An Analysis of Using Coral Many Small Programs in CS1

Joe Michael Allen

Joe Michael Allen¹, Frank Vahid¹,²
¹Computer Science and Engineering, University of California, Riverside
²zyBooks
jalle010@ucr.edu, vahid@cs.ucr.edu
CS1 is difficult

- Problems in CS1
  - High student stress
  - Cheating
  - High DFW rates

- Reason
  - Technical challenges & nuances of learning commercial languages like C++, Python, and Java

- Goal
  - Reduce student stress & improve student experience
  - Our solution: hybrid Coral/C++ MSP teaching approach

```cpp
#include <iostream>

using namespace std;

int main() {
    cout << "Hello World!";
    return 0;
}
```

C++ source code for an introductory output program
Background – MSP teaching approach

- MSP - Many small programs
  - 5-7 programs per week

- Characteristics
  - Concise prompt
  - 20-50 lines of code (solution)
  - One topic per lab

- Benefits\(^{[1,2]}\)
  - Earlier starts
  - Reduced stress
  - Additional practice

---

Background – Coral

› Coral: Ultra-simple code & flowchart language for learning programming
  › Web-based & Fully executable
  › Designed for college students
  › Pseudocode to resemble commercial languages

› Features
  › Supports only 7 instructions
  › Exactly one statement per line
  › Only integer and float data types
  › Requires no main() or include/use directives

Background – Coral

Coral’s online web-based visual simulator

Coral’s visual flowchart simulator
Prompt: Write a program whose inputs are three integers, and whose output is the largest of the three values. Ex: If the input is 7 15 3, the output is: 15

**Example: C++ vs. Coral MSP Solution**

```cpp
int num1;
int num2;
int num3;

cin >> num1;
cin >> num2;
cin >> num3;

if (num1 >= num2 && num1 >= num3) {
    cout << num1;
}
elseif (num2 >= num3) {
    cout << num2;
}
else {
    cout << num3;
}
```

**Coral**

```
integer num1
integer num2
integer num3

num1 = Get next input
num2 = Get next input
num3 = Get next input

if (num1 >= num2) and (num1 >= num3)
    Put num1 to output
elseif (num2 >= num3)
    Put num2 to output
else
    Put num3 to output
```
Methods

Course details
- Spring 2020 CS1 course (300-500 students)
- 50/50 major and non-major students
- zyBooks interactive textbook
- C++ -- input/output, variables, expressions, branches, loops, functions, and vectors.

Data collection
- Gradebook
- Surveys
- Activity log file

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>lab_id</td>
<td>user_id</td>
<td>timestamp</td>
<td>submission</td>
<td>zip_location</td>
<td>score</td>
<td>max_score</td>
</tr>
<tr>
<td>2 LAB: Sample 1</td>
<td>1103</td>
<td>1/1/2021 17:23</td>
<td>1</td>
<td><a href="https://xyz.zip">https://xyz.zip</a></td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>3 LAB: Sample 1</td>
<td>1103</td>
<td>1/1/2021 17:32</td>
<td>0</td>
<td><a href="https://xyz.zip">https://xyz.zip</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 LAB: Sample 2</td>
<td>1103</td>
<td>1/1/2021 18:11</td>
<td>0</td>
<td><a href="https://xyz.zip">https://xyz.zip</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 LAB: Sample 3</td>
<td>1103</td>
<td>1/2/2021 12:00</td>
<td>1</td>
<td><a href="https://xyz.zip">https://xyz.zip</a></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>6 LAB: Sample 7</td>
<td>1103</td>
<td>1/2/2021 12:09</td>
<td>1</td>
<td><a href="https://xyz.zip">https://xyz.zip</a></td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>
Experiment Details

CS1 course at UCR during Spring 2020; 10 week quarter

Pure C++
- Same online textbook
- Same topics
- Same final exam
- 1 week for assignments / different release dates
- 2 sections; 196 students
- Instructor 1
- 10 weeks C++
- C++ Midterm

