2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34705

Work in Progress: Engaging First-year Students in Programming 1 During COVID-19

Dr. Stephany Coffman-Wolph, Ohio Northern University

Dr. Stephany Coffman-Wolph is an Assistant Professor at Ohio Northern University in the Department of Electrical, Computer Engineering, and Computer Science (ECCS). Research interests include: Artificial Intelligence, Fuzzy Logic, Game Theory, Teaching Computer Science, STEM Outreach, Increasing diversity in STEM (women and first generation), and Software Engineering.

Work-in-Progress: Engaging First-Year Students in Programming 1 During COVID-19

Abstract

During the Fall 2020 semester, it became even more important than before to engage students in the “classroom” whether that be in-person, online, or a hybrid model. This paper will introduce various entrepreneurial mindset (EM) techniques to engage students that could be adapted to any engineering course. All the techniques have suggestions for adapting to a fully online course as well as working for an in-person or hybrid class. The first activity presented will be name signs with badges that will promote (1) setting, evaluating, and achieving goals, (2) self-reflection, (3) considering a problem from multiple viewpoints, and (4) seeing the values of others. Example badges include: Being Brave, Stump the Professor, Discussion Board Guru, Peer Tutor Extraordinary, and Nominator. The second activity presented will be Tik-Tok-ing the student’s way into learning concepts. This activity focuses on students’ creating course content via videos that will promote (1) being able to teach and learn from peers, (2) modifying a product based on feedback, and (3) connecting life experiences with class content. The third activity is using Play-Doh to make connections with material. In this activity students use the Play-Doh as a medium to present technical information effectively to a wide audience and make connections with life experiences and class content. Each activity will be explained with examples from introduction to programming along with methods to adapt to other engineering courses.

Introduction

Flipped classrooms, online learning, and hybrid courses are the new “normal” of how we teach. The KEEN organization and the author believe that adding an entrepreneurial mindset creates better engineers and computer scientists [1 and 2]. The KEEN framework contains the three C’s: curiosity, connections, and creating value [2] which are the key skills needed to have an entrepreneurial mindset. Ohio Northern University (ONU) uses an expanded six entrepreneurial mindset (EM) which incorporates the three C’s and three additional skills: communication, collaboration, and character [3]. This paper presents a series of short activities and course techniques to integrate these six skills into a more virtual learning experience using various hands-on activities both individually and as a group. This paper shares techniques for faculty to be more innovative as we live in the new "normal" with an increase in hybrid and online courses. Each activity is designed to not only cement a programming skill (or skills) but integrate EM skills with special attention to making connections between abstract programming concepts and the real-world examples to help ground those concepts for all students and allow students to unleash their creativity. These techniques introduce (1) gamification, (2) students taking control of their learning, and (3) encouragement of creativity which will hopefully spark their internal passion and help to create a better programmer, a better engineer, a better team member, and better person.

About the Course

The introductory programming course (ECCS 1611 Programming 1) at Ohio Northern University (ONU) consists of one-semester four-credit courses. The course consisting of three

50-minute lectures plus a 165-minute laboratory for 15 weeks. Programming 1 is offered in the fall term and focuses on using C++ to implement small programs exercising concepts in sequencing, selection, iteration, pointers, basic data structures, and an introduction to Object-Oriented Programming (OOP) design. This course, normally taken in the first year, is required for all ONU students majoring in computer science, computer engineering, or electrical engineering, along with students pursuing a minor in computer science, math, applied math, data analytics, or security, or a robotics concentration.

Activity 1: Name Signs with Badges

In the Fall 2020 semester, with two sections of primarily first-year students in programming 1, the author wanted to improve learning student names, having the students meet/know each other, increase the amount of groupwork, improve teamwork, increase leadership skill building, and increase EM practices within the classroom. With facemask mandates, social distancing, hybrid courses (for those who cannot join for health or quarantine reasons), and online submission being the norm, this seemed more pressing, challenging, and essential than previous semesters. To achieve these goals, each student created a sign with their first, last, and preferred name, an "avatar" picture, team affiliation (i.e., an animal of the team's choice), and a blank area to display "in-class" and share on the message boards of the content management system (CMS). The students are provided with a sample "instructor" sign (figure 1). This is a first week homework assignment where students submit a screenshot / picture of their sign.

