A Many Small Programs (MSP) Approach in a CS1 Course

Joe Michael Allen

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

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

Our Solution

› Improve students’ experience
  › Improve student satisfaction & happiness
  › Without worsening performance

› Focus on weekly programming assignments
  › Large part of the students’ experience
  › Key source of issues – student struggle/fear

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

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
Outline

- Related work
- What is an MSP teaching approach
- Study 1 – Satisfaction & grade performance
- Study 2 – MSP usage analysis - UCR
- Study 3 – MSP usage analysis - Other universities
- Study 4 – MSP pivot analysis
- Current/Future work
- Conclusion
Related work

- Improving CS1
  - Pair programming [1]
  - Use of different programming languages [2]
  - Various tutoring models [3]
  - Student self agency in assignments [4]
  - Plagiarism detection [5]

- Focus on programming assignments
  - Real-world programming assignments (web-crawler, analyze DNA, predict election outcomes) [6, 7]
  - Gamify programming assignments (missions, leader boards, badges) [8, 9]

- No recent work published on our proposed solution
What is an MSP Teaching Approach?

- Traditional: One Large Program (OLP) each week
  - Solution 50-200 lines
  - Long spec
  - Multiple topics
Many Small Programs (MSPs) each week

- MSP teaching 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

OLP (100%)

MSP (100%)

<table>
<thead>
<tr>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.10 LAB: Count input length without spaces, periods, or commas

Given a line of text as input, output the number of characters excluding spaces, periods, or commas.

Ex: If the input is "Listen, Mr. Jones, calm down." (excluding the quotes), the output is:

21

Note: Account for all characters that aren't spaces, periods, or commas (Ex: "r", "2", ").
MSPs – solution

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

int main() {
    string userText;
    unsigned int i;
    int charCount;

    getline(cin, userText); // Gets entire line, including spaces.

    charCount = 0;
    for (i = 0; i < userText.length(); ++i) {
        if ((userText.at(i) == ' ') && (userText.at(i) != ',')) {
            charCount += 1;
        }
    }

    cout << charCount << endl;
    return 0;
}```
MSPs – test cases

1. Compare output (3 points)
   When input is:
   
   Listen, Mr. Jones, calm down.

   Standard output exactly matches
   
   21

2. Compare output (3 points)
   When input is:
   
   Howdy!

   Standard output exactly matches
   
   6

3. Compare output (2 points)
   When input is:
   
   abcd,,,efgh,...ijkl ....

   Standard output exactly matches
   
   12

4. Compare output (2 points)
   When input is:
   
   ,

   Standard output exactly matches
   
   0

Test cases:

- 10 points per MSP
- Input/output tests
- Unit tests
MSP sample 1 – week 1

1.19 LAB: Input: Mad Lib

Mad Libs are activities that have a person provide various words, which are then used to complete a short story in unexpected (and hopefully funny) ways.

Complete the program to read the needed values from input, that the existing output statement(s) can use to output a short story.

Ex: If the input is “Eric Chipotle 12 cars” (excluding the quotes), the output is:

Eric went to Chipotle to buy 12 different types of cars.

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

int main() {
    string firstName;
    string genericLocation;
    int wholeNumber;
    string pluralNoun;

    cin >> firstName;
    cin >> genericLocation;
    cin >> wholeNumber;
    cin >> pluralNoun;

    cout << firstName << " went to " << genericLocation << " to buy " << wholeNumber << " different types of " << pluralNoun << "." << endl;
    return 0;
}
```
6.8 LAB: Step counter

A pedometer treats walking 2,000 steps as walking 1 mile. Write a program whose input is the number of steps, and whose output is the miles walked.

Ex: If the input is 5345, the output is:

2.6725

Your program must define and call a function:

double StepsToMiles(int userSteps)

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;

double StepsToMiles(int userSteps) {
    return userSteps / 2000.0;
}

int main() {
    int inputSteps;
    cin >> inputSteps;
    cout << StepsToMiles(inputSteps) << endl;
    return 0;
}
```
8.15 LAB: Word frequencies

Write a program that reads a list of words. Then, the program outputs those words and their frequencies. The input begins with an integer indicating the number of words that follow. Assume that the list will always contain less than 20 words.

