

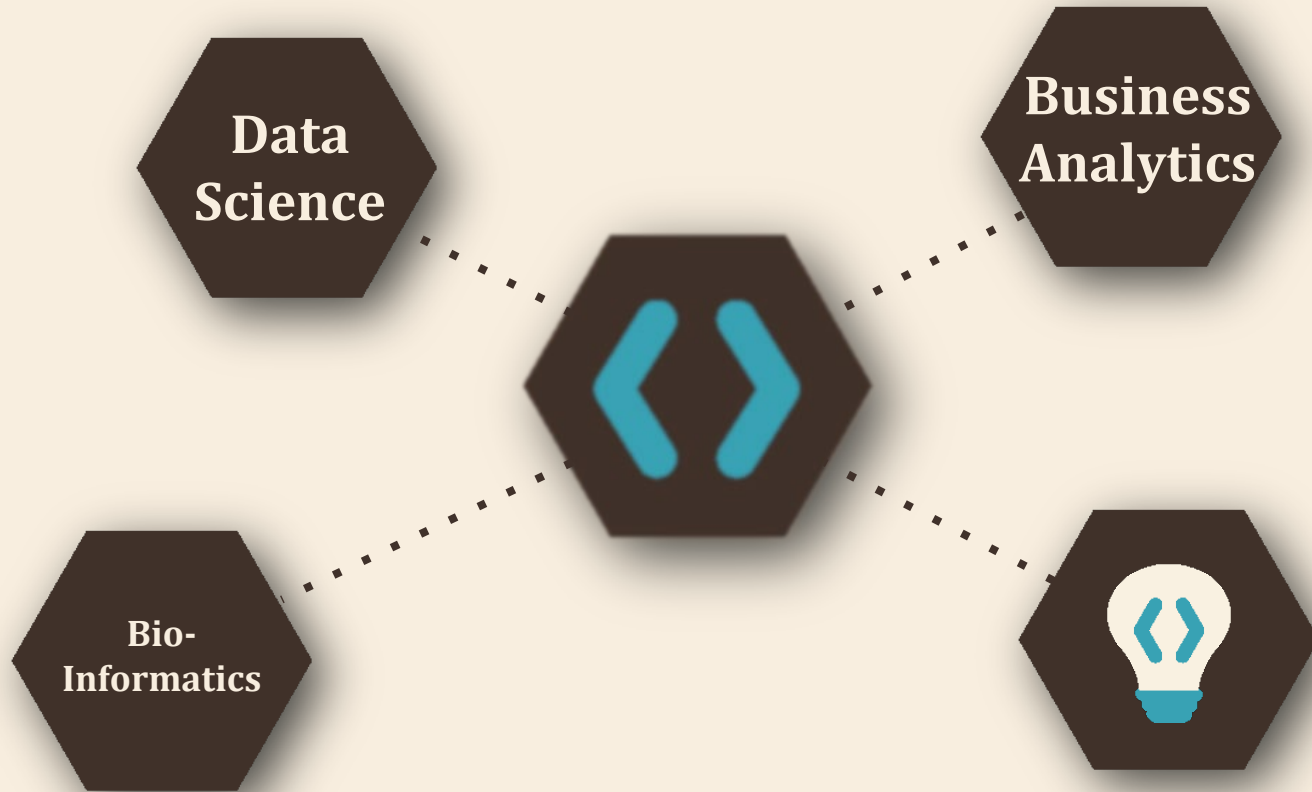
Making Code Relevant to Students' Lives

Kevin Buffardi & Richert Wang

Chico State

UCSB





**Data
Science**

**Business
Analytics**

**Bio-
Informatics**



Coding courses do not reflect California's **diversity**

Advanced Placement Computer Science A course:

- 31% **Girls** (of 49% overall enrollment)
- 29% **Latino/a/é** (of 54%)
- 2% **Black** (of 5%)

via [CSforCA](#)



Coding course content reflects conventional applications

Common assignments

- Games
- Puzzles
- Robotics
- Microcontrollers
- Media Computation

But what are students **interested** in?



