

AT HOME WITH ENGINEERING EDUCATION



JUNE 22 - 26, 2020

Asee's Virtual Conference

#ASEEVC

#ASEEVC

**ASEE'S
VIRTUAL
CONFERENCE**



JUNE 22 - 26, 2020

At Home with
Engineering Education

Teaching Coral before C++ in a CSI Course

Joe Michael Allen¹ and Frank
Vahid^{1,2}

¹University of California, Riverside

²zyBooks

Difficulties of CSI

CSI courses are difficult and commonly have high D/F/W rates

Why?

- Technical challenges in the first several weeks, including nuances of commercial languages like Python, Java, and C++

```
print('Enter wage:', end=' ')  
wage = int(input())  
wage = wage + 10  
print('New wage:')  
print(wage)
```

Python source code for an introductory input/output program

ASEE'S VIRTUAL CONFERENCE

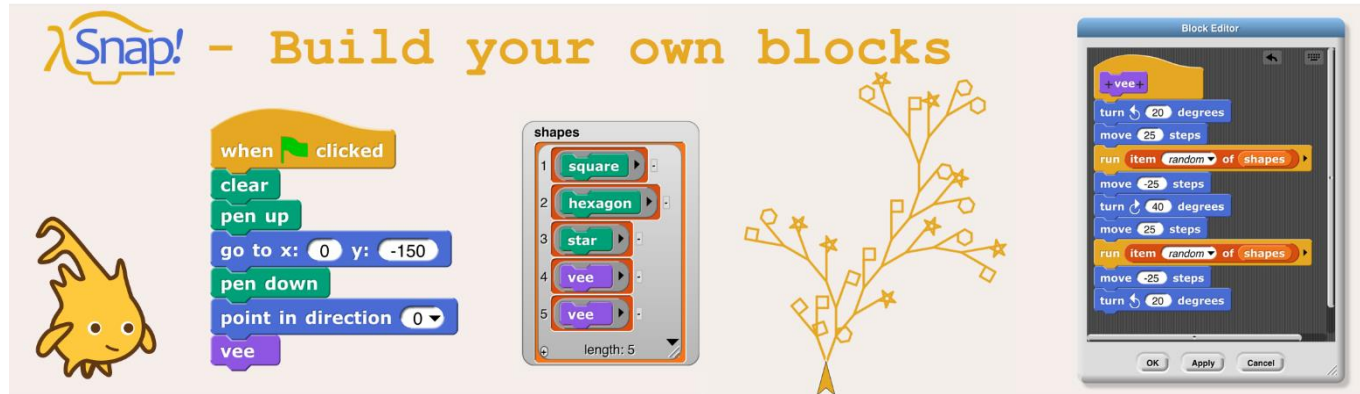


#ASEEVC



Solution 1: Visual programming languages

Use drag-and-drop visual programming languages like Alice, Scratch, and Snap



Snap! - Build your own blocks

The image displays the Snap! programming environment. On the left, a script is shown with the following blocks: "when green flag clicked", "clear", "pen up", "go to x: 0 y: -150", "pen down", "point in direction 0", and "vee". Below the script is a yellow cartoon character. In the center, a "shapes" palette lists: 1 square, 2 hexagon, 3 star, 4 vee, 5 vee, with a "length: 5" indicator. To the right, a tree-like drawing is composed of various shapes. On the far right, a "Block Editor" window shows a script with blocks: "vee", "turn 20 degrees", "move 25 steps", "run item random of shapes", "move -25 steps", "turn 40 degrees", "move 25 steps", "run item random of shapes", "move -25 steps", and "turn 20 degrees".

ASEE'S VIRTUAL CONFERENCE



Solution 2: Coral

Coral is an ultra-simple code & flowchart language for learning programming

- Created in 2017 as a joint project by the Univ. of California at Riverside, the Univ. of Arizona, and zyBooks
- Created for college students with learning & education in mind
- Designed to look like common industry languages
- Ultra-simple syntax with the language looking like pseudocode

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Python vs. Coral

```
print('Enter wage:', end=' ')
wage = int(input())
wage = wage + 10
print('New wage:')
print(wage)
```

Python source code for an introductory
input/output program

```
integer wage
wage = Get next input
wage = wage + 10
Put "New wage " to output
Put wage to output
```

Coral source code for an introductory
input/output program

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Coral's online web-based visual simulator

The screenshot shows a web-based visual simulator interface. On the left, a code editor displays the following code:

```
1 integer wage
2 wage = Get next input
3 wage = wage + 10
4 Put "New wage " to output
5 Put wage to output
6
```

The code editor has two tabs: "Code" (selected) and "Flowchart".

