exams



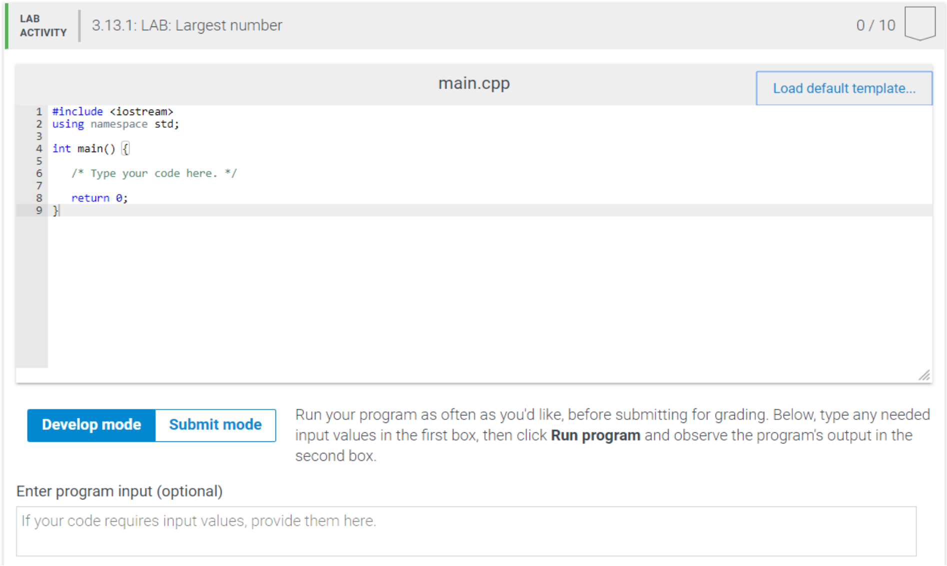


Study 2 – MSP usage analysis - UCR

- Time spent per week?
- Time spent per MSP?
- When do students start on MSPs?
- What % completed each day?
- Given full-credit threshold, do students complete more?
- Do students pivot, or switch among MSPs when stuck?
- Are MSPs used to study for exams?
- **Won't MSP CS1 students do poorly in an OLP CS2?**

Data collection

- UCR CS1 Spring 2017 MSP section: 76 students
- Used zyLabs from zyBooks
- Collected:
  - 48,000 develop runs, 16,000 submissions
  - Each has labID, userID, score, maxScore, time/date



LAB ACTIVITY | 3.13.1: LAB: Largest number 0 / 10

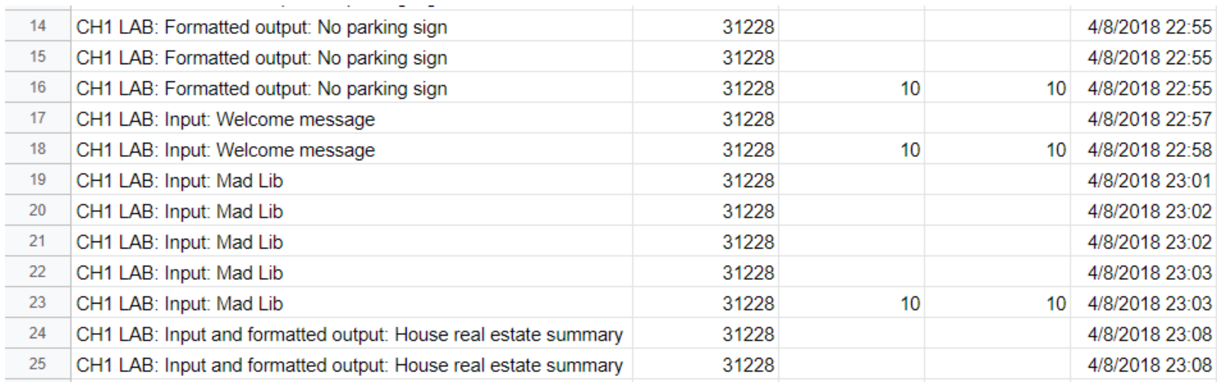
main.cpp

```
#include <iostream>
using namespace std;

int main() {

   /* Type your code here. */

   return 0;
}
```

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

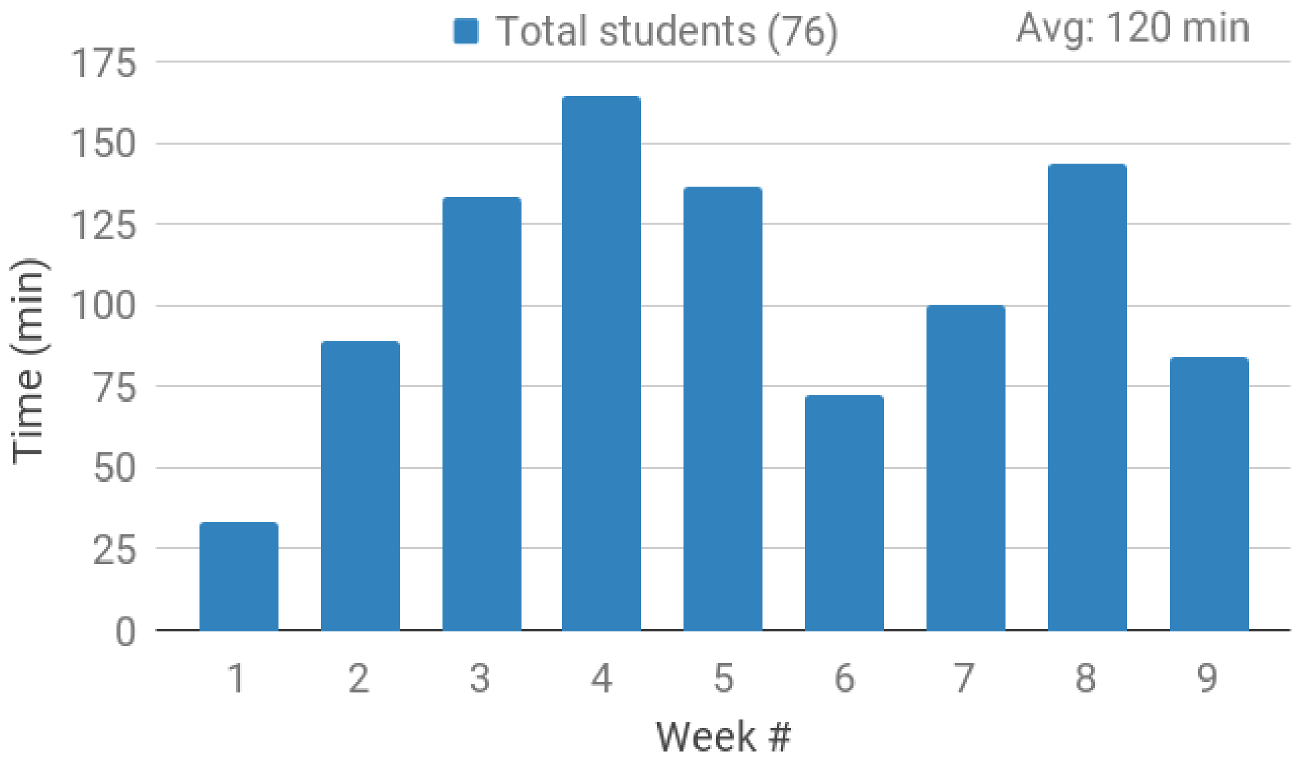
If your code requires input values, provide them here.



	labID	userID	score	maxScore	timestamp
14	CH1 LAB: Formatted output: No parking sign	31228			4/8/2018 22:55
15	CH1 LAB: Formatted output: No parking sign	31228			4/8/2018 22:55
16	CH1 LAB: Formatted output: No parking sign	31228	10	10	4/8/2018 22:55
17	CH1 LAB: Input: Welcome message	31228			4/8/2018 22:57
18	CH1 LAB: Input: Welcome message	31228	10	10	4/8/2018 22:58
19	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:01
20	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:02
21	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:02
22	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:03
23	CH1 LAB: Input: Mad Lib	31228	10	10	4/8/2018 23:03
24	CH1 LAB: Input and formatted output: House real estate summary	31228			4/8/2018 23:08
25	CH1 LAB: Input and formatted output: House real estate summary	31228			4/8/2018 23:08

Q: How much time do students spend working on MSPs each week?

A: At least 120 min / week

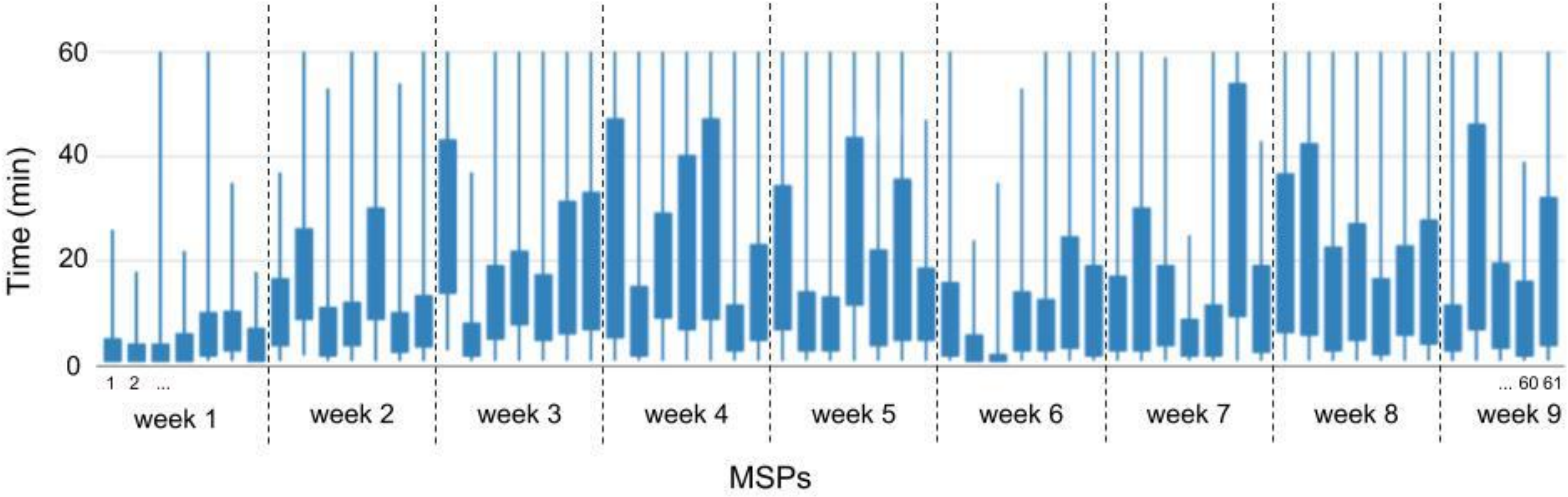


NOTE: ***Underestimate***.

Students with 0 subs or 0 time excluded. Avg is for weeks 2-8.

Q: How much time do students spend working on each MSP?