Surveyed introductory coding students...

What are your hobbies and interests?

Of those hobbies and interests, which do you believe can be relevant to Computer Science or programming?



Surveyed introductory coding students...

- **What are your hobbies and interests?**
- **Of those hobbies and interests, which do you believe can be relevant to Computer Science or programming?**



Two semesters (n=198):

- Gender

Majority: 154 Male (78%),

Minority: 41 Female (21%), 1 Non-binary (<1%),

Excluded: 1 Prefer not to say (<1%), 1 no response (<1%)

- Race

Majority: 119 White and/or Asian (60%),

Minority: 77 (39%) One or more identified:

**American Indian or Alaska Native, Black or African American,
Hispanic or Latin American, Native Hawaiian and Pacific Islander,
Other**

Excluded: 2 no response (1%)



Interests using *Grounded Theory* thematic analysis:

- **Games** - video games, tabletop games, puzzles, etc
- **Reading** - reading any sort of literature
- **Arts and Crafts** - drawing, knitting, etc.
- **Music, Film, and audio/video** - consuming, creating, or editing music, movies, etc.
- **Athletics** - sports and fitness activities, either watching or participating
- **Science, Technology, Engineering, and Math (STEM)** - specific fields (e.g. Biology, Statistics, etc), building computers, and electronics
- **Other academic disciplines (non-STEM)** - any fields not traditionally identified as STEM
- **Socializing** - time or activities with friends and family
- **Sleep and relaxation** - all forms of resting
- **Animals** - interacting or taking care of pets, livestock, or other animals
- **Culinary** - cooking, eating, and drinking
- **Fashion and cosmetics** - makeup, clothes, accessories; studying, shopping, or wearing
- **Nature and outdoors** - camping, fishing, and other outdoor activities that concentrate on nature
- **Travel** - vacations, road trips, domestic and international exploration
- **Automotive** - driving, fixing, and customizing any automobiles



Interests, comparing minoritized students to majority

Games

Gender majority (n=104, **68%**) vs **minority** (n=15, **36%**), $\chi^2=5.50$, $df=1$, **$p<.05^*$**

Racial majority (n=76, 64%) vs minority (n=43, 56%). No sig. diff. (p=.48)

STEM

Gender majority (n=55, **36%**) vs **minority** (n=6, **14%**), $\chi^2=4.87$, $df=1$, **$p<.05^*$**

Racial majority (n=42, 35%) vs minority (n=19, 25%). No sig. diff. (p=.19)

Music/film/audio/visuals

Gender majority (n=51, 33%) vs minority (n=16, 38%). No sig. diff. (p=.62)

Racial majority (n=37, 31%) vs minority (n=30, 39%). No sig. diff. (p=.36)



Hired students with unique perspectives to offer

Instructions:

Record videos that demonstrate C++ concepts in ways that are relevant to *your* life



Jessica: Manage soccer player info with variables

The screenshot displays a video player interface. The main content area is divided into three sections:

- Files:** A sidebar on the left showing a file named `main.cpp`.
- Code Editor:** The central area shows the contents of `main.cpp`:

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     int shirtNum = 4;
8     string lastName= "martinez";
9     bool inGame= true;
10    double average= 15/10;
11    char grade= 'A';
12
13    cout<< "The player "<< lastName<< " who has the number " <<
    shirtNum<< " has an average of "<<average<< |
14
15
16
17 }
```
- Console/Shell:** The right side shows a terminal window with the text: `clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)` and a prompt `>`.

In the bottom right corner, there is a video feed of a woman with long dark hair and glasses, wearing a striped shirt. The video player's control bar at the bottom shows a progress bar, a play button, a volume icon, and a timestamp of `8:07 / 11:38`.



Juan: Detail videos with classes

The screenshot displays a video player interface. On the left, a code editor shows the following C++ code in `main.cpp`:

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4 using namespace std;
5
6 class Video{
7     private:
8         string title;
9         string desc;
10        double length;
11        int rating;
12    public:
13        void initVar(){
14            cout << "Enter title:" << endl;
15            |
16            cout << "Enter desc:" << endl;
17
18            cout << "Enter length:" << endl;
19
20            cout << "Enter rating:" << endl;
21        }
22    };
23
24
25 int main() {
26     Video video1;
27
28     return 0;
29 }
```