Ex: If the input is:

```
5 hey hi Mark hi mark
```

Then, the output is:

```
hey 1
hi 2
Mark 1
hi 2
mark 1
```

Hint: Use two vectors, one vector for the strings and one vector for the frequencies.

```cpp
#include <list>
#include <vector>
#include <string>
using namespace std;

int main() {
    vector<string> userWords;
    vector<int> wordFreq;
    string currWord;
    unsigned int i;
    unsigned int j;
    unsigned int numWords;

    // Integer indicating the number of integers that follow
    cin >> numWords;

    // Gets list of words from input and initialize word’s frequency to 0
    for (i = 0; i < numWords; ++i) {
        cin >> currWord;
        userWords.push_back(currWord);
        wordFreq.push_back(0);
    }

    // Determine frequency of each word
    for (i = 0; i < userWords.size(); ++i) {
        for (j = 0; j < userWords.size(); ++j) {
            if (userWords[i] == userWords[j]) {
                wordFreq[i] = wordFreq[i] + 1;
            }
        }
    }

    // Outputs words and their frequencies
    for (i = 0; i < userWords.size(); ++i) {
        cout << userWords[i] << ' ' << wordFreq[i] << endl;
    }
    return 0;
}
```
Study 1 – Satisfaction & grade performance

RQ’s:
- Do MSPs impact student satisfaction?
- Do MSPs impact student grade performance?

Methods
- Student “stress” survey
  - Given week 8 of the quarter
  - Ask students about their experience
  - 18 questions: Strongly agree (6) to Strongly disagree (0)
- Student grade performance
  - Participation, Challenge, and Programming Activities, Midterm, Final, Total grade

Study 1 - Conclusion

- MSP group had more favorable responses and enjoyed the class more
- MSP group student grade performance did not worsen (higher coding scores on exams)
- Results: UCR CS1 use MSPs, ~200 universities use MSPs, and zyBooks mimicked and maintains MSPs

Student satisfaction – stress survey results

- Higher is better
- Lower is better

Student grade performance results

- Higher is better
- Lower is better

Copyright © 2020 Joe Michael Allen and Frank Vahid, UC Riverside
Study 2 – MSP usage analysis - UCR

RQ’s:
- How do students interact with MSPs?

Methods
- UCR CS1 Spring 2017 MSP section: 76 students
- zyLab metadata

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

A: With policy adjustment in Fall 2018, started 5.3 days before
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 full-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 help themselves when stuck?
A: Each week, 50% of students pivoted (avg. 1.3 pivots)
Q: Can we visually represent pivots?
A: Yes - Student workflow charts (GANTT charts)

Q: Can we visually represent pivots?
A: Yes - Student workflow charts (GANTT charts)
Q: Can we visually represent pivots?
A: Yes - Student workflow charts (GANTT charts)
Student workflow chart

Week 2 Workflow (139min 24sec; 53d; 72s)

Lab #

Time (min)

Copyright © 2020 Joe Michael Allen and Frank Vahid, UC Riverside
Common question – do students skip the ‘hard parts’?

A: ~95% of students score full credit using their top 5 highest scores
**Q: Do students use MSPs to study for exams?**

A. Yes, students use MSPs to study for exams

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total number of students</strong></td>
<td>76</td>
</tr>
<tr>
<td><strong>Total number of MSPs</strong></td>
<td>61</td>
</tr>
<tr>
<td><strong>% of students that used MSPs to study for the midterm</strong></td>
<td>38%</td>
</tr>
<tr>
<td><strong>% of students that used MSPs to study for the final</strong></td>
<td>37%</td>
</tr>
<tr>
<td><strong>% of students that used MSPs to study for either exam</strong></td>
<td>54%</td>
</tr>
<tr>
<td><strong>% of MSPs that were used to study for the midterm</strong></td>
<td>97%</td>
</tr>
<tr>
<td><strong>% of MSPs that were used to study for the final</strong></td>
<td>90%</td>
</tr>
<tr>
<td><strong>% of MSPs that were used to study for either exam</strong></td>
<td>98%</td>
</tr>
</tbody>
</table>
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
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

› And, MSP CS1 students do just as well as OLP CS1 students in an OLP CS2.
Study 3 – MSP usage analysis - Other universities

» RQ’s:
  » How do students interact with MSPs at other universities?