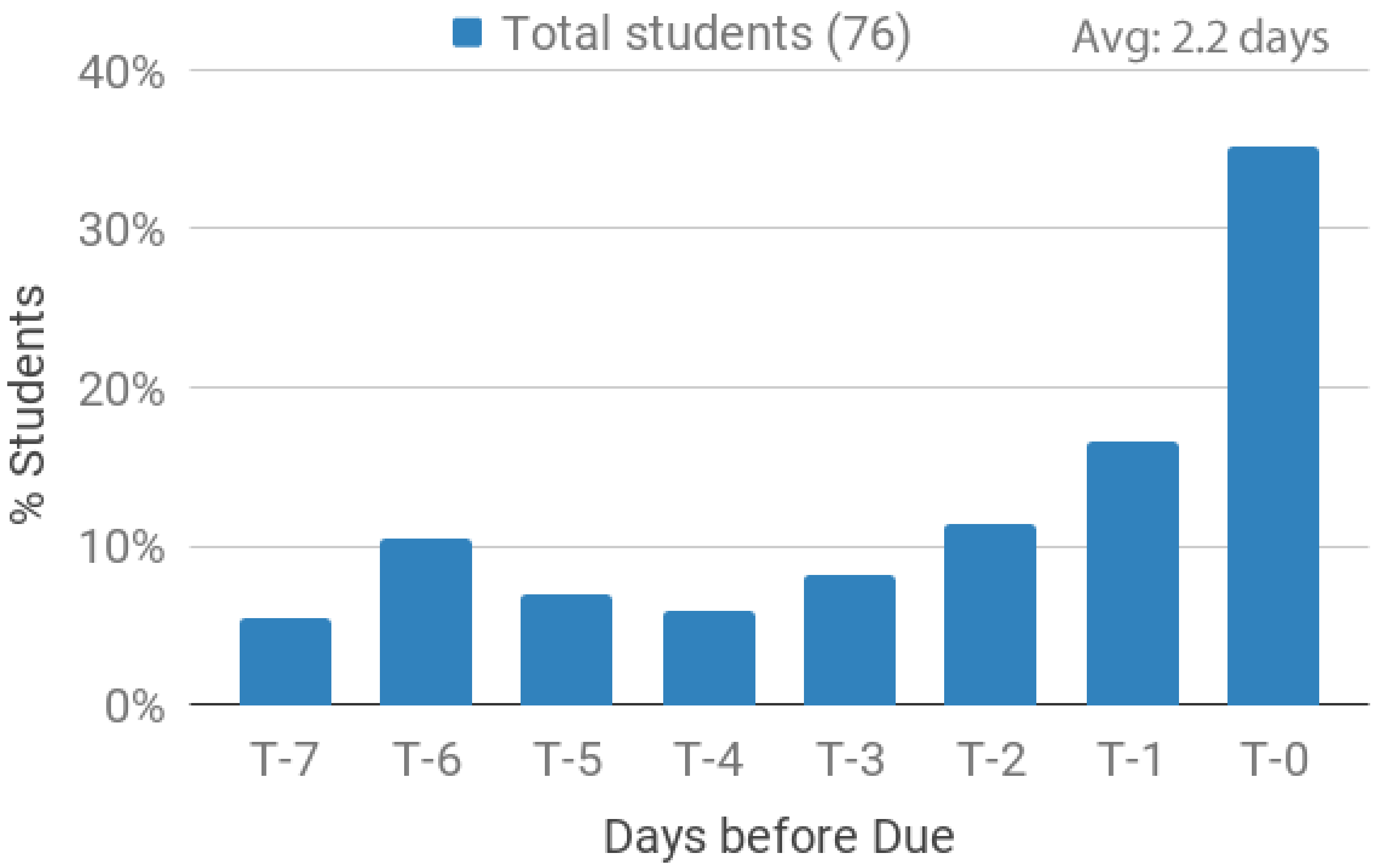
A: About 17 min / MSP



Average time spent per MSP - 17 min / MSP (weeks 1 and 9 excluded).

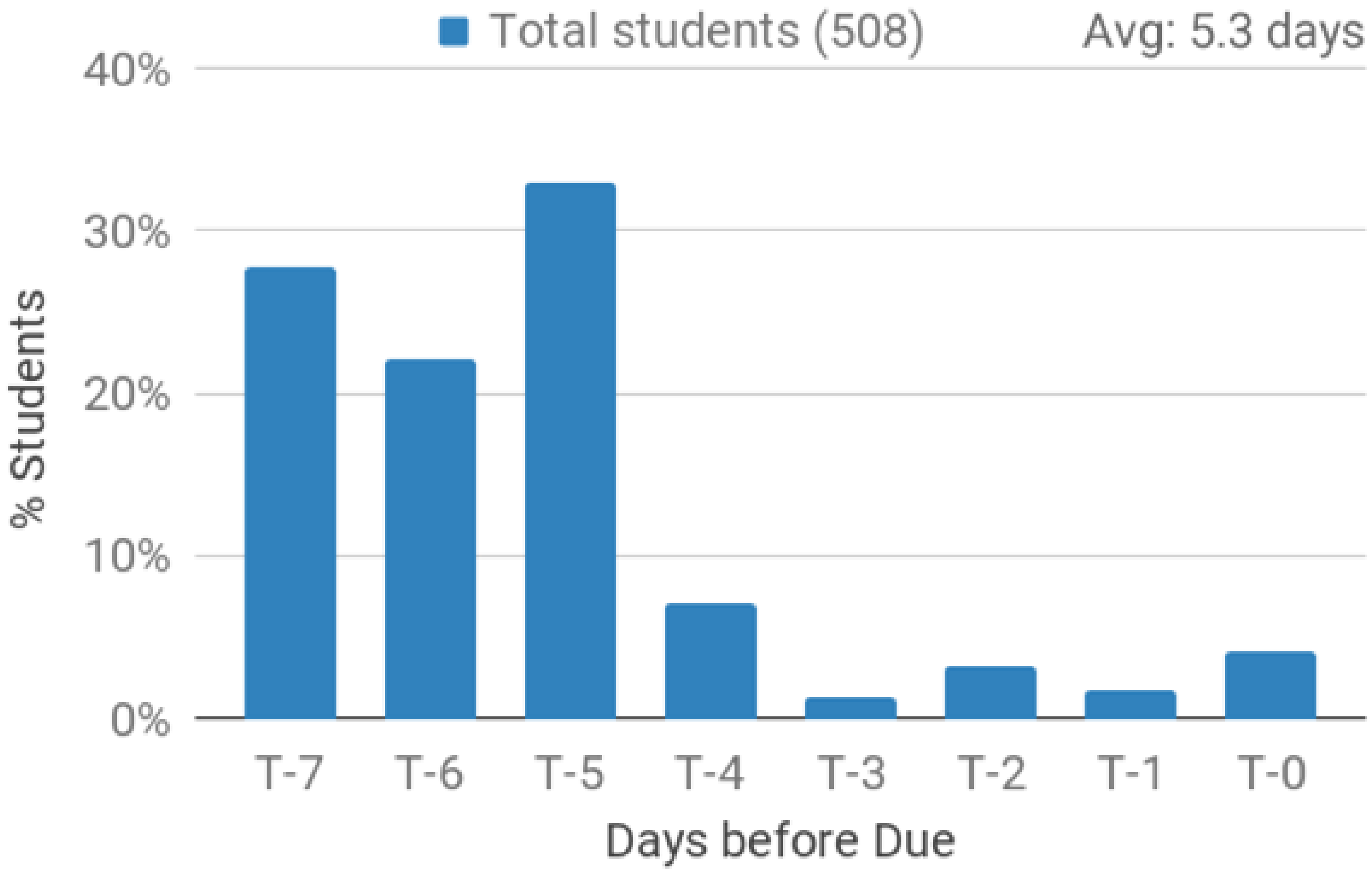
Q: How many days before the due date do students start working on MSPs?

A: MSPs started 2.2 days before due date



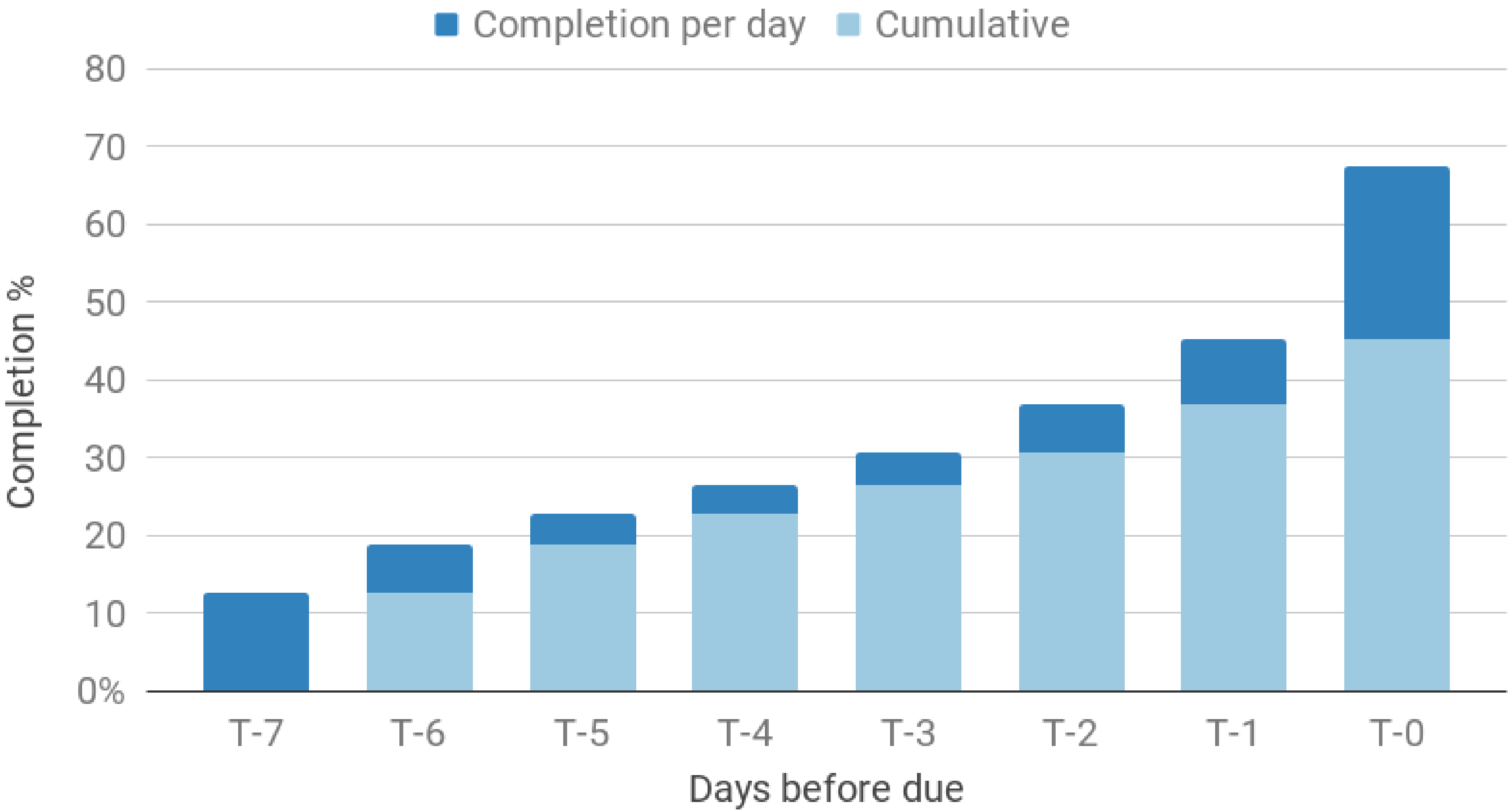
Q: How many days before the due date do students start working on MSPs?