On the right, the "Variables" panel shows a variable named "wage" of type "integer" with a value of "35".

Below the variables, the "Input" panel shows a text box containing the value "25".

Below the input, the "Output" panel shows the text "New wage 35_" in green.

At the bottom of the interface, there are several control buttons and a dropdown menu:

- "EXIT EXECUTION" (dark blue button)
- "STEP" (orange button)
- "RUN" (orange button)
- "Execution speed" dropdown menu (set to "Instant")

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Coral's online web-based flowchart visual simulator

Start

x = Get next input

y = Get next input

x > y

TRUE max = x

FALSE max = y

Put max to output

End

Variables

0	x	integer
0	y	integer
0	max	integer

Input

55 79

Output

-

Code Flowchart

ENTER EXECUTION

STEP

RUN

Execution speed
Medium

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Coral's features

- Only 7 kinds of statements
- Allows exactly one statement per line, and requires 3-space indents for sub-statements
- Supports integer and float data types, which are sufficient to learn programming fundamentals by processing data
- Requires no surrounding `main()` or `include/use` directives

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Coral statements converted into C++ statements

Type	Coral	C++
Variable declaration	<pre>integer numCats float wallHeight integer array(5) numSequence float array(5) wallAreas</pre>	<pre>int numCats; double wallHeight; vector<int> numSequence(5); vector<double> wallAreas(5);</pre>
Input statement	numCats = Get next input	cin >> numCats;
Output statement	Put numCats to output	cout << numCats;
Assignment statement	numCats = numCats + 1	numCats = numCats + 1;
Branch statements	<pre>if numCats > 5 Put "too many" to output elseif numCats > 1 Put "sufficient" to output else Put "need more" to output</pre>	<pre>if (numCats > 5) { cout << "too many"; } else if (numCats > 1) { cout << "sufficient"; } else { cout << "need more"; }</pre>
While loop statement	<pre>while wallHeight <= 0 wallHeight = Get next input</pre>	<pre>while (wallHeight <= 0) { cin >> wallHeight; }</pre>
For loop statement	<pre>for i = 0; i < 5; i = i + 1 Put i to output</pre>	<pre>for (i = 0; i < 5; i = i + 1) { cout << i; }</pre>
And/or/not	(a < b and a < c) or (not(a > d))	(a < b && a < c) (!(a > d))

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Research question

“Can a CSI course start by using Coral, then switch to a commercial language, and still have students attain the same proficiency in the commercial language?”

- Coral’s original intent was for CS0
- But, could using Coral in CSI have immediate benefits?

ASEE’S VIRTUAL CONFERENCE



#ASEEVC



Methodology: Course information



Course

- Serves 300-500 students per 10-week quarter (80-100 students per section)
- Half computing/engineering majors and half non-majors
- Course topics: input/output, variables, branches, loops, functions, and vectors, all in C++
- zyBooks: (1) online reading with auto-graded short question activities, (2) online small auto-graded coding homework activities, and (3) 5-7 small/medium programming assignment activities

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Methodology: Data collection

Gathered data from 3 primary sources

1. Student grades
2. Anonymous weekly surveys (“stress surveys”)
3. University's anonymous end-of-quarter course/instructor evaluations

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Methodology: Experiment details

CSI course at UCR during Fall 2019; 10 week quarter

Coral-to-C++

1 section; 80 students

Instructor 1

5 weeks Coral; 5 weeks C++

Same online textbook

Same topics

Same final exam

C++-only

5 sections; 434 students

Instructor 2

10 weeks C++

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Methodology: Demographics

Coral-to-C++

- 16% (13) of students were computing majors (Computer Science, Computer Engineering, or Computer Science with Business Applications)
- 84% (67) were non-major students (Biology, Math, Physics, etc.)
- Most of the students (60%), were freshmen
- “Do you have prior programming experience? (Don't worry; none is expected)”
 - Responses range from 0 (No experience) to 5 (Lots of experience): Avg response was 0.84

