Weekly Programs in CS 1: Experiences with Many-Small Auto-Graded Programs

BY JOE MICHAEL ALLEN, FRANK VAHID, KELLY DOWNEY, AND ALEX EDGCOMB

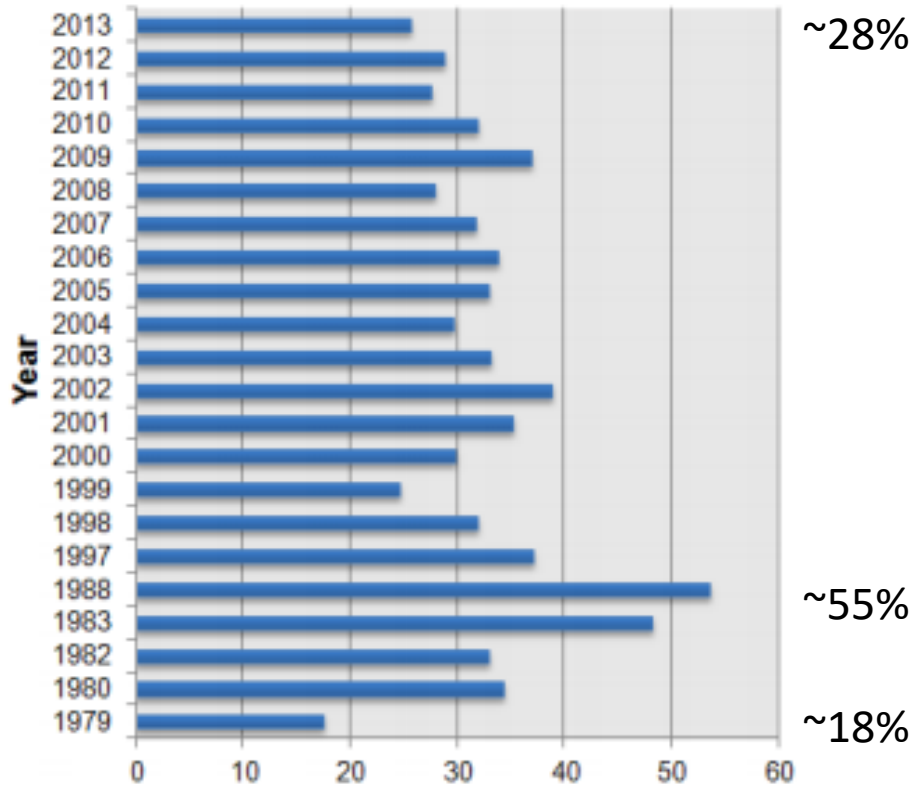
DEPT. OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF CALIFORNIA, RIVERSIDE

This work was supported by the U.S. Dept. of Education (GAANN fellowship) and by Google

Problem

Mean Percentage of Non-Passing Students in CS1



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

CS 1 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

Wk 6 Program: Authoring assistant

(1) Prompt the user to enter a string of their choosing. Store the text in a string. Output the string. (1 pt)

(2) Implement a printMenu() method, which outputs a menu of user options for analyzing/editing the string, and returns the user's entered menu option. Each option is represented by a single character.
If an invalid character is entered, continue to prompt for a valid choice. *Hint: Implement Quit before implementing other options.* Call printMenu() in the main() method. Continue to call printMenu() until the user enters q to Quit. (3 pts)

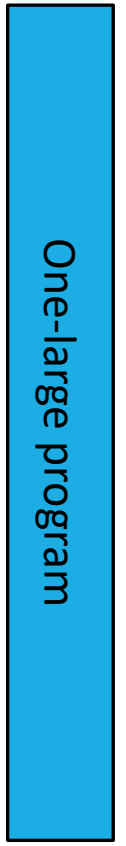
(3) Implement the getNumOfNonWSCharacters() method. getNumOfNonWSCharacters() has a string as a parameter and returns the number of characters in the string, excluding all whitespace. Call getNumOfNonWSCharacters() in the main() method. (4 pts)

(4)…(5)...(6)...(7)...