A: With policy adjustment in Fall 2018, started 5.3 days before



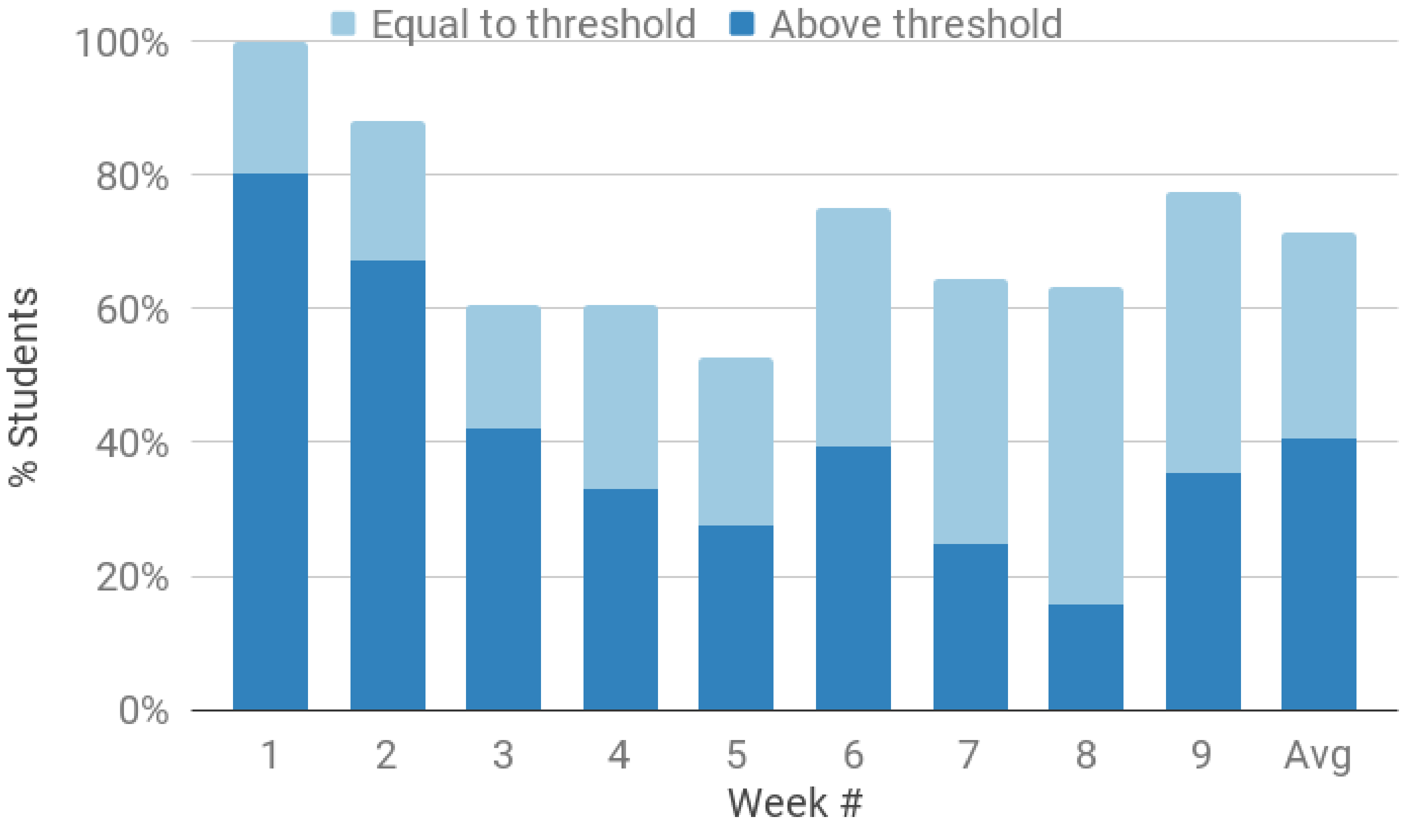
Q: What % of MSPs do students complete each day?

A: Completed 10% of MSPs each day



Q: Given a full-credit threshold, do students complete more MSPs than required?

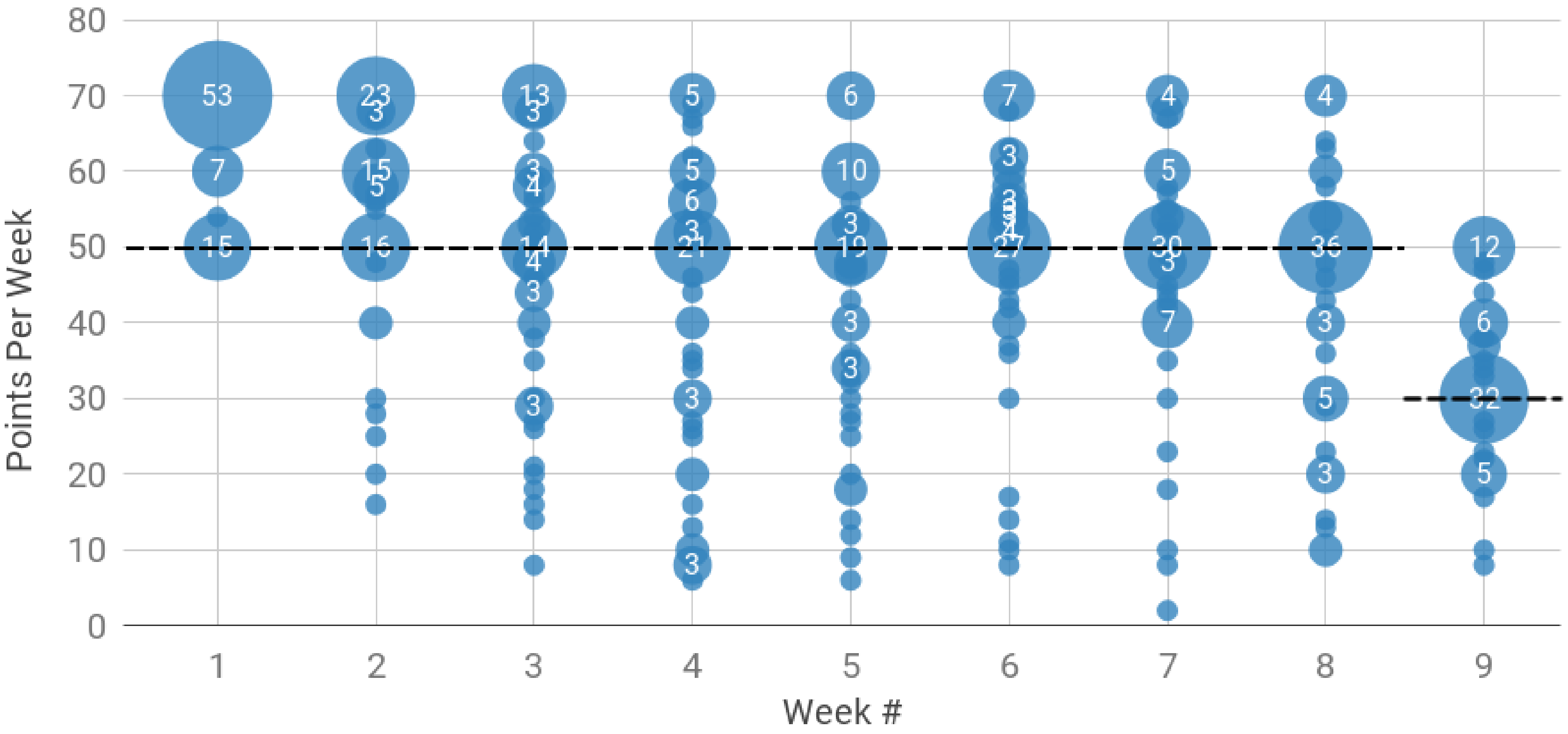
A: 40% of students completed more MSPs than required



No extra credit given for exceeding full-credit threshold

Q: Given a fill-credit threshold, how many points do students score each week?

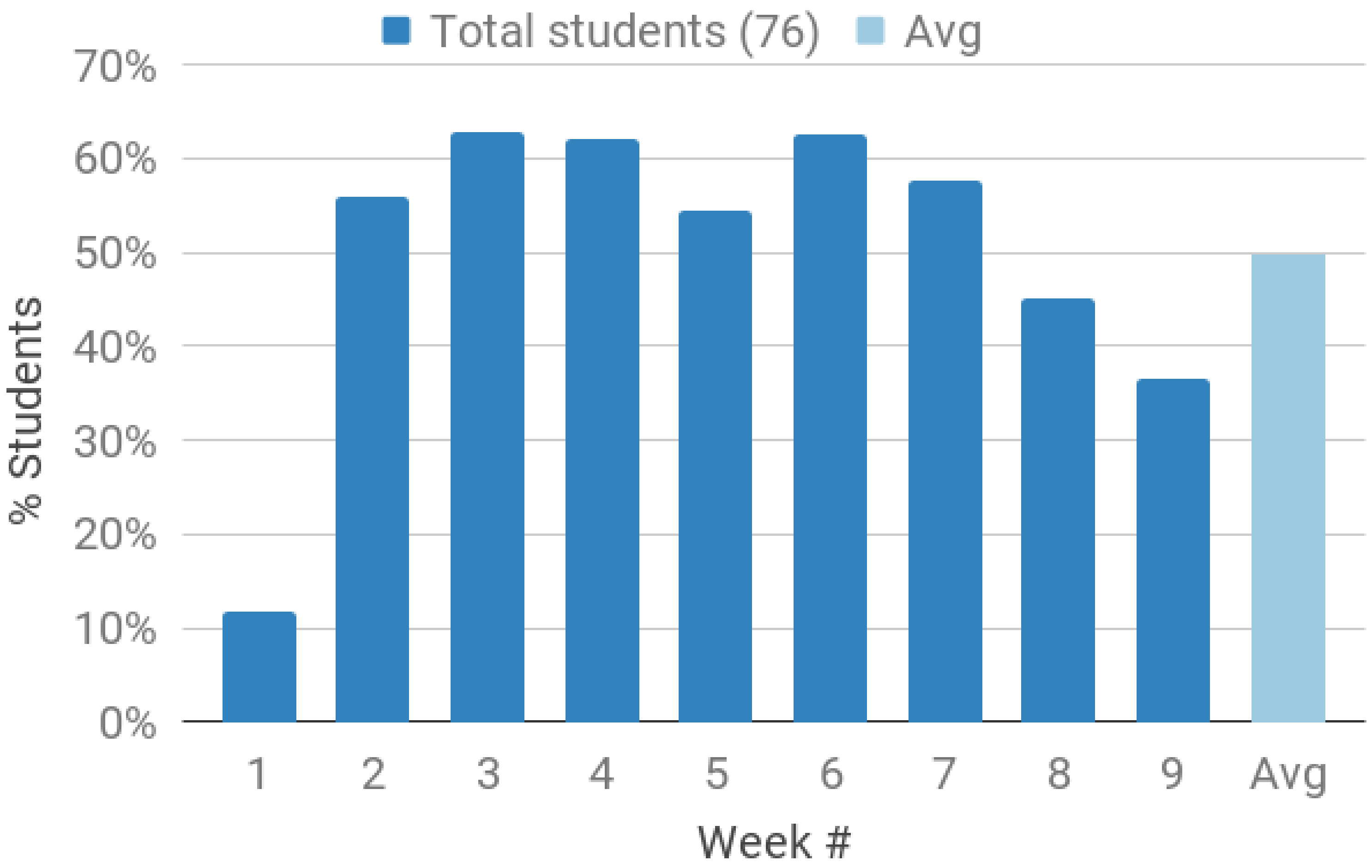
A: Total points per week – Avg 13 more points