On the right, a terminal window shows the compilation and execution of the program:

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
> |
```

At the bottom right, a video inset shows a person (Juan) gesturing with his hands while speaking. The video player controls at the bottom indicate the video is at 3:41 / 7:23.



Destiny: Write horoscopes with switch statements

The video player shows a C++ code editor with the following code in `main.cpp`:

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main() {
7     cout<<"Enter your zodiac sign"<<endl;
8     cout<<"1 - Sagittarius\n" << "2 - Capricorn\n"
9     << "3 - Aquarius\n" << "4 - Pisces\n" << "5 - Aries\n"
10    << "6 - Taurus\n" << "7 - Gemini\n" << "8 - Cancer\n"
11    << "9 - Leo\n" << "10 - Virgo\n" << "11 - Libra\n"
12    << "12 - Scorpio\n";
13
14    int zodiac = 0;
15    cin >> zodiac;
16
17    switch(zodiac){
18        case 1: // if
19            cout<<"Sagittarius will have a fire year.\n";
20            break;
21
22        case 2: // else if
23            cout<<"Capricorn will always start the day with an earthy breakfast.\n";
24            break;
25
26        default: //else
27            cout<<"\n";
28    }
29
30    return 0;
31 }
```

The terminal window shows the output of the program, which is a list of 12 zodiac signs with their corresponding numbers. The video player interface includes a play button, a progress bar at 3:32 / 5:11, and a video thumbnail of a person in the bottom right corner.



Phinease: Plan workout routine with vectors

```
1 #include<iostream>
2 #include<vector>
3
4 using namespace std;
5
6 int main(){
7     vector<string>workouts;
8     string work = "";
9     string day;
10
11     do{
12         cout<<"Hello! Lets plan your workout schedule (to end list type 'done')\n";
13         getline(cin, work);
14         workouts.push_back(work);
15     }while(work!="done" void pop_back()
16     workouts.pop_back()
17
18
19
20
21
22
23 }
```

clang version 7.8.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)





Popular topic: “Adulthood”



Jason: Energy-saving thermostat with while loops

The screenshot displays a video player interface. The main content area is split into two panes. The left pane shows a code editor with a file named 'While Loop.cpp'. The code is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int temperature;
7     cout << "How many degrees is it inside today? ";
8     cin >> temperature;
9
10    while (temperature >= 75) {
11        cout << temperature << endl;
12        temperature--;
13    }
14    cout << "It is cool inside now, the AC is turning off. " << endl;
15 }
```

The right pane shows a terminal window with the text 'clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)' and a cursor on the next line.

At the bottom of the video player, there is a red progress bar and a control bar with the following elements from left to right: a play button, a next button, a volume icon, a time display '4:57 / 7:34', a dropdown arrow, a pause button, a full screen button, an HD icon, a signal strength icon, and a refresh icon.

In the bottom right corner of the video player, there is a small inset video of a man with dark hair wearing a blue hoodie, sitting in a white chair. Behind him is a whiteboard with some handwritten notes and a coat hanging on a rack.



Jessica: Apartment hunting with structs

The screenshot displays a video player interface. The main content area is divided into three sections:

- Files:** A sidebar on the left showing a file tree with 'main.cpp' selected.
- main.cpp:** A code editor showing the following C++ code:

```
6 string name;
7 double value;
8 int rooms;
9 bool availability;
10 }
11
12
13 int main() {
14
15     property apartments;
16     char repeat;
17     do{
18         cout<<"what is the name of the property";
19         cin>>apartments.name;
20
21         cout<<"what is the value of the property";
22         cin>>apartments.value;
23
24         cout<<"how many rooms does the property have";
25         cin>> apartments.rooms;
26
27         cout<<"is it available 1=true, 0=false";
28         cin>> apartments.availability;
29
30         cout<<"do you want to enter another property y/n";
31         cin>>repeat;
32     }while(repeat!=n)
33
34
35     return 0;
36 }
```
- Console:** A terminal window showing the output of the clang compiler:

```
clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)
>
```

In the bottom right corner, there is a small video inset of a woman with glasses, identified as Jessica, who is the presenter of the content. The video player controls at the bottom show a progress bar at 6:58 / 12:57 and various playback icons.



Andrea: Keeping recipes with pointers

Pointers (C++)

Computer Memory

0 1 2 3 4 5 6 7 8 9 10 11

0 1 2 3 4 5 6 7 8 9 10 11

0:00 / 7:52

⏪ ⏩ 🔊 ⏸ 📄 ⚙️ 📺 ⛶



Created problems on corresponding concepts

CodeWorkout

- **Hands-on** coding exercises in the web browser
- Automated **testing and feedback**
- “Workouts” of **related exercises**



X538: Factorial C++

Write a function in C++ called 'factorial' that will take a positive integer as input and returns its factorial. In math, a factorial is the product of all positive integers less than or equal to a given positive integer. If the integer is zero, the factorial is 1. For example, calling `factorial(3);` should calculate $3 \times 2 \times 1$ and return 6.

Examples:

```
factorial(0) -> 1
```

```
factorial(3) -> 6
```

Your Answer:



```
1 int factorial(unsigned int n)
2 {
3     if( n == 0 )
4         return 1;
5 }
6
```

Check my answer!

Reset

[Practice a different C++ exercise](#)

Feedback

Result	Behavior
	factorial(0) -> 1
	factorial(3) <i>Expected:<6> but was:<6357416></i>
	hidden test(s)



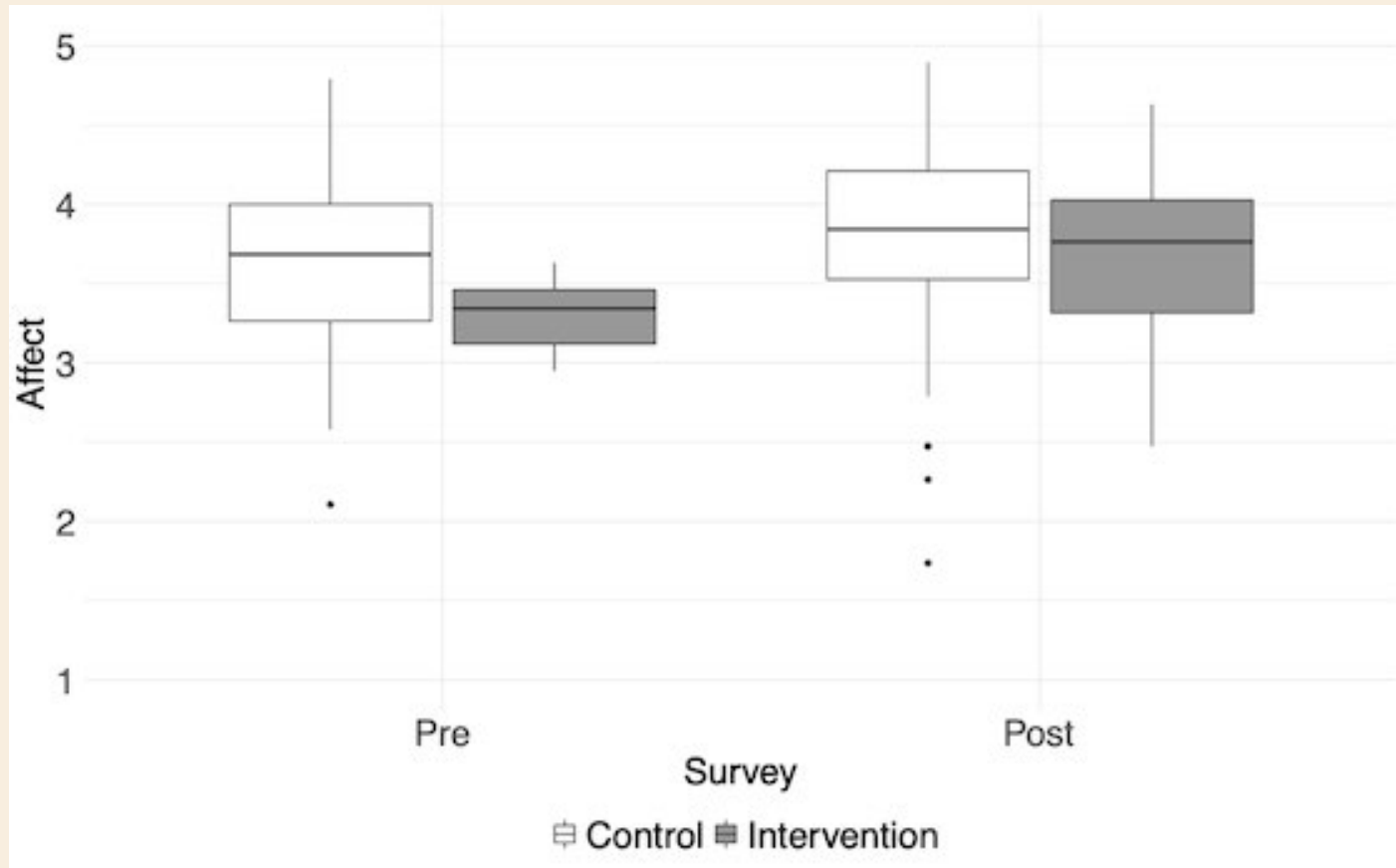
Initial adoption of videos and exercises at **Chico State**

Pre/Post surveys on **affective outcomes** (belonging, self-efficacy, etc) and **confidence in C++ skills**

- Control (n=161) from semesters with weekly readings
- Intervention (n=55) with supplementary material

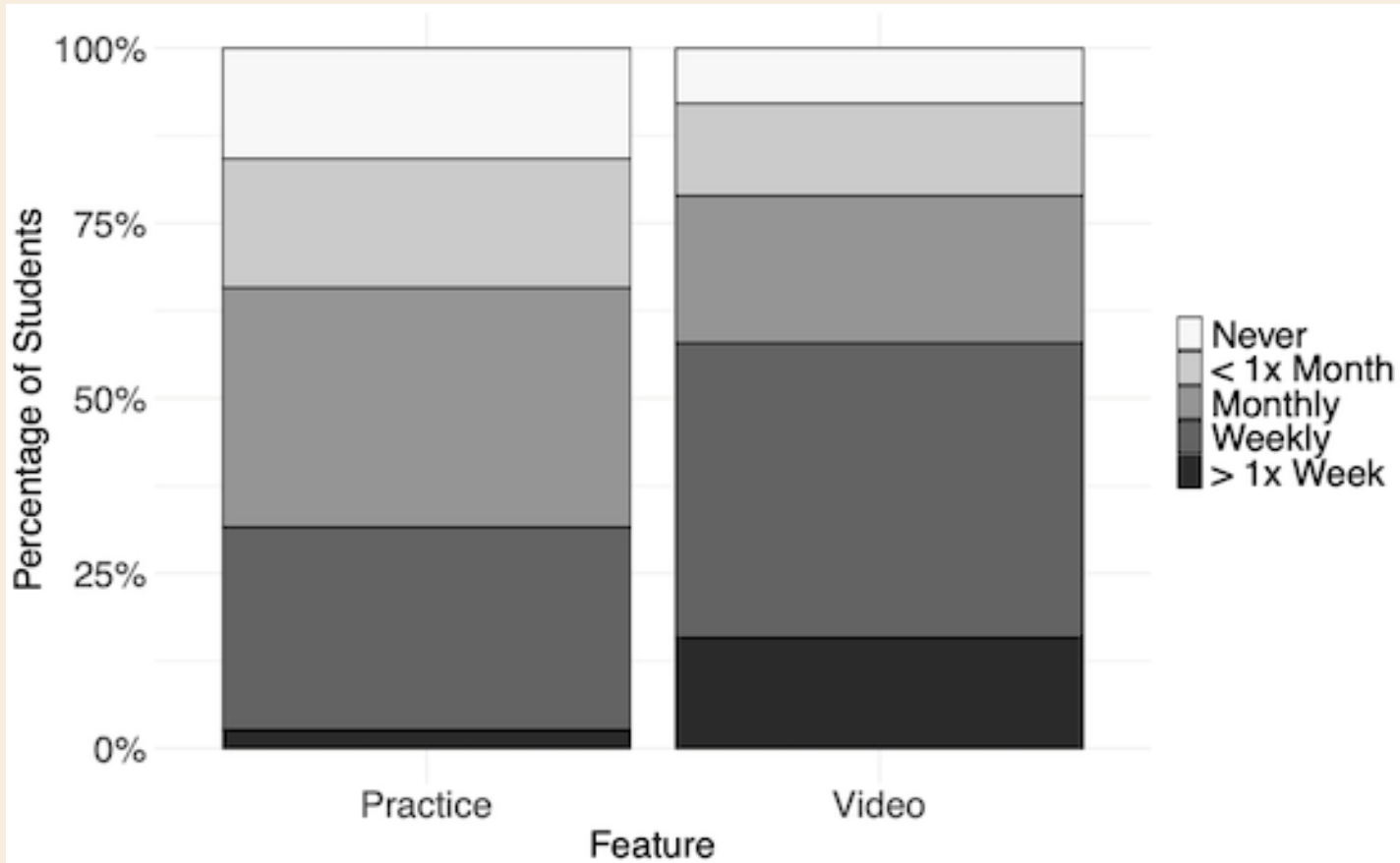


Significant greater gains in **affective outcomes** ($p < .05^*$)





Less **practice** than video watching





Developed [Codewit.us](https://codewit.us) proof-of-concept

Codewit

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     int shirtNum = 4;
8     string lastName = "martinez";
9     bool inGame = true;
10    double average = 15/10;
11    char grade = 'A';
12
13    cout << "The player " << lastName << " who has the number " <<
    shirtNum
14
15
16
17 }
```

