

REFLECTIONS FROM FIRST TIME BLENDED PROGRAMMING INSTRUCTORS

TEACHING CHALLENGES AND LESSONS LEARNED!

SHORT AND  – LAURA REID & KEMI OLA

THE NEW “BLENDED” COURSE:

- Approachable Apps: A Gentle Introduction to Programming using JavaScript
- Blended:
 - 2 Hour Flipped Classroom EVERY OTHER WEEK
 - Groups of 3 worked on activities.
 - Lectures – all online
 - Labs – all online
 - Quizzes – all online
 - 3 Assignments – worked individually



GOAL OF THE COURSE

- To show students who had never written one line of code before that:
 - Coding is NOT magic
 - Coding is FUN!
 - You can code with just Notepad and a browser → Nothing special needed!

TEACHING/BLENDED CHALLENGES:

1. Assessing People Online → Online Quizzes?



2. Drinking from the  of information about how to create a
AWESOME Online Course → Go Agile!

3. The Nature of Our Material – Programming for people who have NEVER
programmed before → Online caused some problems ☹️

4. Flipped Classroom → Shy People?

LESSONS LEARNED – 1. ONLINE QUIZZES ARE GOOD NOT EVIL!

- “Online Quizzes” – What if students work together? Or find answers online?
- A “process” is still happening...students are still learning, so maybe think of the quiz as a *learning* tool rather than *assessment* tool (don’t make the quizzes worth much)
- “Cheating” is OKAY here!



Sound Clip from: lynda.com



Image from: www.educationnews.org

LESSONS LEARNED – 2. GO “AGILE”

- For the first release of the course, go AGILE, all that matters is content!
- Worried to much about the process and not enough about the content
- Write down the course objectives and then DIVIDE and CONQUER!



From: dilbert.com

LESSONS LEARNED – 2 ½ SHORT VIDEOS FOR THE LECTURES

- Again → DIVIDE and CONQUER!
- None of my videos were more than 15 minutes, this was HARD!
- Tell students how long each video is before they watch it.
- Each video should just address ONE topic/task/idea
- DO NOT JUST RECORD A 50 MINUTE LECTURE!

The screenshot shows a Blackboard LMS interface for a course. The browser address bar displays the URL: <https://owl.uwo.ca/portal/site/4ee3a6a7-bbcb-4917-a012-7daeac41d648?panel=Main>. The page header includes the Western University logo, "My Workspace", and course information: "COMPSCI 1046A 201 SU15" and "Science Outreach". A navigation menu on the left lists: Home, Syllabus, Week By Week (selected), Assignments, Announcements, Schedule, Gradebook, Tests & Quizzes, Forums, Drop Box, and Help. The main content area is titled "Lectures > Week 1 > Lectures" and "WEEK 1: Programs and HTML". Below the title, it states: "Upon completion of the lecture videos and labs for week 1, you should be able to:" followed by a list of learning objectives: define what a program is, view the HTML for a webpage, identify and explain the purpose of at least 10 HTML tags, identify an HTML element, identify the parts and attributes of an HTML tag, create a simple webpage using HTML with a title, heading, paragraph, image, and link, and validate and verify the HTML using an external website. A list of video links follows, each with a duration: "Introduction to the Course - Will happen in the first class on Wed, May 6", "Video 2 - What do I need - 4 minutes", "Video 3 - HTML vs CSS vs JavaScript - 5 minutes", "Video 4 - Viewing some HTML tags - 7 minutes", "Video 5 - Creating your first webpage - 6 minutes, 20 seconds", "Video 6 - Creating lists in a webpage - 1 minutes, 40 seconds", "Video 7 - Adding an image to a webpage - 3 minutes, 50 seconds", "Video 8 - Adding a link to a webpage - 4 minutes, 30 seconds", "Video 9 - Creating a table on a webpage - 4 minutes", "Video 10 - Getting input from the user - 4 minutes", and "Video 11 - Checking your HTML tags - 8 minutes". A "Back" button is located at the bottom left of the video list.

LESSONS LEARNED – 3. WE SHOULD HAVE HAD ONLINE CONSULTING TIME SLOTS

- Maybe not for all sciences but for computer science, if a student had this:

```
if (x = 5) {
```

but should have had this:

```
If (x == 5) {
```

this sort of mistake is often VERY hard to figure out by yourself if you have never programmed before...very easy to throw in the towel!