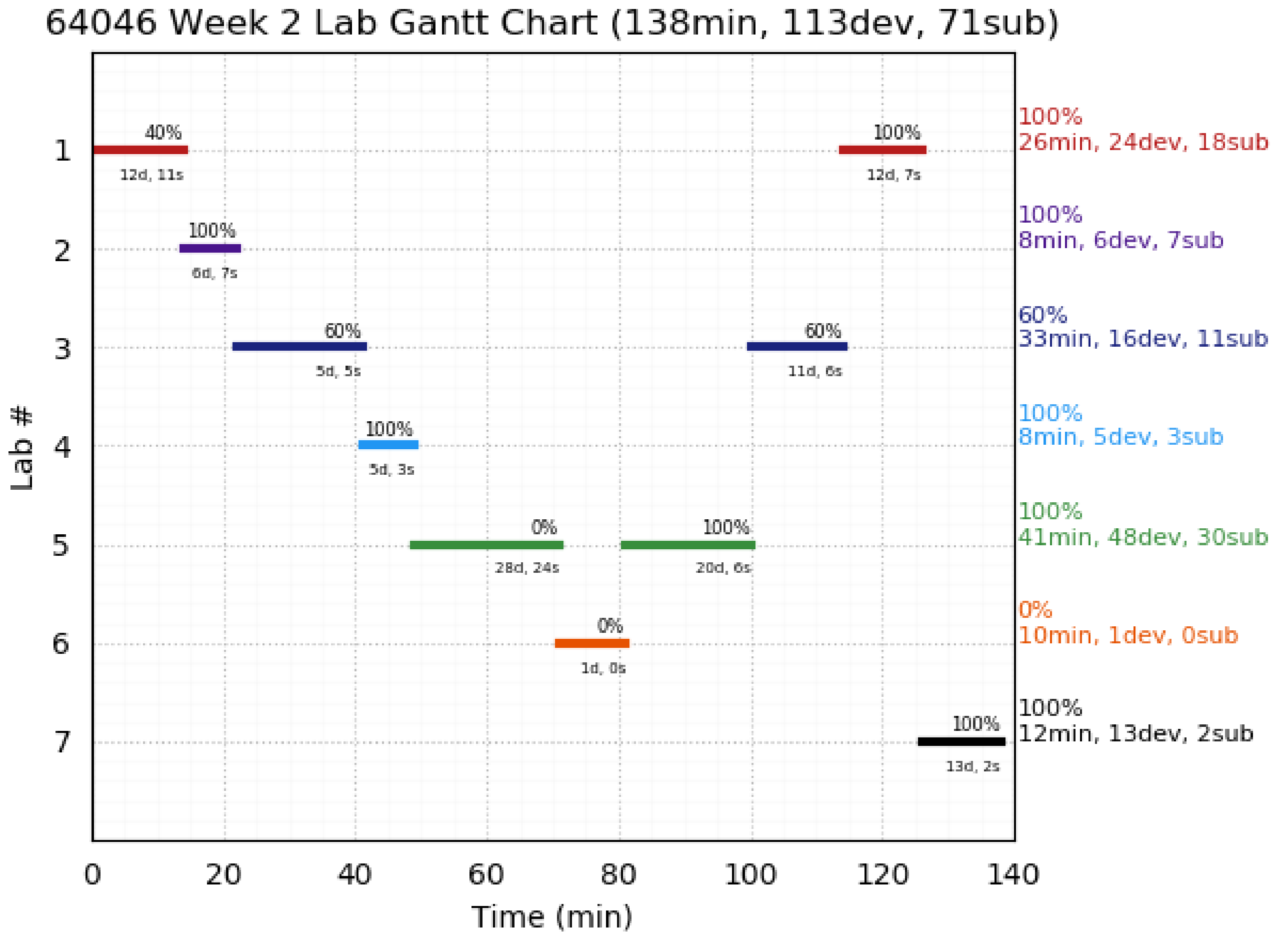
Bubble size represents number of students. Dashed line indicates full-credit threshold.
Students who scored 0 points for a week excluded.

Q: Do students pivot, or switch among MSPs when stuck?

A: Each week, 50% of students pivoted (avg. 1.3 pivots)



Workflow pattern (GANTT chart)



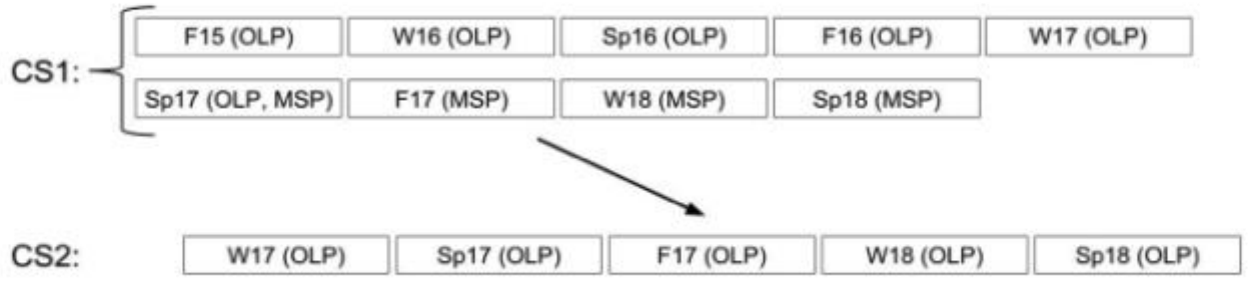
Q: Do students use MSPs to study for exams?

A. Yes, students use MSPs to study for exams

Total number of students	76
Total number of MSPs	61
% of students that used MSPs to study for the midterm	38%
% of students that used MSPs to study for the final	37%
% of students that used MSPs to study for either exam	**54%**
% of MSPs that were used to study for the midterm	97%
% of MSPs that were used to study for the final	90%
% of MSPs that were used to study for either exam	**98%**

Q: *Won't MSP CS1 students do poorly in an OLP CS2?*

A. MSP CS1 students do fine in an OLP CS2, in fact slightly better



CS1:

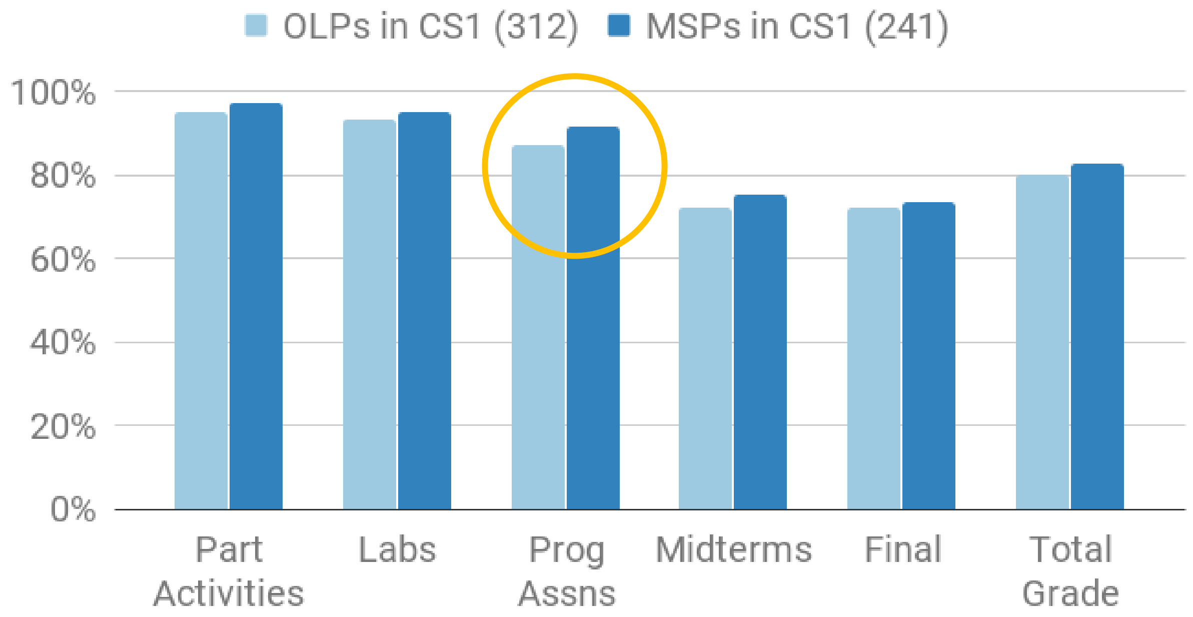
F15 (OLP) | W16 (OLP) | Sp16 (OLP) | F16 (OLP) | W17 (OLP)

Sp17 (OLP, MSP) | F17 (MSP) | W18 (MSP) | Sp18 (MSP)

CS2:

W17 (OLP) | Sp17 (OLP) | F17 (OLP) | W18 (OLP) | Sp18 (OLP)

CS2 scores

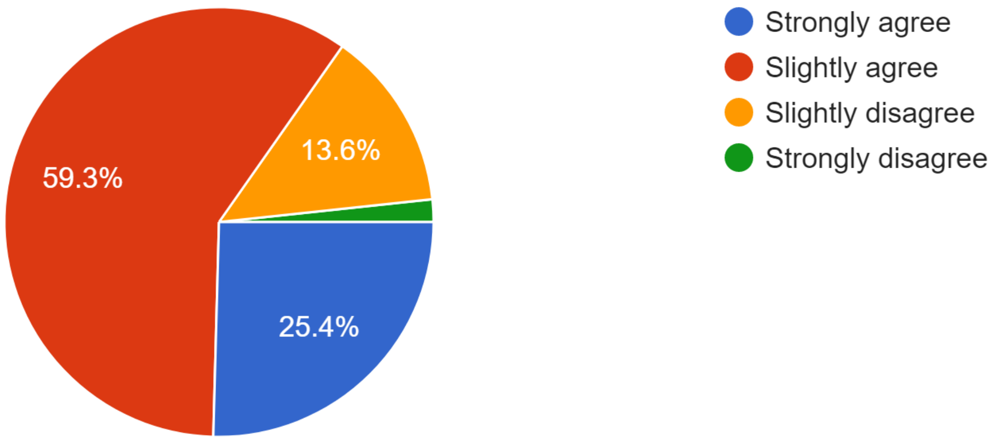


Student feedback

- Good practice
- They are engaging and fun
- Short and simple tasks to improve my coding
- They give me enough practice to know the material

Solving the lab activities is kind of fun.