CodeWorkout

X1182: Codewit Jessica: C++ Variables for Soccer Players

Write a program that stores the following information about a soccer player in variables with the following identifiers and corresponding values :

- shirtNum has the player's jersey number, 4
- lastName has her last name, "martinez"
- inGame says whether or not she is currently playing, or true
- average represents the points she scores per game, 1.52
- grade stores the letter grade of her performance, 'A'

The program should then display those variables' values in a message following the template:

The player lastName who has the number shirtNum has an average of average which allows her to have a grade of grade

inGame

Complete the code *within* the brackets ({ }) below.

Your Answer:

```
1 {
2
3     return 0;
4 }
5
```




Piloted proof-of-concept at **UCSB**

Codewit.us offered as optional material in addition to eTextbook

- End-of-quarter survey for formative feedback
- Observed frequency of practice *as well as* video watching
- Compared to study with separate practice, video
 - Increased likelihood of practicing *at least as often*

$\chi^2=17.037, df=2, p<.001^*$

	Practice Less	Practice Same	Practice More
Expected	58%	11%	31%
Observed	38%	27%	36%



Moving forward

- **Codewit.us** experiment as required assignments
- Additional videos, upgraded recordings
- Extended development for **Codewit.us**
 - Improved design and user experience
 - Learning Management System integration
 - More programming languages
- Scaling larger, broader