CSE VLC Session 5
Flipped Classroom and Peer Instruction
Homework Assignment

Instructions: Post to Session 5 folder of our web portal
Due: Monday at Noon, November 25th

Course Title

COP3515 IT Program Design, USF, Alessio Gaspar

Flipped Classroom

A Flipped Classroom is a pedagogical approach where students acquire knowledge before coming to class and engage in activities during class. Have you tried a flipped classroom? If so, please share the following in your homework assignment.

1) What do students do to prepare outside of class? Videos? Reading assignments? Homework exercises?

In the hybrid version I used a few years back, students were assigned readings from the textbook every week, along with a few videos when these were needed to either clarify or expand on the reading assignment.
There was no reading comprehension quiz used. Instead, students were rewarded for finding questions on their reading assignment on the parts which were the most difficult to them, and posting them on a so-called "peer learning" forum. They were also rewarded for addressing questions from their peers.

The following advice was provided to help them understand how this process would help them learn;

- Read a first time to get the lay of the land, write down the main items, revisit them with a focus on identifying the hardest to understand parts or simply what you see might be deeper than your first reading led you to believe, note down questions you may have including "what if" scenarios, post your questions on the forums.
- For the next 1-2 days, keep an eye on the forums for this module. When others post questions you didn't think of you might be in one of the following situations;
- You didn't think of this question because you already understood the material. Take a minute to share your insights with the poster. This will allow your instructor to verify your perspective is right and help you adjust it if it turns out you missed something
- You didn't think of this question because you thought you understood the material but apparently missed something. You now have one more question to address in the next step

- Within 1-2 day of your first read, review the same material with a focus on addressing your own questions but also those from your peers. Post in the forum to share your thoughts. The idea is not necessarily to solve everything but to "think out loud". Remember that anything you post will be reviewed by the instructor so you open the possibility of a didactic dialog during which your hypothesis will be validated or corrected. If you keep your misunderstandings or understandings to yourself, no feedback may be provided.

2) What do you have students do inside of class instead of lecture?

The questions and responses to the peer learning forum would be used to prepare mini-lectures during the weekly class meeting.
These mini-lectures would focus on the very topics which proved to be problematic to students; e.g. questions asked in the PLF, incorrect responses provided…
The duration would be short, e.g. 30 minutes maximum, with occasional prompts for the students to respond to clicker-like questions to ensure of their understanding of the difficulties associated with the topic being reviewed.

3) Provide 1-2 resources would you recommend for a professor considering this approach?

- http://www.peerinstruction4cs.org/
- http://www.danielzingaro.com/pics.php

Peer Instruction

Peer Instruction is a pedagogical approach where students spend much of the class time responding to multiple-choice questions and discussing answers in groups of 2-4.

In preparation for our next virtual meeting:

Read:
Simon et. al. 2010. Experience Report: Peer Instruction in Introductory Computing. *Proceedings of SIGCSE Symposium*.

Crouch & Mazur. Peer Instruction: Ten years of experience and results. *American Journal of Physics*. 69 (9), September 2001.

Watch:
There are many good videos posted by the Carl Wieman Science Education Initiative
http://www.cwsei.ubc.ca/resources/SEI_video.html

In particular:

- Student and Teachers Speak about Clickers

- How to use Clickers Effectively
- Anatomy of a Clicker Question

Write:
Create three clicker questions for your course that include appropriate distracters (wrong answers) and that are not too easy or too hard for students to do on their own.

Question #1

Let’s consider the following programs. Which ones will display the value “3” on the screen? Keep an eye out for possible syntax or logic errors!

Program #1

```
int n;
for(n= 0 ; n < 3 ; n++) {};
printf("%d\n", n);
```

Program #2

```
int n;
for(n= 0 ; n <= 3 ; n++) {};
printf("%d\n", n);
```

Program #3

```
int n;
for(n= 0 ; n =< 3 ; n++) {};
printf("%d\n", n);
```

Program #4

```
for(int n= 0 ; n < 3 ; n++) {};
printf("%d\n", n);
```

Program #5

```
for(int n= 0 ; n <= 3 ; n++) {};
printf("%d\n", n);
```

Program #6

```
for(int n= 0 ; n =< 3 ; n++) {};
printf("%d\n", n);
```

Question #2

The following program continuously reads data from the user which is stored in a dynamically size array of integer. When we have an array ready, we pass it to a *doSomething* function written by another programmer. We then proceed to get the next array of integers from the user. In which of the numbered slots would you insert a *free(data);* statement to address the memory leak?

```
// user-provided size of the int array to allocate
int size =0;

// pointer to the dynamically allocated int array
```

```
int * data = NULL;

do {
        printf("How many integers in your array?\n");
        printf("Enter 0 to end this program\n");
        scanf("%d", &size);

        if( size <= 0 ) break;

        // allocating the array based on requested size
        data = (int*)malloc(sizeof(int) * size);
        if( !data){
                perror("Unable to allocate int array");
                exit( EXIT_FAILURE );
        }

        // SLOT #1

        printf("Let us fill this array with integers:\n");
        for(int n= 0 ; n < size ; n++){
                printf("data[%d] \t=\t") ;
                scanf("%d", & data[n]) ;

                // SLOT #2
        }

        // SLOT #3

        // we send off this data to another part of the program
        doSomething( data );

        // SLOT #4

} while(1);

// SLOT #5
```

Question #3

The other guy[TM] just wrote an unnecessarily convoluted function

```
int mystery(int v1, int v2){
        if( (v1 < 1) || (v2 < 1) )      return 0;
        if( v1 < v2 )                   return mystery(v1, v2-1)+1;
        if( v2 < v1 )                   return mystery(v1-1, v2)+1;
        return mystery(v1-1, v2-1);
    }
```

What is the largest number of activations records at any time for the above-function which will be pushed on the program's execution stack if we use *mystery(5,9);* from the *main*

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15