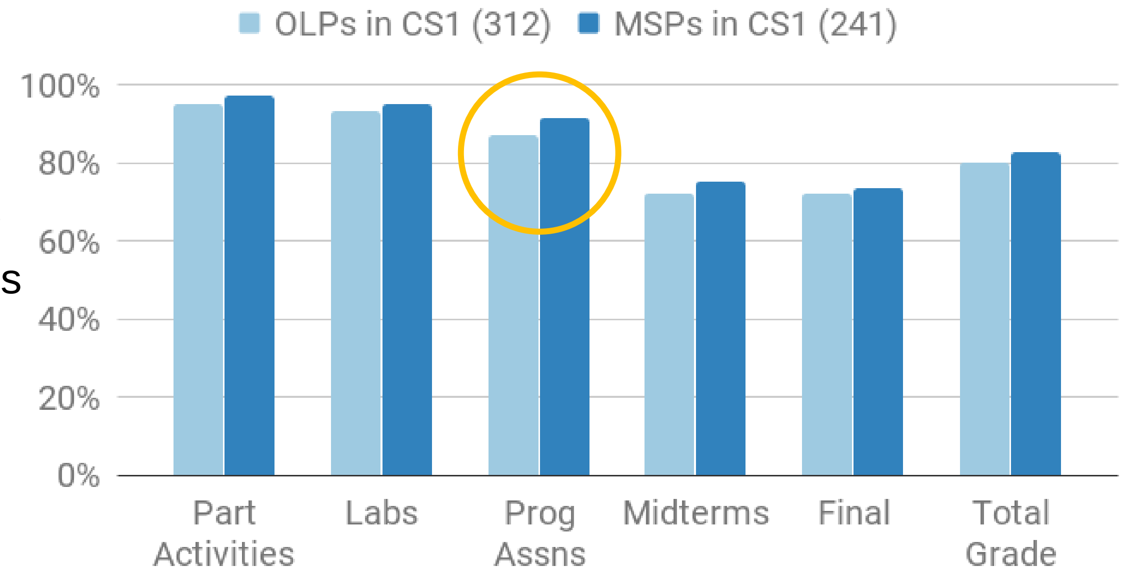
59 responses



Study 2 - Conclusion

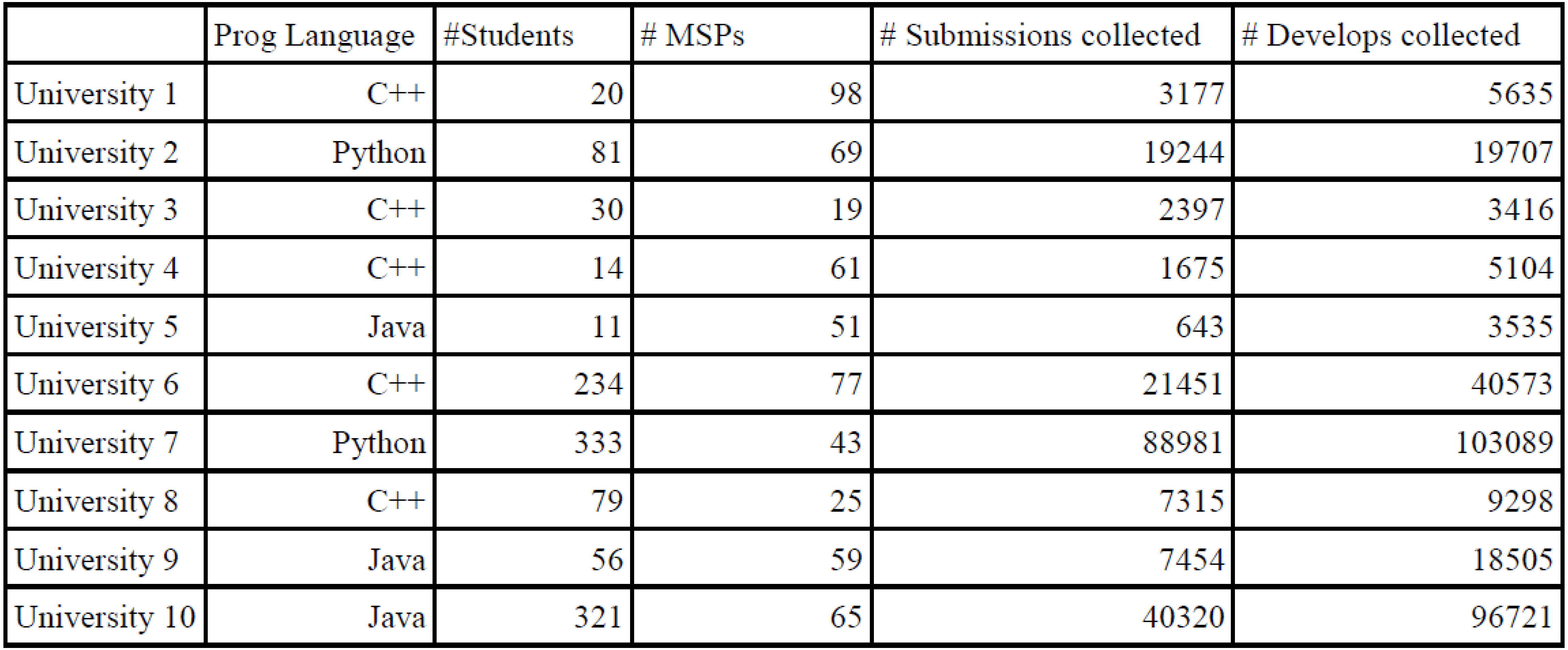
- Students make good use of MSPs
  - Sufficient time
  - Started early
  - Completed more than necessary
  - Pivoted to help selves when stuck
  - Used MSPs to study for exams

CS2 scores



- *And*, MSP CS1 students do <u>just as well </u>as OLP CS1 students in an OLP CS2.

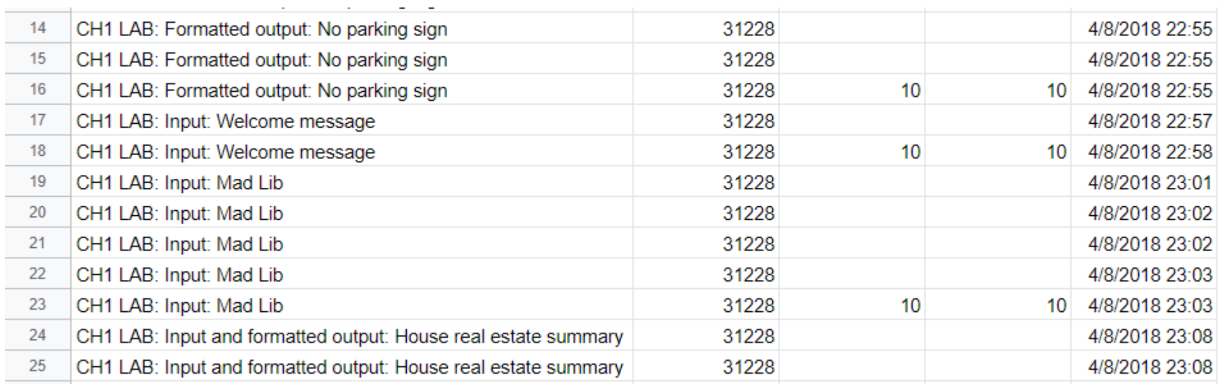
Study 3 – MSP usage analysis - Other universities



	Prog Language	#Students	# MSPs	# Submissions collected	# Develops collected
University 1	C++	20	98	3177	5635
University 2	Python	81	69	19244	19707
University 3	C++	30	19	2397	3416
University 4	C++	14	61	1675	5104
University 5	Java	11	51	643	3535
University 6	C++	234	77	21451	40573
University 7	Python	333	43	88981	103089
University 8	C++	79	25	7315	9298
University 9	Java	56	59	7454	18505
University 10	Java	321	65	40320	96721

Data collection

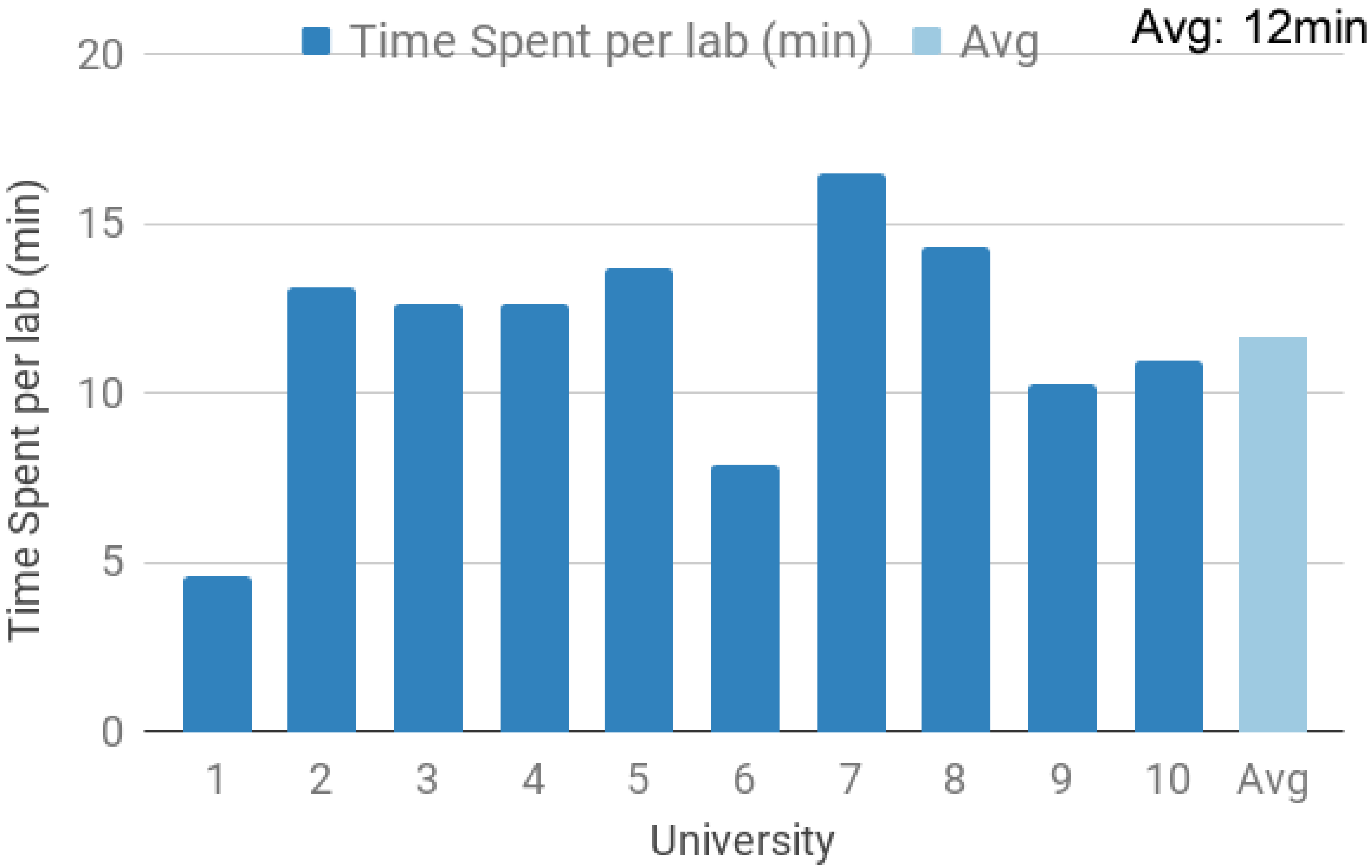
- 10 universities, over 1,000 students included
- Multiple programming languages
  - C++, Python, Java
- Used zyLabs from zyBooks (MSPs)
- Collected:
  - 300,000 develop runs, 190,000 submissions



	labID	userID	score	maxScore	timestamp
14	CH1 LAB: Formatted output: No parking sign	31228			4/8/2018 22:55
15	CH1 LAB: Formatted output: No parking sign	31228			4/8/2018 22:55
16	CH1 LAB: Formatted output: No parking sign	31228	10	10	4/8/2018 22:55
17	CH1 LAB: Input: Welcome message	31228			4/8/2018 22:57
18	CH1 LAB: Input: Welcome message	31228	10	10	4/8/2018 22:58
19	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:01
20	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:02
21	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:02
22	CH1 LAB: Input: Mad Lib	31228			4/8/2018 23:03
23	CH1 LAB: Input: Mad Lib	31228	10	10	4/8/2018 23:03
24	CH1 LAB: Input and formatted output: House real estate summary	31228			4/8/2018 23:08
25	CH1 LAB: Input and formatted output: House real estate summary	31228			4/8/2018 23:08

Q: How much time do students spend working on each MSP?

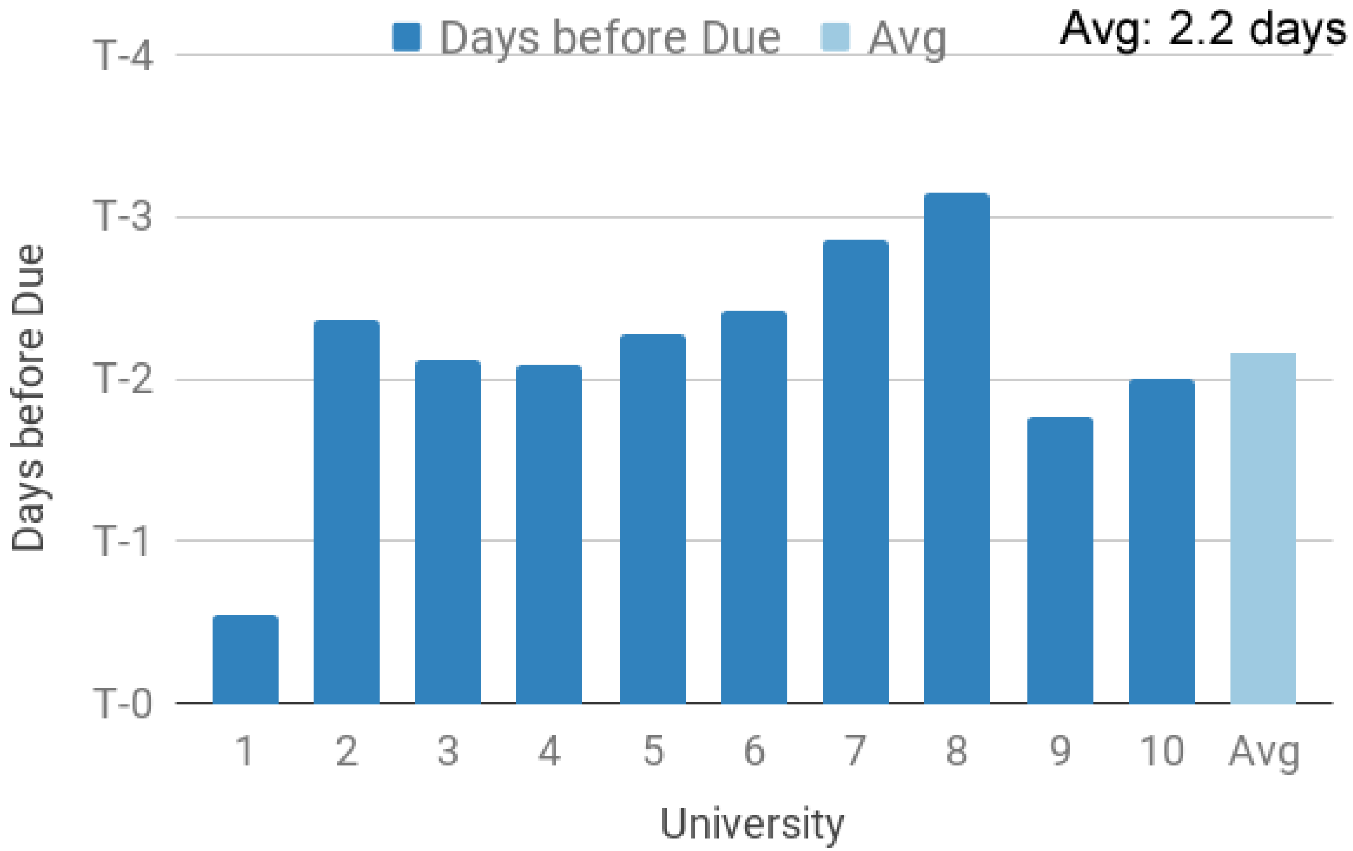
A: At least 12 min / MSP



UCR average time spent per MSP - 17 min / MSP

Q: How many days before the due date do students start working on MSPs?

