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
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)
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

```c++
#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

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

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

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%

One large program
Many small programs

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

Copyright © 2018 Joe Michael Allen and Frank Vahid, UC Riverside
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

<table>
<thead>
<tr>
<th>Question</th>
<th>Control group average</th>
<th>Experimental group average</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I enjoy the class</td>
<td>4.53</td>
<td>✓ 4.87</td>
<td>0.046*</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Question</th>
<th>Control group average</th>
<th>Experimental group average</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I enjoy the class</td>
<td>4.53</td>
<td>4.87</td>
<td>0.046*</td>
</tr>
<tr>
<td>I was prepared for the midterm exam</td>
<td>3.63</td>
<td>4.18</td>
<td>0.004*</td>
</tr>
<tr>
<td>The weekly programming assignments were enjoyable</td>
<td>3.37</td>
<td>4.13</td>
<td>0.001**</td>
</tr>
<tr>
<td>I learned a lot from the weekly programming assignments</td>
<td>4.58</td>
<td>4.94</td>
<td>0.029*</td>
</tr>
<tr>
<td>I am often anxious about the class</td>
<td>3.72</td>
<td>3.15</td>
<td>0.020*</td>
</tr>
<tr>
<td>I spend a lot of time in the class figuring out system issues rather than learning programming</td>
<td>2.99</td>
<td>2.43</td>
<td>0.022*</td>
</tr>
<tr>
<td>The number of tools and websites for this class are somewhat overwhelming</td>
<td>3.15</td>
<td>2.50</td>
<td>0.010*</td>
</tr>
<tr>
<td>I feel anxious about the final exam</td>
<td>4.89</td>
<td>4.37</td>
<td>0.020*</td>
</tr>
</tbody>
</table>

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