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Results: Grade performance

Both groups performed about the same on the final exam

	Coral-to-C++ average	C++-only average	p-value
Midterm	82.35	79.32	0.0762
Midterm MC	84.73	82.46	0.0944
Midterm Code	79.97	76.19	0.0921
Final	77.60	79.72	0.1753
Final MC	80.33	82.57	0.1081
Final Code	74.86	76.86	0.2476
Total Class	82.89	85.65	0.0782

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Results: Stress survey

Week 8 Stress survey

- On most questions, both groups' responses were about the same
- Coral-to-C++: "I enjoy the class"
- C++-only: "The weekly zyLab programming assignments were stressful" & "The weekly zyLab programming assignments were frustrating"

Question	Coral/C++	C++-only	p-value
I enjoy the class.	4.95	4.54	0.0620
This class is an appropriate amount of work per week for the number of units.	4.06	4.41	0.0571
I was prepared for the midterm exam.	4.25	4.04	0.2840
I feel prepared for the final exam.	2.80	3.06	0.1767
The weekly zyLab programming assignments were enjoyable.	3.71	3.91	0.1744
The weekly zyLab programming assignments contributed to my success in the course.	4.74	4.58	0.3004
I learned a lot from the weekly zyLab programming assignments.	4.88	4.68	0.2299
I feel confident in my ability to write a small (< 50 line) useful program.	3.91	3.95	0.3926
I spend a lot of time in the class figuring out system issues rather than learning programming.	2.00	2.38	0.0864
The number of tools and websites for this class are somewhat overwhelming.	2.23	2.29	0.3749
I felt anxious before the midterm exam.	4.22	4.25	0.3986
I feel anxious about the final exam.	4.95	4.72	0.1851
The weekly zyLab programming assignments were stressful.	3.82	3.11	0.0089*
The weekly zyLab programming assignments were frustrating.	3.98	3.31	0.0114*
I am often anxious about the class.	3.17	3.05	0.3855



Results: Feedback survey

End-of-course Coral-to-C++ anonymous survey

- 64 students (80% of students) completed the survey
- Questions above the black line are those where higher scores are better, while below the line are those where lower scores are better
- Overall, the students had positive responses related to the Coral-to-C++ approach

Question	Answer (0-4)
I enjoyed learning Coral	3.11
I understood Coral compiler errors	3.38
I was able to capture my programs in Coral without too much difficulty	3.11
I was able to successfully debug my Coral programs	3.28
I found the Coral simulator helpful in learning concepts	3.23
I found the Coral simulator helpful when debugging my programs	3.19
I enjoyed learning C++	3.03
I understood C++ compiler errors	2.28
I was able to capture my programs in C++ without too much difficulty	2.22
I was able to successfully debug my C++ programs	2.34
I think learning programming concepts (like loops or functions) was easier in Coral than it would have been in C++	2.86
I think learning Coral first made C++ easier to learn	2.75
I think my C++ code is neater/cleaner because I learned Coral first	2.52
I struggled with Coral's syntax	1.31
I struggled with C++'s syntax	2.41
I would have preferred to learn C++ through the whole course, without Coral	1.97
I think learning Coral first caused me to struggle more with C++ syntax than I would have otherwise	1.55

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Results: Student comments

Overall positive:

- “Coral provided a good foundation for students to build on. Coral taught that structure and readability is important. Coral has an amazing simulator that can step through the code like a debugger. Very good for beginners.”
- "made it a lot easier to grasp the concepts (especially the FLOWCHARTS) so by the time i got to C++ it was easy to get the hang of“

Even more experienced programmers recognized the benefits:

- "Since I already had experience programming before it didn't make too much of a difference to learn Coral before C++ but I could see why it would be easier for people who had no coding experience to learn Coral before C++."

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Discussion

- Coral was very enjoyable to teach
 - Simulators are very helpful
- Improvements
 - Switch from Coral to C++ sooner
 - Don't assign too much work

ASEE'S VIRTUAL CONFERENCE



#ASEEVC



Conclusion

- Does a Coral-to-C++ approach lead to equally-proficient learning of C++? Yes
- Does the Coral-to-C++ transition stress students more? No
- Do students enjoy learning Coral before C++? Yes

Coral is free to use at <https://coralLanguage.org/>

Email: jalle010@ucr.edu

ASEE'S VIRTUAL CONFERENCE



#ASEEVC