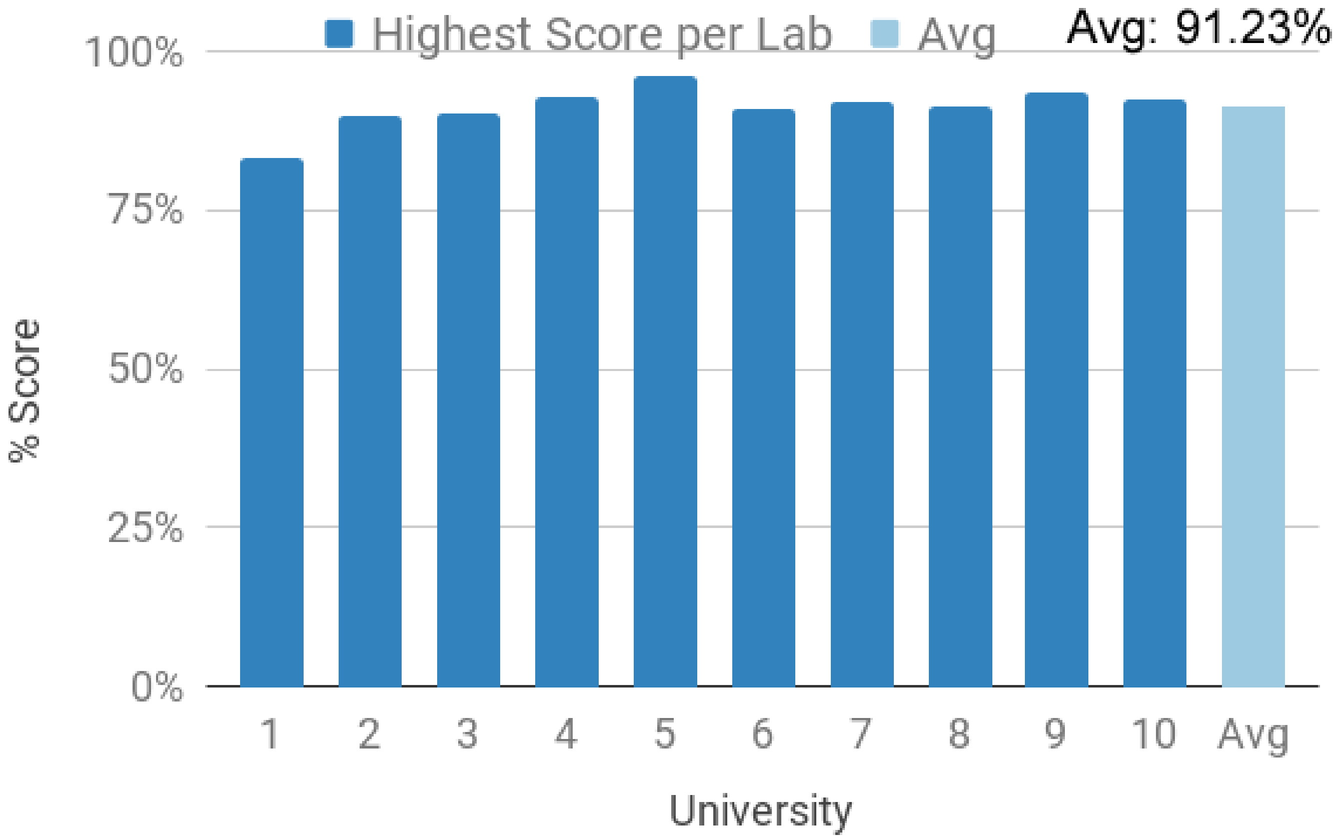
A: MSPs started 2.2 days before due date



UCR average days before due - 2.2 days

Q: What score do students earn per MSP?

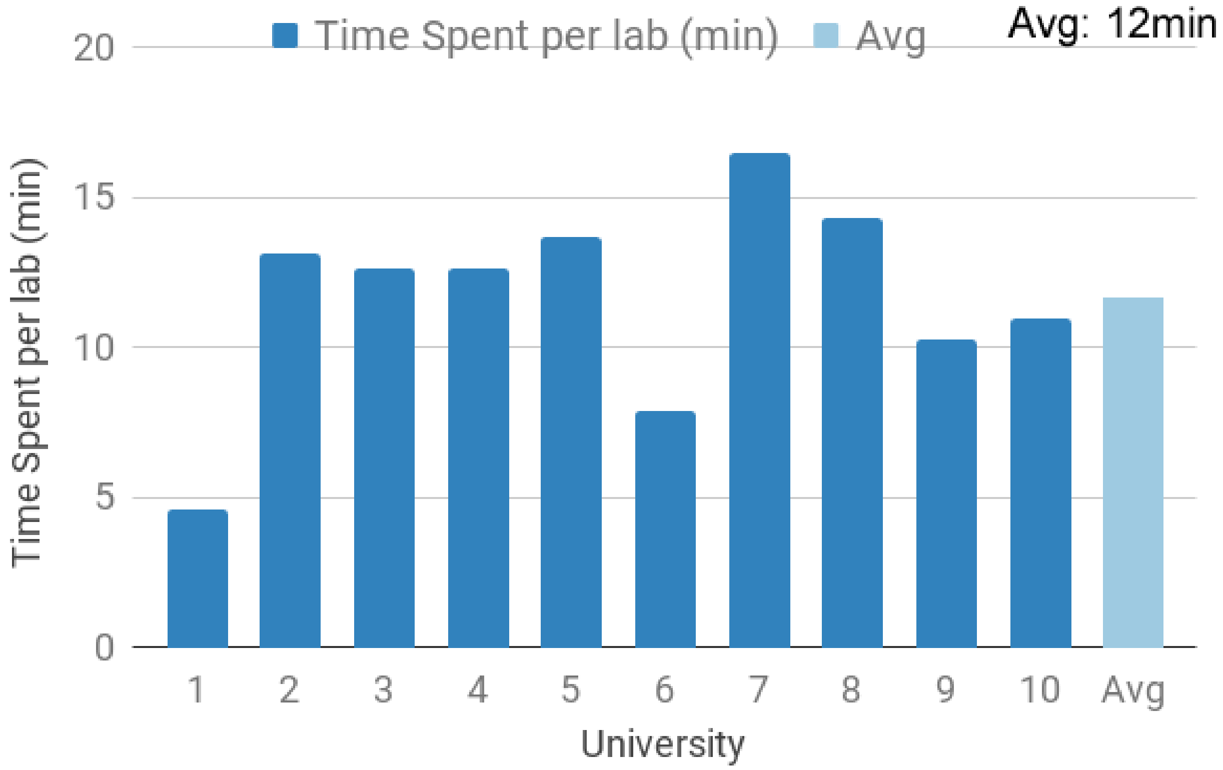
A: Students score an average of 91% per MSPs



UCR score per MSP – 89.45%

Study 3 - Conclusion

- Similar results from other universities
  - Spending sufficient time
  - Starting early
  - Completing most MSPs

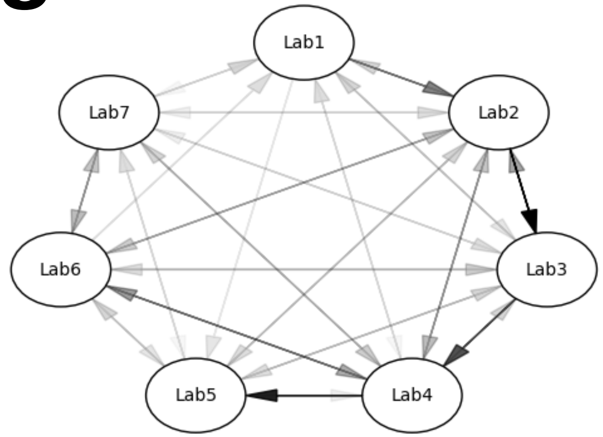


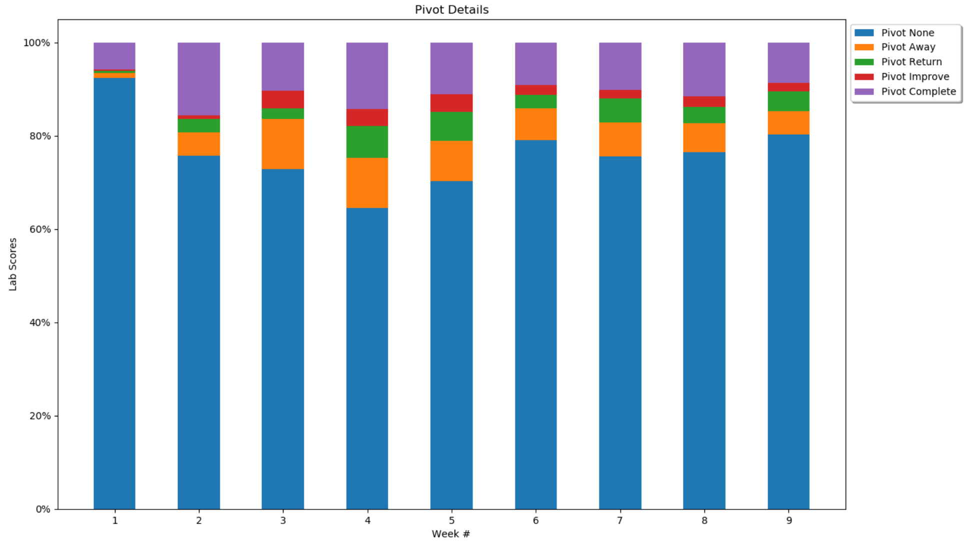
Future work

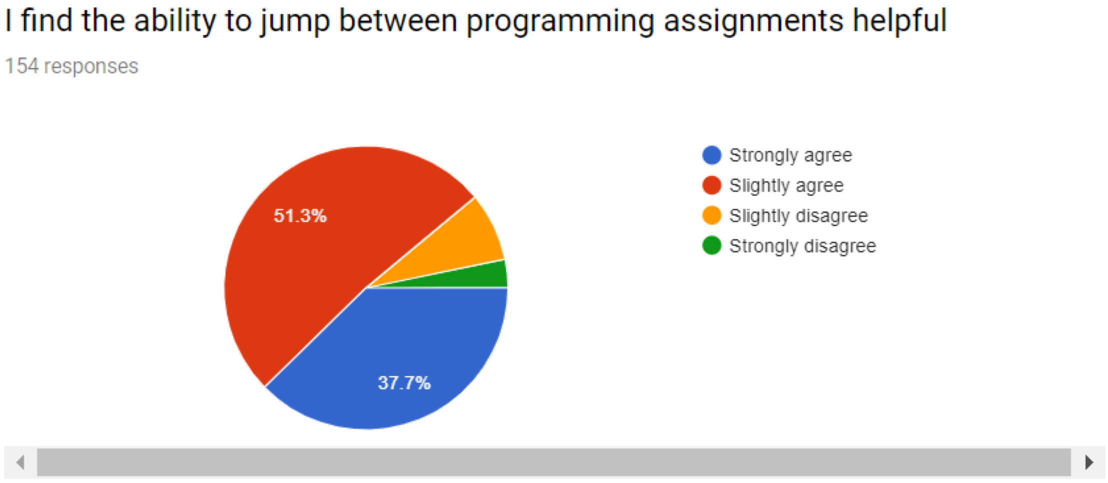
- Gain better insight on student pivoting
- Gain better insight on using a threshold with MSPs
- Understand the impacts of allowing collaboration

Gain better insight on student pivoting

- What are general pivot patterns?
- Do students return to pivoted MSPs?
- How do students feel about the ability to pivot?