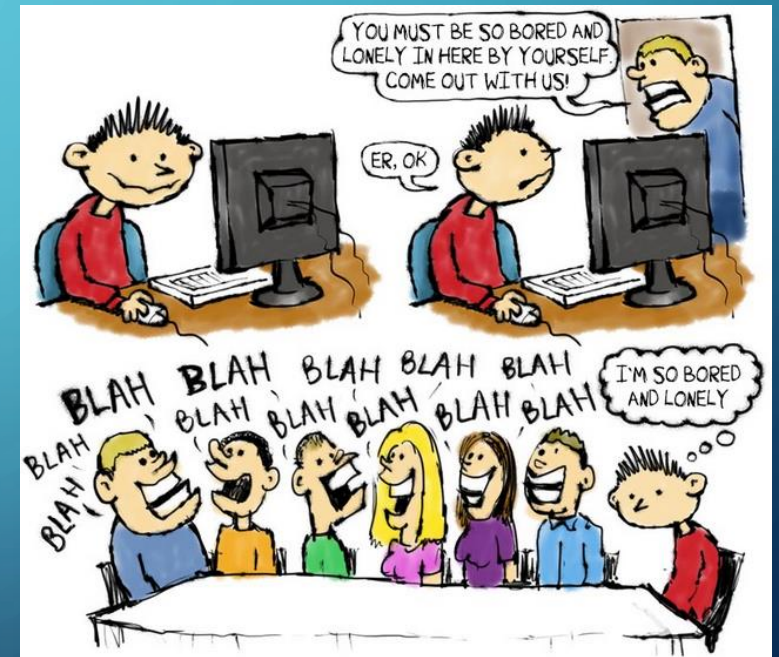


LESSONS LEARNED – 4. FLIPPED CLASSROOM, HAVE A: PRETEST WITH ONE OR TWO QUESTIONS

- When the flipped classroom worked it was AWESOME!
- Can't solve shyness but we can help with experience. Next time we will have a pretest, before the course starts with questions like:

1. *Have you ever written one line of code before?*
 - a) YES
 - b) NO
2. *Are you a good problem solver? Do you LOVE logic puzzles/problems?*
 - a) YES
 - b) NO

- Then we will try to put at least one person who said YES to one of the questions above in each group of 3



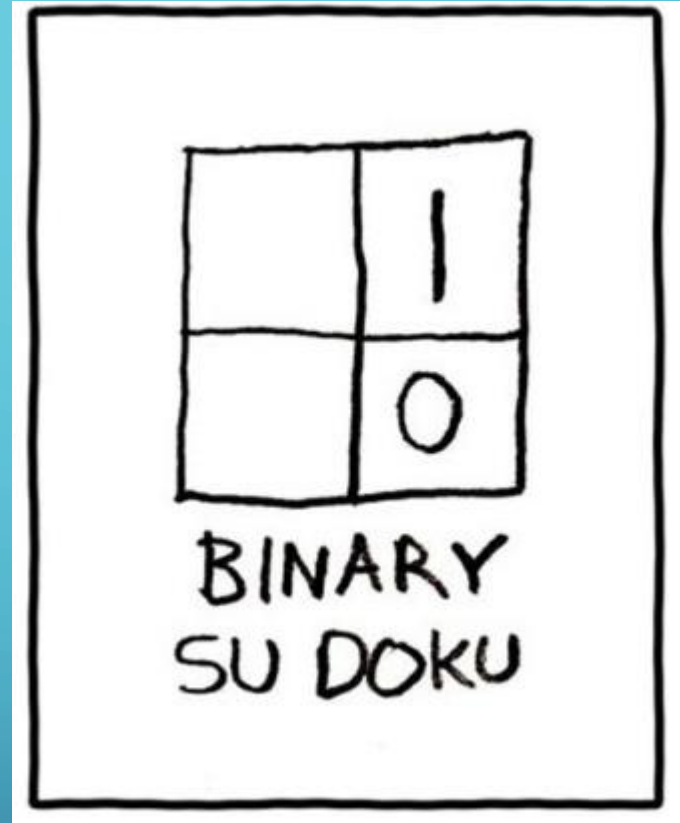
From:
<http://sociallyawkwardmisfit.com/post/90716578322/sociallyawkwardmisfit-com>

LESSONS LEARNED – ALLOW FOR CREATIVITY, MAKE IT FUN!

- We tried to use as many games as possible to teach concepts like:
 - Jeopardy
 - Hangman
 - Mad Libs → [One of the actual assignments submitted by one of our students](#)

THANK YOU!

IN CLOSING, PLEASE ENJOY SOME
COMPUTER SCIENCE HUMOUR →



From: xkcd.com