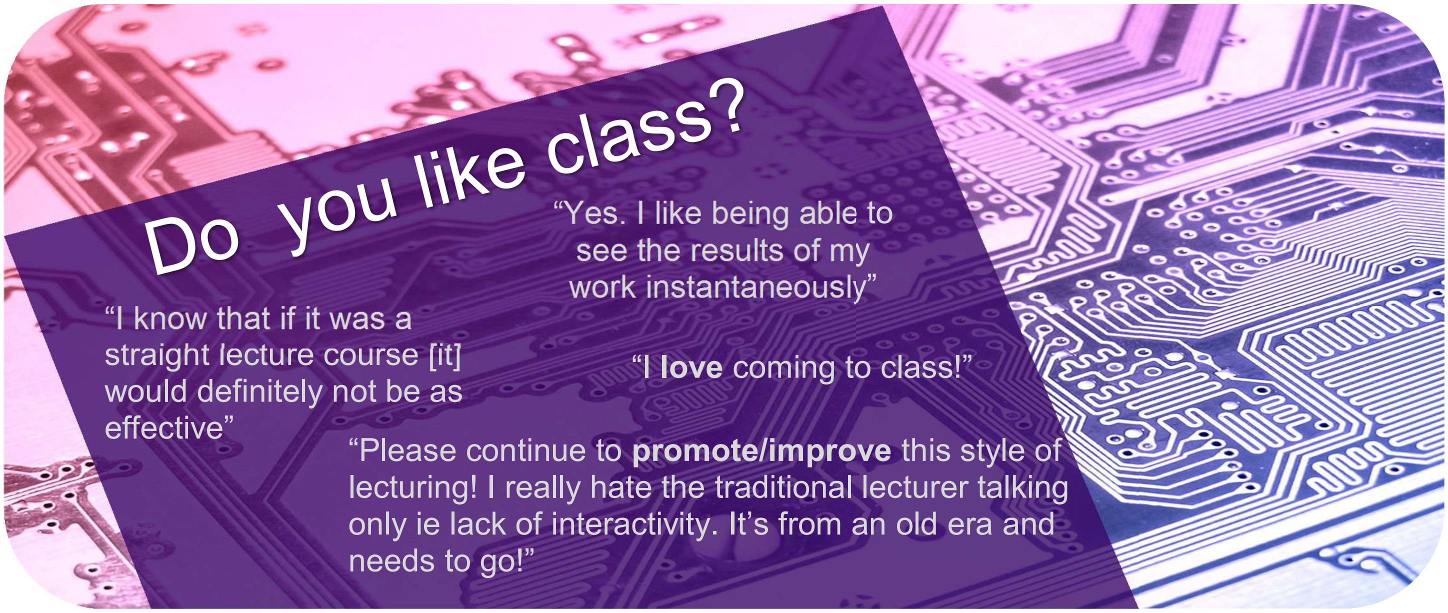
Great ideas for teaching

DEBUG THIS!

Jenna Butler



Do you like class?

"Yes. I like being able to see the results of my work instantaneously"

"I know that if it was a straight lecture course [it] would definitely not be as effective"

"**I love** coming to class!"

"Please continue to **promote/improve** this style of lecturing! I really hate the traditional lecturer talking only ie lack of interactivity. It's from an old era and needs to go!"

CS2120: Computing for Informatics

How often do you have to kick students out of the classroom because the next class is coming in?

I never have, until this year, when I taught Computer Science 2120 using the microteaching method[1] and completed a "debugging contest", held via our in-house created website, for my students.

Microteaching allows students to get the most of the lecturer. They don't just sit and listen, the regularly engage with the lecturer and their peers, to immediately practice what they are learning. Pairing this concept with technology turns an ordinary programming class into an exciting, engaging and empowering experience where students leave each class excited for the next.

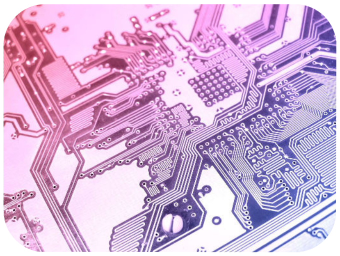
I saw this most clearly when I ran a debugging contest with my students. This course is an introduction to Computer Science and after learning the basics, it was time to apply them to "real" programs. A great way to practice everything you've learned is to "debug" a broken program – that means fixing the errors in code to make it run correctly. Debugging is often considered boring and time consuming, so to make it fun I held a contest in my class. The class before we learned various techniques to debug a program, then I told them they would be debugging a series of programs in teams next lecture.

Actual results from an anonymous survey done in CS2120:

"The class is very enjoyable and the teaching method of doing activities throughout the class keeps me very engaged."

"I am extremely pleased. I can't imagine anybody wouldn't be"

"I love coming to class. I enjoy how we get to explore different functions in class and learn them by experimenting with them."



Learning goals:

* How to work with technologies they will encounter in the field

* How to debug a program

Key Concepts:

* Use technologies they will see when they graduate

* Make it fun – use a contest or some kind of game

Resources needed:

* Access to technology used in "real world"

* Time to create a problem they can solve throughout the class

* One full class to dedicate to the contest

Submitted by:

* Jenna Butler
* PhD student in Computer Science
*Limited Duties Lecturer
*jcamer7@uwo.ca

We had a course website that was created using Sphinx - a documentation generator that is used to create all the help and documentation for the Python language (what we teach in the class) that is available online. To get the class used to using real-world documentation, the course notes were generated using Sphinx by Professor Daley (who created the class last year). I used Sphinx to generate my debugging contest website and uploaded a host of files to debug. I put it on our course website but made it invisible to the students by changing the permissions. Then, at exactly 2:30pm when our class started, I showed them the website and BOOM! The files became visible! The students immediately downloaded the files and were able to start working on the contest.

The students spent the class using whatever technical resources they needed (build in debugger in the Python language, online help files, stackoverflow, etc) to solve the challenging puzzles. I saw my classroom come to life. As I walked through the classroom I would hear students engaging with each other saying things like "What was it we learned? That we don't need brackets to pass in functions right?" "Right!! Try that!!". They were so excited!

When the clock struck 3:20 it was time to turn in the files. I had created an assignment on Edmodo (the classroom management system we used to accept assignments) and the students were to upload their solutions within 15 minutes of the end of class. At 3:25, I had to start asking people to leave as they were still trying to solve the problems as the next class came in!

The next class I had again created a hidden part of our website with the winners name announced. I opened the website at the start of class and revealed the debugging champions – the team that earned some delicious TimBits! The class was excited to see the results and learned a great deal about programming. Debugging requires you to understand all of the concepts but more importantly, put them together in a new way.

Technology played a part throughout not just this class but all of our classes. We held polls in the middle of class using Edmodo; students would email me their code solutions to in-class activities mid-class and I would run it on the projector to show everyone; and I created a "mission impossible" mission video to get my students pumped up to study for exams! Others can use these ideas in their class too. All it takes is time to examine your field and figure out what technology your students are going to need to use when they leave Western – then turn that into a part of your classroom. Whether you are using the documentation style they will encounter (like Sphinx) or giving them a real-life test situation (like debugging under a tight time line) they will appreciate learning something that will be part of their daily life when they graduate and enter the work force.

1: Deslauriers, Louis, Ellen Schelew, and Carl Wieman. "Improved learning in a large-enrollment physics class." *science* 332.6031 (2011): 862-864.