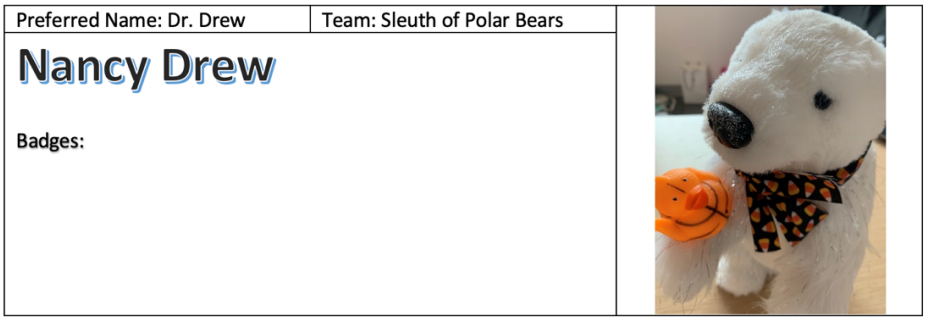


Figure 1: Given Example Name Sign with Space for Badges

The blank area is intended to display badges that students earn over the course of the semester. Given the stress and mental health issues of COVID-19, the author wanted a gamified activity that was causal, informal, and optional. Students can "earn" badges by completing a range of various tasks. Some badges were tied to course performance (e.g., Analysis Assignment Rockstar, Quiz Rockstar, Debugging Wizard). Other badges are team performance focused (e.g., Great Teamwork, Great Team Player, Leadership). The final category of badges is for students who demonstrate strength of character (e.g., Recognizing Needing Help, Phoenix, Random Act of Kindness). A selection of badges from programming 1 are shown in figure 2.

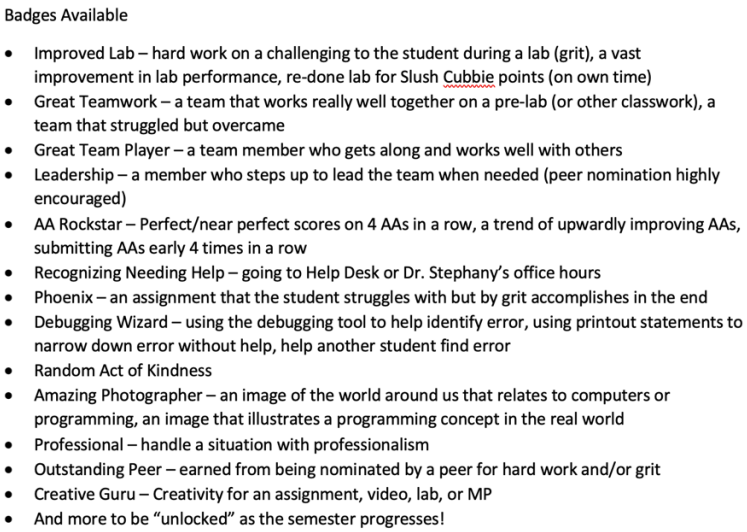


Figure 2: Example Badges from Programming 1 Course

The instructor places a "0%" weighted column into the CMS gradebook and turns notifications "on" for this column to "notify students when changes are made". The instructor can simply increment the "counter" in this grade column (the maximum grade is the total number of badges available). In the comment section, the instructor adds the recently earned badge name. The students are encouraged to write (or draw) each badge earned onto their name signs.

Below are the specific EM/KEEN course objectives [3] that students will gain with throughout the course if you implement this type of system.

KEEN Related Course Outcomes/Learning Objectives [3]:

- Develop an appreciation of hard work & recognize the benefits of focused and fervent effort
- Accept responsibility of their own actions and credit the action of others
- Demonstrate an ability to set, evaluate, and achieve personal & professional goals
- Be able to teach and learn from peers
- Be able to network and see the value of others
- Consider a problem from multiple viewpoints
- Articulate an idea to diverse audiences
- Modifies an idea/product based on feedback

Lessons Learned from Activity 1 and Suggested In-Person Expansions

These signs had two advantages. The first was administrative: it made taking attendance via pictures for each class easy to identify who was where at each class period (without assigning seating). The second was for the team members to more easily identify each other. Given they were first-year students and did not know each other, this introvert-friendly technique helped teams and encouraged networking among the students regardless of them meeting online via Google Meet or in-person during labs. (The course was set up for students to complete pre-labs, 2/3 of the homework assignments, class related activities, and one project with their teammates).

The badges often helped as stress relief and occasionally rewarded students for "grit". These badges being a 0% column in the gradebook allow students to access and see their achievements whenever they wanted. Additionally, they would receive a notification from the system when the badge was "entered". The idea behind this badge system is to encourage students to work hard and achieve various skills through the semester. The badges are like those often earned in a video game. Although not all programming students are huge video game players, many of them are, and most have dabbled a little bit. The badges help encourage students to go the extra mile – like help a peer or identify good work in a peer.

In non-COVID-19 circumstances, this activity could be expanded to include badges in the form of custom printed stickers or small toys (e.g., a plastic bug for debugging badge). The students could place these directly onto their name signs. Blank stickers (or labels) can be purchased at office supplies stores and are easy to design using free templates. Small toys can be found in the party supply area or the educational toy section.