Hybrid Coral/C++
- 1 section; 98 students
- Instructor 2
- 3 weeks Coral / 7 weeks C++
- Coral Midterm
Results: Grade performance (lab activities)

- Pure C++ group: 96% avg.
- Hybrid Coral/C++ group: 93% avg.
# Results: Grade performance (class)

Table 1: Student grade performance on all categories of our CS1 class

<table>
<thead>
<tr>
<th>Class category</th>
<th>Pure C++</th>
<th>Hybrid Coral/C++</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total class grade</td>
<td>88%</td>
<td>95%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Final exam</td>
<td>83%</td>
<td>88%</td>
<td>0.043</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>83%</td>
<td>95%</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Participation activities</td>
<td>94%</td>
<td>95%</td>
<td>0.482</td>
</tr>
<tr>
<td>Challenge activities</td>
<td>94%</td>
<td>95%</td>
<td>0.616</td>
</tr>
<tr>
<td>Lab activities</td>
<td>96%</td>
<td>93%</td>
<td>0.134</td>
</tr>
</tbody>
</table>

*Spring 2020 was start of pandemic
Results: Time spent

- Pure C++ group: 81 minutes avg.
- Hybrid Coral/C++ group: 91 minutes avg.

Copyright © 2021 Joe Michael Allen and Frank Vahid, UC Riverside
Results: Activity runs (develops & submits)

- Pure C++ group: 72 runs | 48 devs | 24 subs avg.
- Hybrid Coral/C++ group: 83 runs | 67 devs | 16 subs avg.
Results: Start date

- Pure C++ group: 4.5 days / 4.8 days adj.
- Hybrid Coral/C++ group: 4.6 days / 3.9 days adj.

![Graph showing days before due by week with statistical significances.]

\[ p = \begin{array}{ccccccc}
< 0.001 & < 0.001 & 0.854 & 0.961 & 0.052 & < 0.001 & < 0.001 \\
0.039 & 0.003 & & & & & \\
\end{array} \]

*No weeks 1 or 2 (grace period)*
Results: Pivots

- Pure C++ group: 1.3 pivots / 1.5 pivots adj.
- Hybrid Coral/C++ group: 2.4 pivots / 2.2 pivots adj.

p = 0.022 0.076 < 0.001 0.034 0.002 < 0.001 < 0.001 0.058 0.004

*No week 6 (midterm)*
Results: Student surveys

I enjoyed learning Coral
91 responses

- Strongly disagree: 36.3%
- Disagree: 40.7%
- Neutral: 12.1%
- Agree: 2.6%
- Strongly agree: 9.9%

I think learning Coral first made C++ easier to learn
91 responses

- Strongly disagree: 11%
- Disagree: 45.1%
- Neutral: 18.4%
- Agree: 17.6%
- Strongly agree: 7.7%

I found the transition from Coral to C++ difficult
91 responses

- Strongly disagree: 26.4%
- Disagree: 36.3%
- Neutral: 9.9%
- Agree: 20.4%
- Strongly agree: 20.4%

I would have preferred to learn C++ from the beginning of the course, without Coral
91 responses

- Strongly disagree: 38.5%
- Disagree: 17.6%
- Neutral: 13.2%
- Agree: 23.1%
- Strongly agree: 7.7%
Results: Student comments

- [Coral] made the transition extremely easy and I think learning C++ first would have confused me. Coral was good for visual learning (which applies to me) and it help me a lot.

- Coral was easier to learn initially than I think C++ would have been because its easier to identify my thought process of what I want my code to do in Coral than in C++

- Its nice in concept but a tad overwhelming only on the syntax side since its much more specific in C++

- If the goal is to teach students with absolutely no programming experience then it is good. However students with experience in engineering may find the approach dumbed down and less engaging. […]
Conclusion

- Our experience
  - Similar high grade performance
  - Coral/C++ spent more time
  - Coral/C++ more activity
  - C++ group works earlier
  - Coral/C++ pivot more

- Student comments
  - Enjoyed the approach; could be slow for folks with prior coding experience

*Not meant to conclude that one teaching approach is better, but both work*