Traditional approach

One-large program:

- One-large assignment each week
- Teach many concepts
- Multiple parts
- More text
- Larger solution size

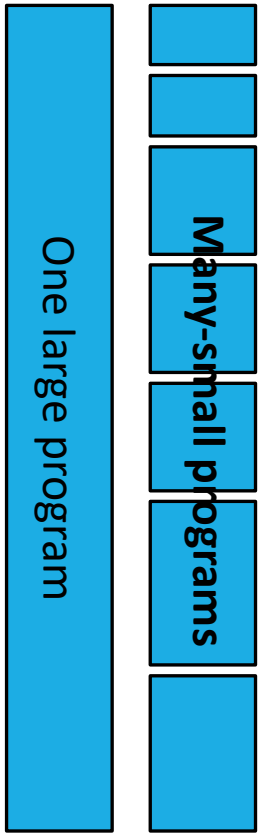


3.5 hrs

Our solution - MSPs

Many-small programs

- Multiple small programs each week
- Teach one specific concept
- Short
- Minimal text
- Smaller solution size
- Total time about same (~3.5 hrs)



Benefits

- Less intimidating
- Simpler labs build confidence
- Pivot if stuck
- More practice

Made possible by program auto-grader (with easy web-based creation so any instructor or TA can create/modify)

MSPs - prompt

5.13 CH5 LAB: Print name in reverse

Edit lab Share Note

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

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)

Upload a solution

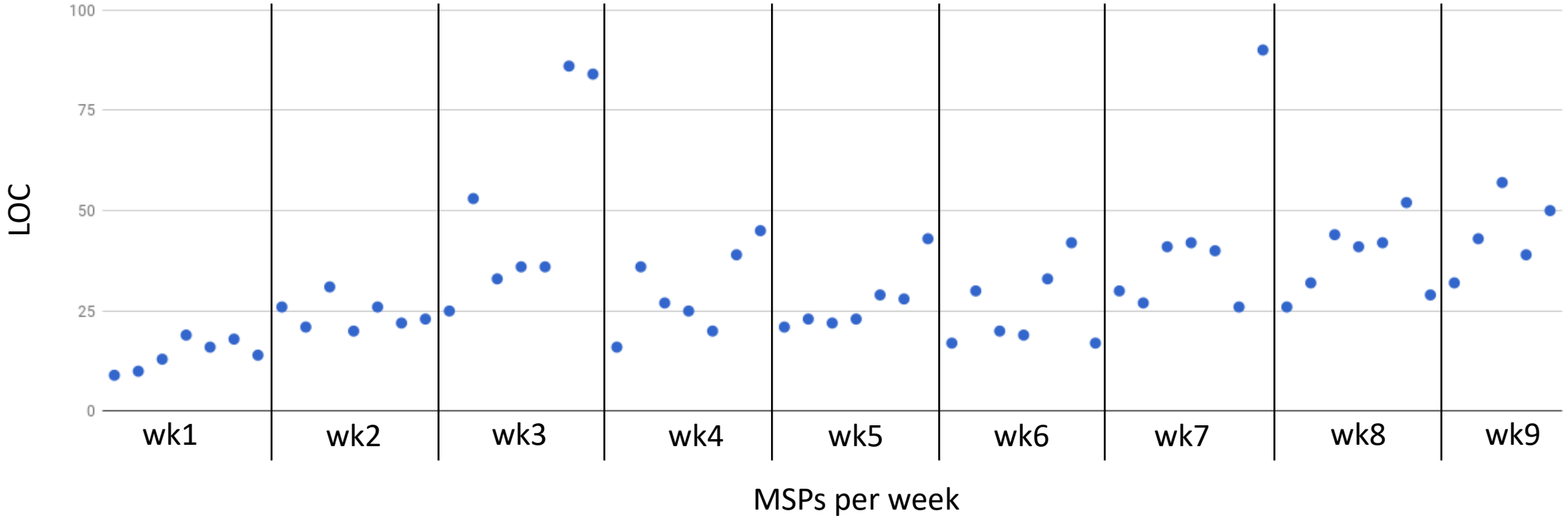
```
#include <iostream>
using namespace std;

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
```