Activity 2: Tik-Tok-ing Your Way to Understanding Computer Science Concepts

The assignment write-up for this activity begins with: "Are you a fan of TikTok? YouTube? Do you find it more helpful to watch a 5-minute video on how to do something? Students often learn material more deeply by teaching other people the material." Students in 3-4 person teams generated two videos on programming 1 material and one improvement video at the end of the semester. This allows students to be value creators within this course and even create some of the course content. Students' sign-up for the topics from instructor generated lists (figure 3) on a first-come, first-serve on a CMS online forum during the second week of semester.

Topic List A	Topic List B
Arrays	Defining and Using Pointers
2D Arrays	Arrays and Pointers
Vectors	C and C++ Strings
Functions as Black Boxes	Dynamic Memory Allocation
Functions (without value passing/return values)	Arrays and Vector of Pointers
Parameter Passing with Functions	Structures
Arrays and Functions	Pointers and Structures
Functions with Return Values	Command Line Arguments
Variable Scope	Random Access and Binary Files
Global Variables	Parsing and Formatting Strings
Recursive Functions	Implementing a Simple Class

Figure 3: List of Video Topics from Programming 1 Course

Each group generates two short (approximately) 5-minute videos. After the student receive confirmation for their topics, they begin by researching that topic using their textbooks and suggested online resources. Each of the 5-minute videos contained the same content:

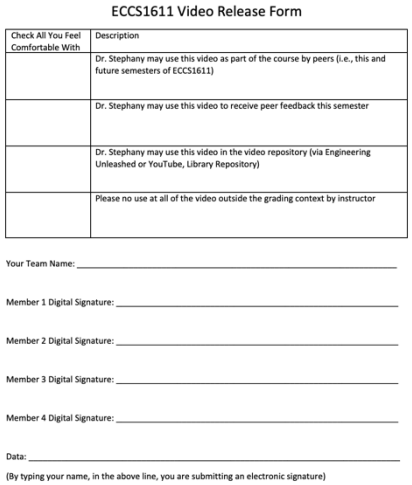
- Definition/Description, Use
- Simple Code Example
- Picture/Example from Real-Life that illustrates the concept
- 2 (potential) Exam Questions

The students are instructed that the aim of each video is to introduce the Programming 1 Topic to their peers. The students can assume the videos are viewed in the correct order (i.e., the same order as shown in the textbook).

The students can introduce the topic in any method that they wish. Suggested methods include: "lecture" on the topic via Google Meet with shared screen slides, displaying the slides behind them, using animation, puppets, drawings, or graphics to illustrate concepts, a mini game (flashcards, Kahoot, or similar), a demonstration, a rap, debate, poem, a story, or a musical. The students are challenged to be as creative as possible but there are a few additional criteria: (1) the video must be rated PG and under and (2) not all team members need to appear on screen, but all group members should participate equally in creating the video.

The instructor generated one basic video that covered the aforementioned content (definition, simple code example, real-life example, and two exam questions) for a topic that did not appear on the student option list. Additionally, the instructor provided a generated video that did not cover the content but used a more innovative approach to demonstrating the concept. This provided students with a successful and unsuccessful model to help them achieve a meaningful and effective assignment [5].

Each team submits a link to the video and their completed Video Release Form (figure 4). All students received criticism and feedback from the instructor based on a rubric [4]. For students who allowed peer feedback, videos were released to the entire class and the creators received a composite form of the feedback given by their peers. Students who opted to not participate in this process were given an alternative in-depth self-reflection assignment. After receiving instructor and peer feedback, all students completed a self-reflection assignment. The students evaluated the video, their individual work on the video, and the work of the team. The focus was on ways to improve future videos based on the provided feedback. Each reflection was approximately two paragraphs.



ECCS1611 Video Release Form

Check All You Feel Comfortable With	Description
	Dr. Stephany may use this video as part of the course by peers (i.e., this and future semesters of ECCS1611)
	Dr. Stephany may use this video to receive peer feedback this semester
	Dr. Stephany may use this video in the video repository (via Engineering Unleashed or YouTube, Library Repository)
	Please no use at all of the video outside the grading context by instructor

Your Team Name: _____

Member 1 Digital Signature: _____

Member 2 Digital Signature: _____

Member 3 Digital Signature: _____

Member 4 Digital Signature: _____

Data: _____

(By typing your name, in the above line, you are submitting an electronic signature)

Figure 4: Video Release Form from Programming 1 Course

Below are the specific EM/KEEN course objectives [3] that students will gain with throughout the course if you implement this type of system.