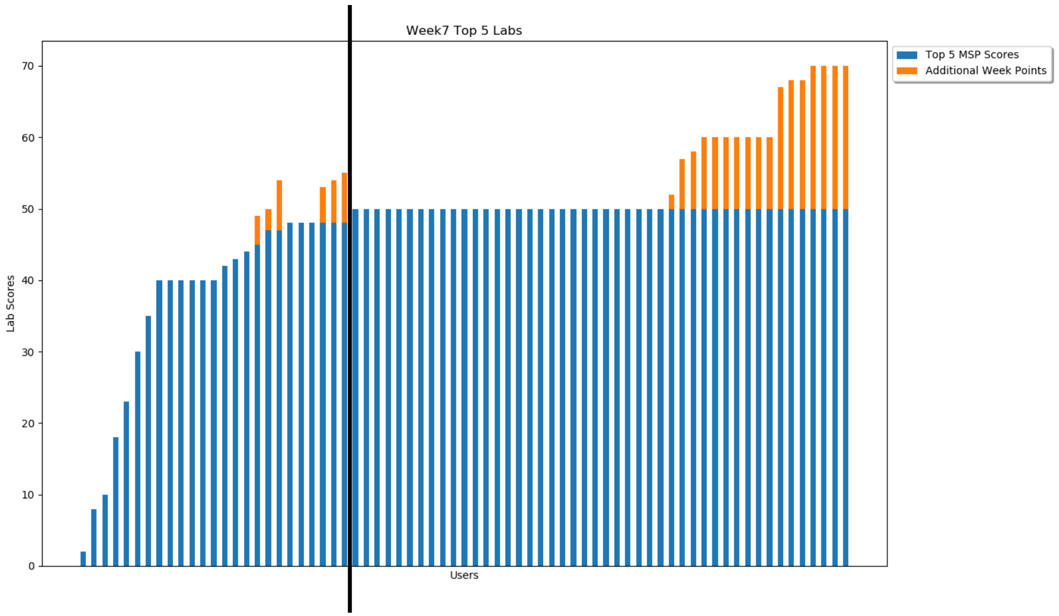



I find the ability to jump between programming assignments helpful

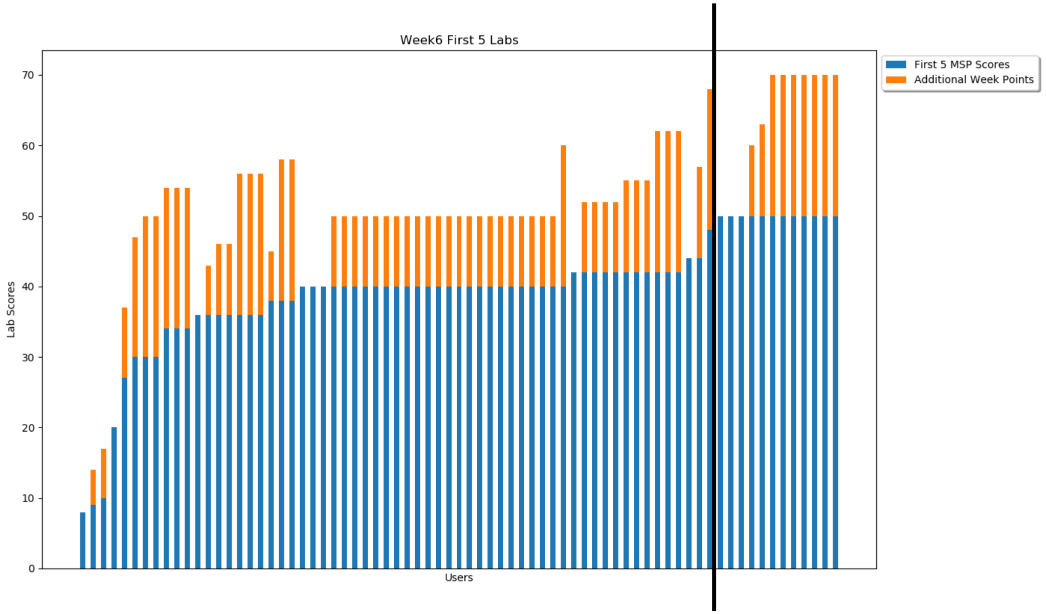
154 responses

Gain better insight on using a full-credit threshold

- Given a full-credit threshold, how do students earn their points?



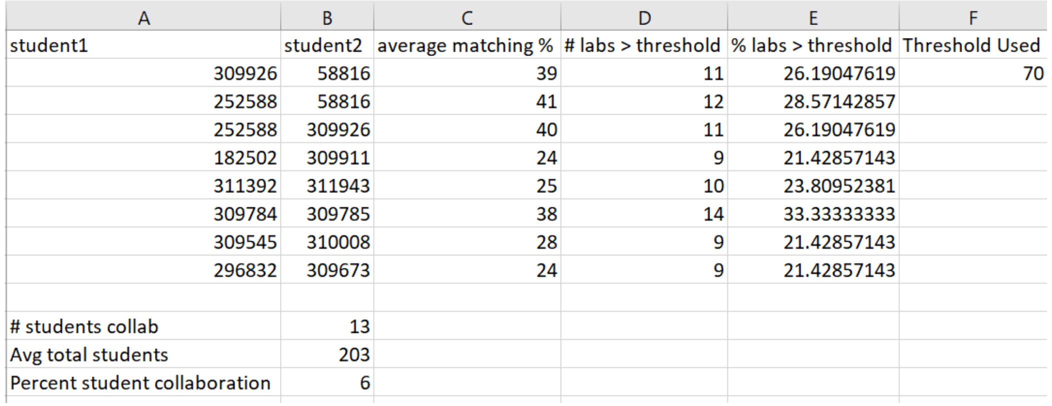
Top 5 Labs



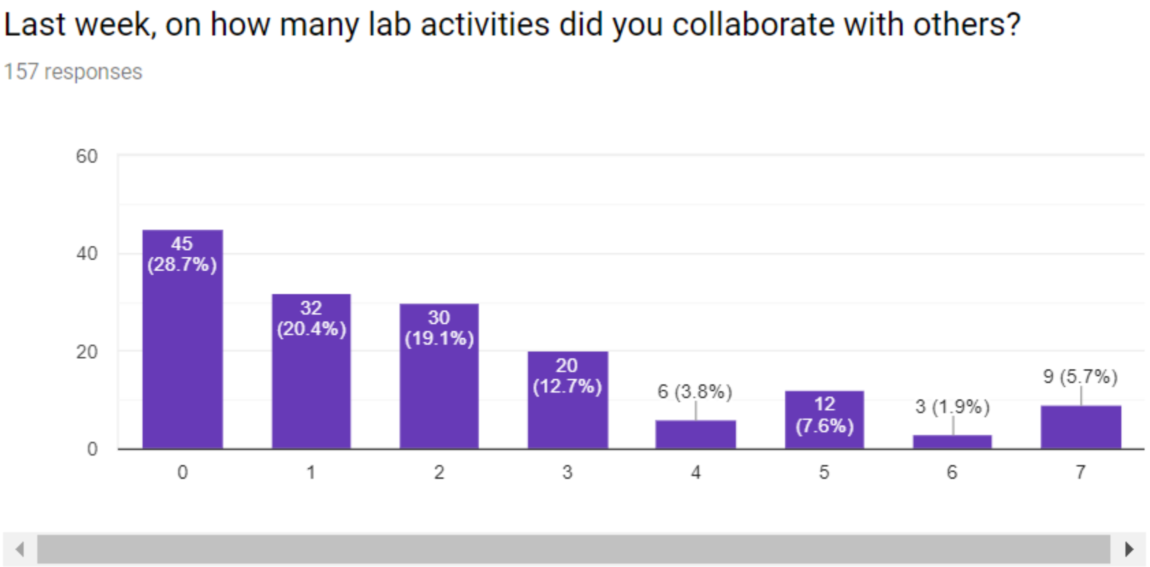
First 5 Labs

Impacts of allowing collaboration

- Are students collaborating?
  - MOSS
  - Polls
- Who are students collaborating with?

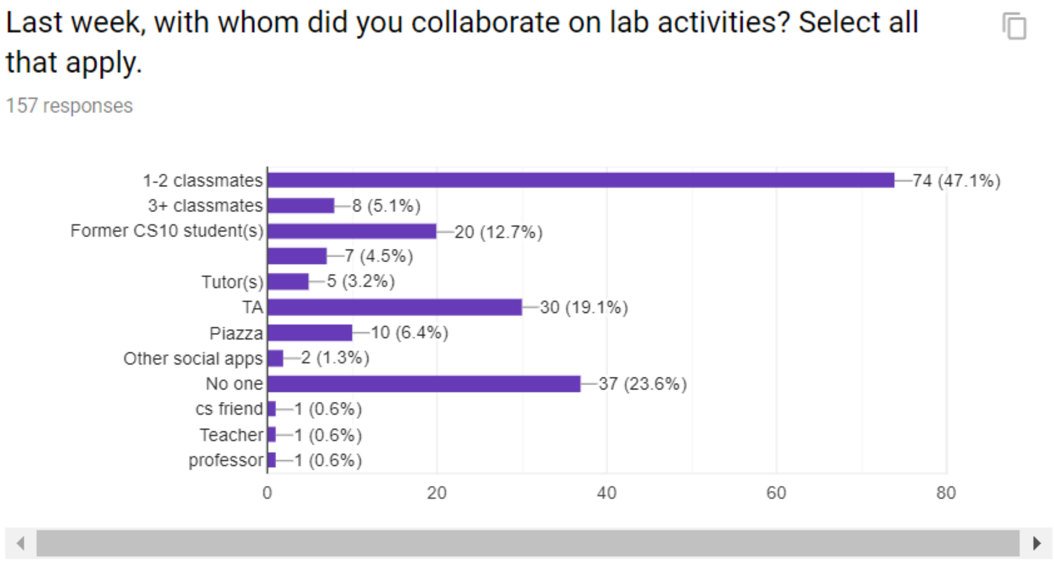


A	B	C	D	E	F
student1	student2	average matching %	# labs > threshold	% labs > threshold	Threshold Used
309926	58816	39	11	26.19047619	70
252588	58816	41	12	28.57142857	
252588	309926	40	11	26.19047619	
182502	309911	24	9	21.42857143	
311392	311943	25	10	23.80952381	
309784	309785	38	14	33.33333333	
309545	310008	28	9	21.42857143	
296832	309673	24	9	21.42857143	
# students collab	13				
Avg total students	203				
Percent student collaboration	6				



Last week, on how many lab activities did you collaborate with others?

157 responses



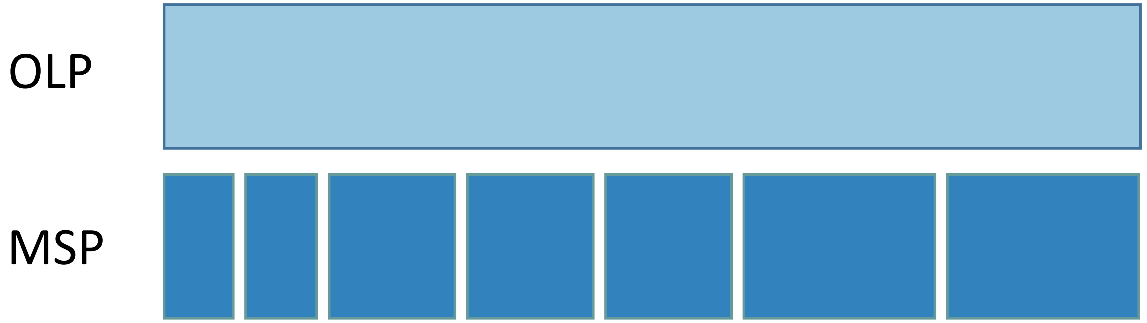
Last week, with whom did you collaborate on lab activities? Select all that apply.