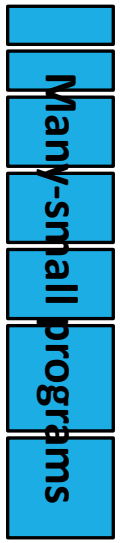
Test cases:

- 10 points per MSP
- Input/output tests
- Unit tests

Experiment

CS 1 course at UCR during Spring 2017; 10 week quarter

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



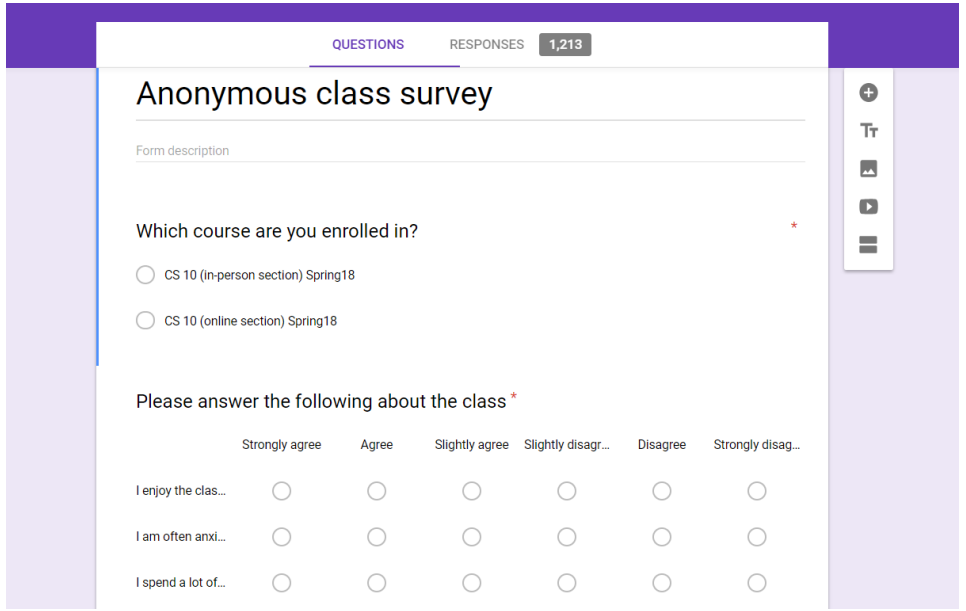
2 sections; 166 students
Instructor 1
No collaboration
Programming assignments: 25%, Midterm: 20%

1 section; 77 students
Instructor 2
Yes collaboration
Programming assignments: 15%, Midterm: 30%

Methods

Student surveys (“Stress survey”)

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



Student outcomes

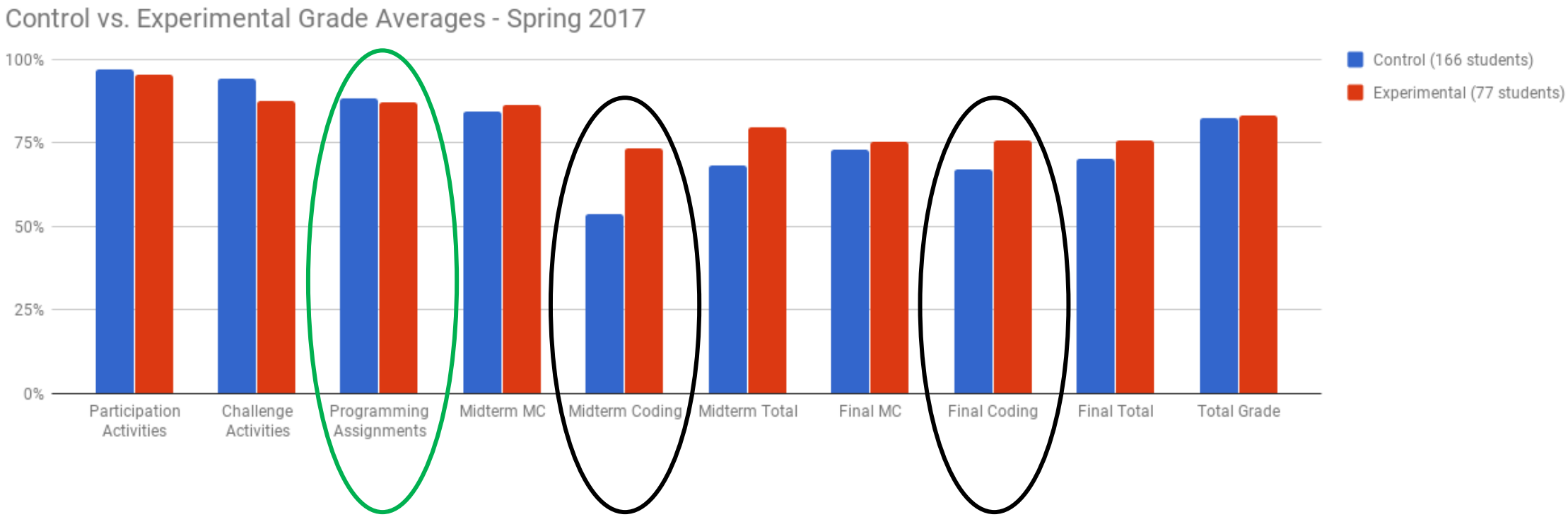
- Participation, Challenge, and Programming Activities, Midterm, Final, Total grade
- Bonferroni correction