» Methods

<table>
<thead>
<tr>
<th>Prog Language</th>
<th>#Students</th>
<th># MSPs</th>
<th># Submissions collected</th>
<th># Develops collected</th>
</tr>
</thead>
<tbody>
<tr>
<td>University 1</td>
<td>C++</td>
<td>20</td>
<td>98</td>
<td>3177</td>
</tr>
<tr>
<td>University 2</td>
<td>Python</td>
<td>81</td>
<td>69</td>
<td>19244</td>
</tr>
<tr>
<td>University 3</td>
<td>C++</td>
<td>30</td>
<td>19</td>
<td>2397</td>
</tr>
<tr>
<td>University 4</td>
<td>C++</td>
<td>14</td>
<td>61</td>
<td>1675</td>
</tr>
<tr>
<td>University 5</td>
<td>Java</td>
<td>11</td>
<td>51</td>
<td>643</td>
</tr>
<tr>
<td>University 6</td>
<td>C++</td>
<td>234</td>
<td>77</td>
<td>21451</td>
</tr>
<tr>
<td>University 7</td>
<td>Python</td>
<td>333</td>
<td>43</td>
<td>88981</td>
</tr>
<tr>
<td>University 8</td>
<td>C++</td>
<td>79</td>
<td>25</td>
<td>7315</td>
</tr>
<tr>
<td>University 9</td>
<td>Java</td>
<td>56</td>
<td>59</td>
<td>7454</td>
</tr>
<tr>
<td>University 10</td>
<td>Java</td>
<td>321</td>
<td>65</td>
<td>40320</td>
</tr>
</tbody>
</table>

Study 3 - Conclusion

- Similar results from other universities
  - Spend sufficient time (avg 12min per lab)
  - Start early (avg 2.2 days)
  - Complete most MSPs (avg 91% completion)
Study 4 – MSP pivot analysis

» RQ’s:
  » Do students make use of pivoting with MSPs?

» Pivot definition:
  » When a student switches to a different activity before completing the previous one first
Pivot outcomes

% Students that pivot each week

# Pivots each week (avg, stdev)
Pivot outcomes

Week 4 Workflow (220min 55sec; 128d; 11s)

Week 8 Workflow (202min 6sec; 126d; 78s)
Study 4 – Conclusion

- Students make use of pivoting
  - 65% of students pivot each week
  - Avg 2.2 pivots each week

- Student workflow charts help us visually recognize pivot patterns
Current/Future work

- Does showing student their workflow charts impact their behavior
- MSP student workflow chart classifications
- Make all tools and analyses available online
Does showing student their workflow charts impact their behavior

- Fall 2020 – CS10A
- Are there any impacts in:
  - Timespent
  - Number of develops / submits
  - Cheating

- Interactive website
MSP student workflow chart classifications

Week 7 Workflow (119min 25sec; 120d; 19s)

Features
- Start: early
- End: late
- Work type: spread
- Num subs: high
- Timespent: avg
- Cheating: false

Copyright © 2020 Joe Michael Allen and Frank Vahid, UC Riverside
Make all the tools and analysis available online

UCR - zyBooks zyLab Analysis

Lorem ipsum.

Questions? Email: analysis-support@cs.ucr.edu

- Upload zyBooks zyLab log: [Choose File] No file chosen
- sample_1.csv (View file) (Download file)
- sample_2.csv (View file) (Download file)

Analysis options

- Run all analyses
- Run time spent analysis
- Run score analysis

ANALYZE
Conclusion

- Using an MSP approach in CS1 has positive results
  - Students are more satisfied & grades are not harmed

- Students make good use of MSPs
  - Sufficient time, start early, complete more than required, pivot when stuck, study for exams
  - MSP CS1 students do just as well as OLP CS1 students in an OLP CS2

- Other universities show similar results

- Students pivot when stuck
  - Workflow charts provide deep insight into student engagement
References and Publications

References

- J. Bennedsen and M. E. Caspersen: Failure rates in introductory programming: 12 years later. ACM Inroads 10, 2 (June 2019), 30–36
References and Publications

Included Publications


Other Publications


Other Talks

Appendix A: Student workflow chart

Week 2 Workflow (139min 24sec; 53d; 72s)
Appendix B: Online tools

UCR - zyBooks zyLab Analysis

Lorem ipsum.
Questions? Email: analysis-support@cs.ucr.edu

Interactive website

Analysis options

- Run all analyses
- Run time spent analysis
- Run score analysis

Upload zyBooks zyLab log: Choose File No file chosen
- sample_1.csv (View file) (Download file)
- sample_2.csv (View file) (Download file)