157 responses

Conclusion

- Teaching CS1 via MSPs show positive results
  - Students are more satisfied in CS1
  - Student grades are not harmed, but slightly improve on exams

- Students make good use of MSPs
  - Sufficient time
  - Started early
  - Completed more than necessary
  - Pivoted to help selves when stuck
  - Used MSPs to study for exams
  - MSP CS1 students do just as well as OLP CS1 students in an OLP CS2



- Other universities show similar results

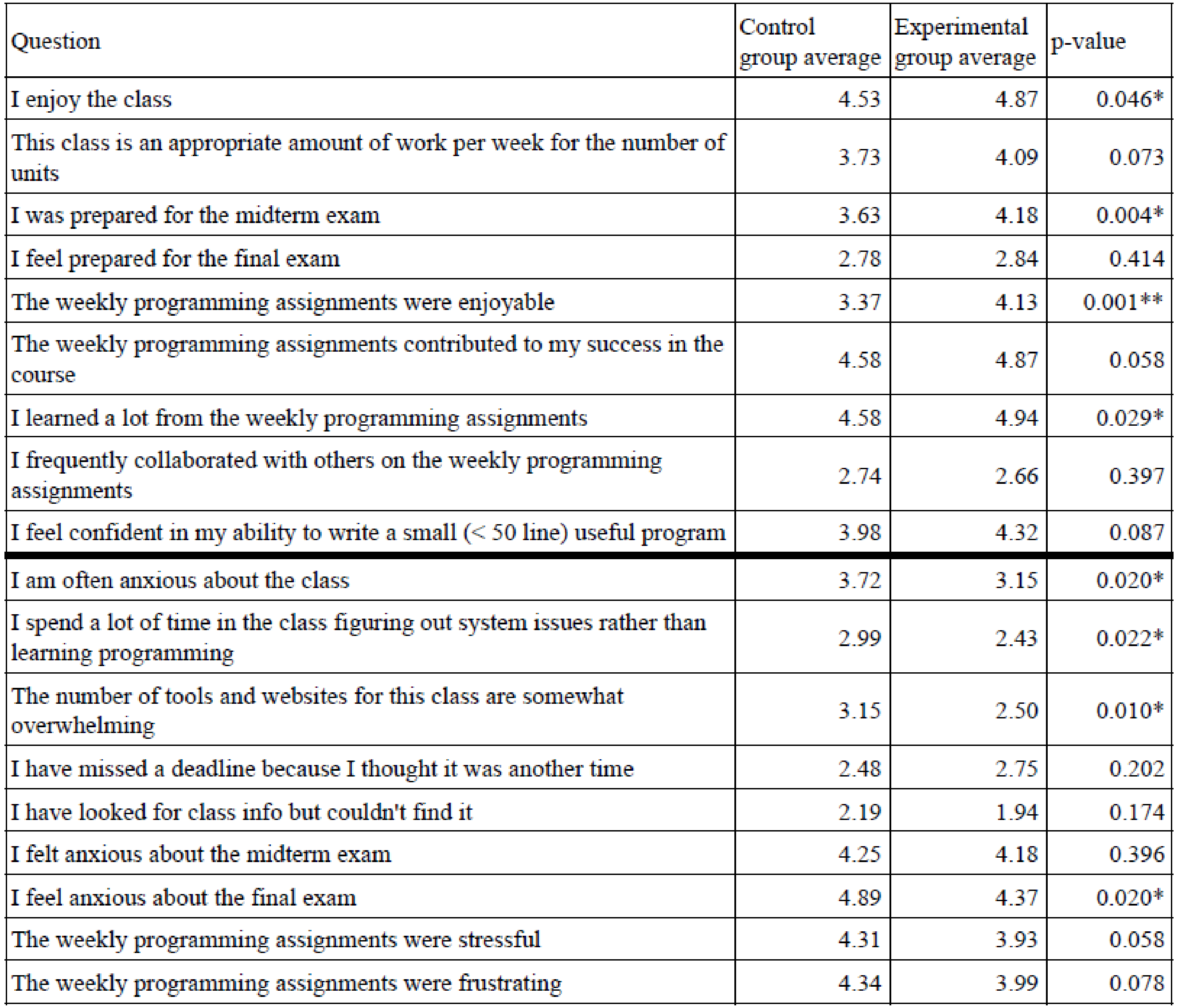
References and Publications

- References
  - Watson, C. and Li, F. “Failure Rates in Introductory Programming Revisited, ” iTiCSE, 2014 http://dro.dur.ac.uk/19223/1/19223.pdf%3FDDD10%2Bd74ks0%2Bdcs0lw

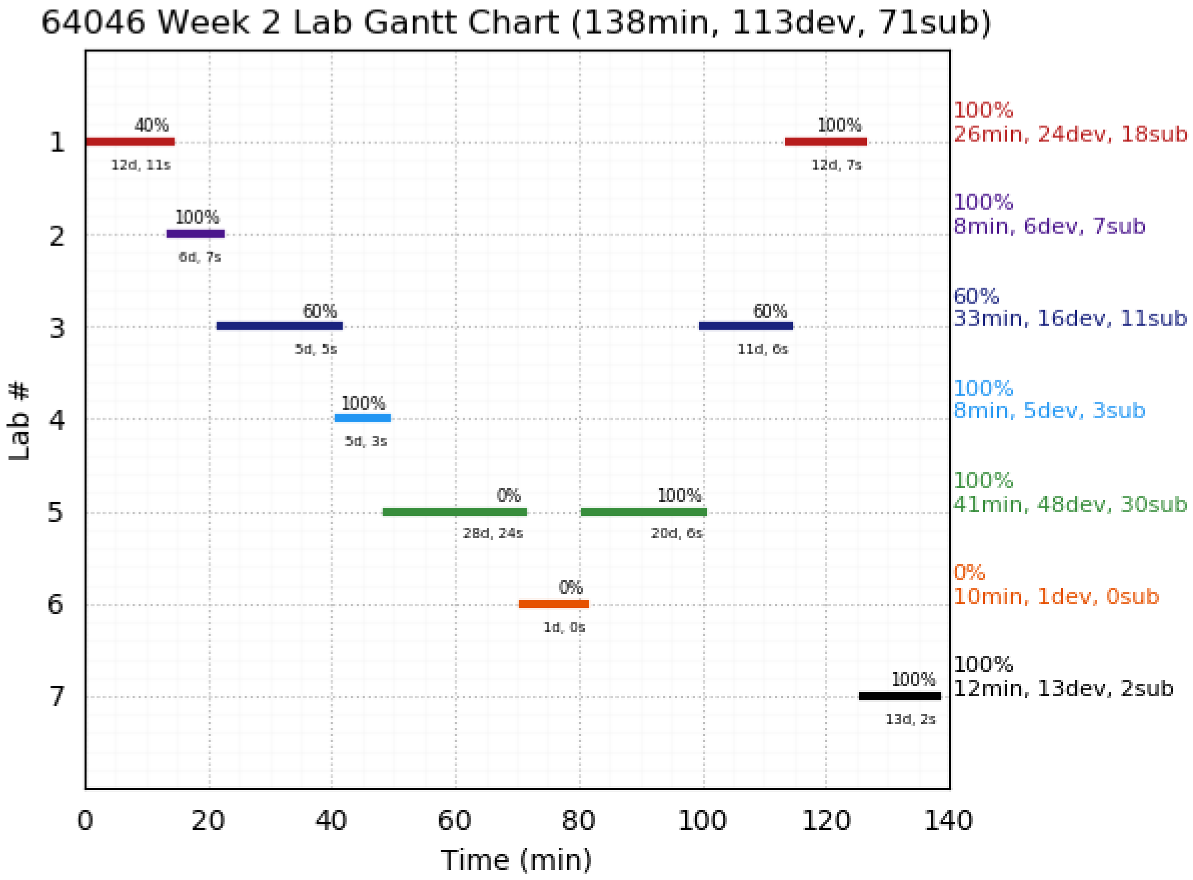
- Publications
  - J.M. Allen, F. Vahid, A. Edgcomb, K. Downey, and K. Miller. An Analysis of Using Many Small Programs in CS1, ACM SIGCSE Technical Symposium on Computer Science Education, 2019.
  - J.M. Allen, F. Vahid, K. Downey, and A. Edgcomb. Weekly Programs in a CS1 Class: Experiences with Auto-graded Many-small Programs (MSP), Proceedings of ASEE Annual Conference, 2018.
  - J.M. Allen, F. Vahid, S. Salehain, and A. Edgcomb. Serious games for building skills in computing and engineering, Proceedings of ASEE Annual Conference, 2017.
  - F. Vahid, J.M. Allen, and A. Edgcomb. Web-based games to master core skills in introductory college mathematics, Joint Mathematics Meetings, 2017, abstract.

- Pending Publications
  - J.M. Allen, F. Vahid, K. Downey, K. Miller, and A. Edgcomb. Many Small Programs in CS1: Usage Analysis from Multiple Universities, Proceedings of ASEE Annual Conference, 2019.
  - J.M. Allen, F. Vahid, A. Edgcomb, K. Downey, and K. Miller. Auto Graded Many Small Programs (MSPs) in CS1 for Improvements in Learning, Confidence, and Stress, ACM Transactions on Computing Education.
  - J.M. Allen, F. Vahid, K. Downey, K. Miller, and A. Edgcomb. Analyzing Pivoting Among Weekly Many Small Programs in a CS1 Course.

- Other Talks
  - J.M. Allen, F. Vahid, K. Downey, and A. Edgcomb. Weekly Programs in CS 1: Experiences with Many-Small Auto-Graded Programs, UCR Symposium, 2018.



Question	Control group average	Experimental group average	p-value
I enjoy the class	4.53	4.87	0.046*
This class is an appropriate amount of work per week for the number of units	3.73	4.09	0.073
I was prepared for the midterm exam	3.63	4.18	0.004*
I feel prepared for the final exam	2.78	2.84	0.414
The weekly programming assignments were enjoyable	3.37	4.13	0.001**
The weekly programming assignments contributed to my success in the course	4.58	4.87	0.058
I learned a lot from the weekly programming assignments	4.58	4.94	0.029*
I frequently collaborated with others on the weekly programming assignments	2.74	2.66	0.397
I feel confident in my ability to write a small (< 50 line) useful program	3.98	4.32	0.087
I am often anxious about the class	3.72	3.15	0.020*
I spend a lot of time in the class figuring out system issues rather than learning programming	2.99	2.43	0.022*
The number of tools and websites for this class are somewhat overwhelming	3.15	2.50	0.010*
I have missed a deadline because I thought it was another time	2.48	2.75	0.202
I have looked for class info but couldn't find it	2.19	1.94	0.174
I felt anxious about the midterm exam	4.25	4.18	0.396
I feel anxious about the final exam	4.89	4.37	0.020*
The weekly programming assignments were stressful	4.31	3.93	0.058
The weekly programming assignments were frustrating	4.34	3.99	0.078


64046 Week 2 Lab Gantt Chart (138min, 113dev, 71sub)
Lab #
1
2
3
4
5
6
7
Time (min)
0
20
40
60
80
100
120
140
40%
12d, 11s
100%
12d, 7s
100%
6d, 7s
60%
5d, 5s
60%
11d, 6s
100%
5d, 3s
0%
28d, 24s
100%
20d, 6s
0%
1d, 0s
100%
13d, 2s
100%
26min, 24dev, 18sub
100%
8min, 6dev, 7sub
60%
33min, 16dev, 11sub
100%
8min, 5dev, 3sub
100%
41min, 48dev, 30sub
0%
10min, 1dev, 0sub
100%
12min, 13dev, 2sub