KEEN Related Course Outcomes/Learning Objectives [3]:

- Take ownership of, and express interest in topic/expertise/project (curiosity)
- Produce effective verbal presentations (communication)
- Provide and accept constructive criticism, including self-evaluation (communication)
- Be able to teach and learn from peers. (collaboration)
- Modifies an idea/product based on feedback. (creating value)
- Connect life experiences with class content (connections)

Lessons Learned from Activity 2 and Suggested In-Person Expansions

Many of the students seemed to enjoy the non-standard assignment options during the course, but the author believes that was not true for all the students. Approximately half of the students elected to receive peer feedback, allow the videos to be available for peers to use, and allow the videos to be placed into a repository. If this was presented in-person, it could be fun to have students submit and watch various "bloopers" or vote in-person for "best" and "most creative". (This can also be done online, but in-person has an additional level of excitement).

Activity 3: Play-Doh to Make Connections with Material

Students receive a small container (i.e., a party pack size) of Play-Doh during the first week of class (supplied by the instructor). The students use this Play-Doh several times during the semester to assist them in visualizing programming concepts and build connections with objects and concepts from everyday life. Introductory programming students often find concepts to be a bit abstract. The aim of this tactile exercise is to assist students by giving context to these abstract concepts.

There are four different assignments written using the Play-Doh: (1) Play-Doh and Variable Sizes (figure 5), (2) Play-Doh and Assignment Statements, (3) Play-Doh and if Statements, and (4) Play-Doh and Functions. Each assignment requests that the students develop several visual representations of the concept using the Play-Doh as an aid. The students submitted pictures of their solutions to the CMS.

Assignment 1: Play-Doh and Variable Sizes

Meet with your team online (or appropriately socially distanced in-person). Everyone will need to bring their container of Play-Doh passed out in class. Using your textbook or looking it up online, find the sizes for the various C++ variable data types in terms of **bytes**. (If you look it up online, make sure you specify C++ programming language). Brainstorm and create 2 or 3 ways to visually represent the variable sizes using the Play-Doh as your medium. Be as creative as possible – it can be simple or complex. 2D or 3D. Take pictures and post onto Moodle. [Reminder: Hold onto the Play-Doh, you will be using it again!]

Figure 5: Assignment 1 with Play-Doh from Programming 1 Course

Below are the specific EM/KEEN course objectives [3] that students will gain with throughout the course if you implement this type of system.

KEEN Related Course Outcomes/Learning Objectives [3]:

- Connect life experiences with class content

- Be able to self-reflect and evaluate preconceived ideas, thoughts, and accepted solutions
- Take ownership of, and express interest in topic/project/experience
- Present technical information effectively
- Be able to teach and learn from peers

Lessons Learned from Activity 3 and Suggested In-Person Expansions

These assignments where a welcome activity during the pandemic and really seemed to help the teams communicate. The students seemed to "chat" while working on these assignments as they worked with the Play-Doh trying to figure out various representations. In the spring semester, when the author asked the students to recall the variable sizes – it appeared that more students seemed to remember then previous semesters before this activity existed. The author's suggestion for switching this to an in-person / in-class assignment and add a "gallery walk" allowing the students to see those solutions created by other teams.

Future Work and Next Steps

This research is the preliminary stages with activities being piloted in the Fall 2020 semester. Next the author will be applying for an IRB, so the author can conduct preliminary research during the Fall 2021 term including gather student survey information, grades, course evaluation comments, etc. The author plans to continue this research each fall for the next 5 years. Using this information gather, the authors will attempt to gain insight into the helpfulness of these activities grade-wise instead of only anecdotal.

Available Resources

An information repository (i.e., "card") is available on the Engineering Unleashed website: https://engineeringunleashed.com/card/2403 [6]. The card provided the instructional material mentioned within the paper including the full assignment write ups, examples of student work, instructor notes, rubrics, and an hour-long video presentation of the materials. All of these materials can be freely downloaded, modified, and used.

Resources

[1] "KEEN - Home," KEEN - *Home*. [Online]. Available: https://engineeringunleashed.com/. [Accessed: 06-Mar-2021].

[2] *KEEN - The Framework*. [Online]. Available: https://engineeringunleashed.com/framework. [Accessed: 08-Mar-2021].

[3] J. B. Hylton, D. Mikesell, J.-D. Yoder, and H. LeBlanc. (2019) "Working to Instill the Entrepreneurial Mindset Across the Curriculum," *Entrepreneurship Education and Pedagogy*, vol. 3, no. 1, pp. 86–106.

[4] Coffman-Wolph, S., & Gray, K. (2020, June), Work in Progress: Student-generated Material for Artificial Intelligence Course Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual Online. 10.18260/1-2—35685

[5] Miller, H. (2007). Designing effective writing assignments. Teaching with writing. University of Minnesota Center for Writing. Retrieved Mar. 8, 2021, from http://writing.umn.edu/tww/assignments/designing.html

[6] Coffman-Wolph, S., "Online/Hybrid/Flipped EM Learning in a Programming 1 Course and Beyond (2021 KNC Session)." Engineering Unleashed, https://engineeringunleashed.com/card/2403.