Results – Experimental group indicated better student satisfaction

Question	Control group average	Experimental group average	p-value
I enjoy the class	4.53	✓ 4.87	0.046*

*Note these questions are not presented in the same order they were given to the students

Results – Experimental group did not perform worse – in fact, did better



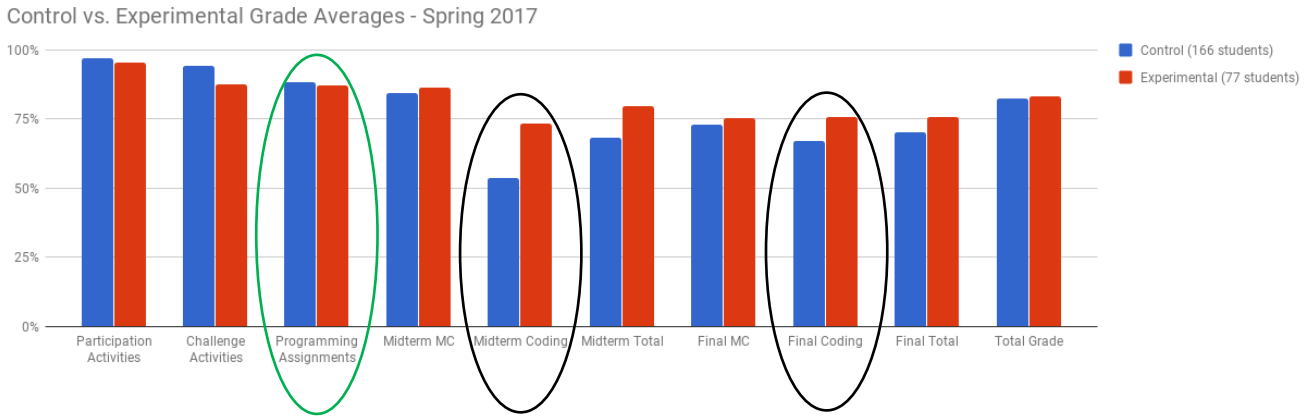
Conclusion – MSPs improved the students' experience

Students are happier

Student performance did not worsen

- In fact performed better

Question	Control group average		Experimental group average	p-value
I enjoy the class	4.53	✓	4.87	0.046*
I was prepared for the midterm exam	3.63	✓	4.18	0.004*
The weekly programming assignments were enjoyable	3.37	✓	4.13	0.001**
I learned a lot from the weekly programming assignments	4.58	✓	4.94	0.029*
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 feel anxious about the final exam	4.89	✓	4.37	0.020*



Additional work: One-year results, CS2 performance, MSP variations

This work was supported by the U.S. Dept. of Education (GAANN fellowship) and by Google

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