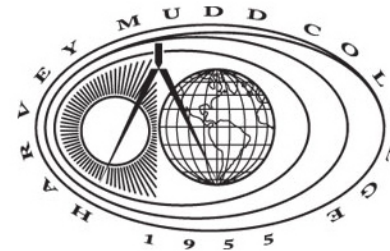


# Christine Alvarado

- Current position: Lecturer with Security of Employment, UCSD (1.5 years)
- Previously: Associate Professor of Computer Science, Harvey Mudd College (7 years, with 1 sabbatical)

UC San Diego

Large public school  
~500 students, per year in  
Computer Science!



HARVEY MUDD  
COLLEGE

Small private school  
~800 students, total

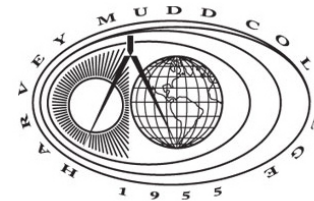
Q: What do these schools have in common?

A: My job description

UC San Diego

2 courses per quarter  
= 6 courses per year

Teaching occupies ~75% of my time.  
Other expectations include “scholarly activity”, service and contributions to diversity



HARVEY MUDD  
COLLEGE

2.5 courses per semester  
= 5 courses per year  
Teaching occupies ~75% of my time  
Other expectations include  
“scholarly activity”, service and  
contributions to diversity

# What courses have I taught?



Intro programming (in Java and Python; for advanced students and beginners)

Intro programming 2 (CS 1.5)

“Baby” data structures

Advanced data structures

Computer Architecture

Operating Systems

Programming Languages

Artificial Intelligence

User Interface Design

Pen-based Interfaces

Teaching Methods in Computer Science (TA training)

Summer programs for high school students

...

Perhaps a better question is what have I *not* taught

Do I get a choice? Yes and no...

# Who is my audience? Do they want to be there?



- Typically I teach to CS majors, but many of my lower div courses have non majors who are required to take the class
- My students want to be there because I convince them that they want to be there

# 3 ESSENTIAL TIPS FOR HOW TO LOVE TEACHING AND STAY SANE



# Limit your new preps



- Make a 2-3 year plan with your chair
- Teach each class at least twice (preferably more)
- Limit the number of different classes you teach as much as possible for the first 3-5 years

# Put hard boundaries on your prep time



- My first year... I saw a lot of 2's and 3's in the wrong digit on the clock
- My solution:
  - ▣ Do my class prep in the N hours before class
  - ▣ Forced me to limit prep time
  - ▣ Usually class went just as well!

# Put hard boundaries on your prep time



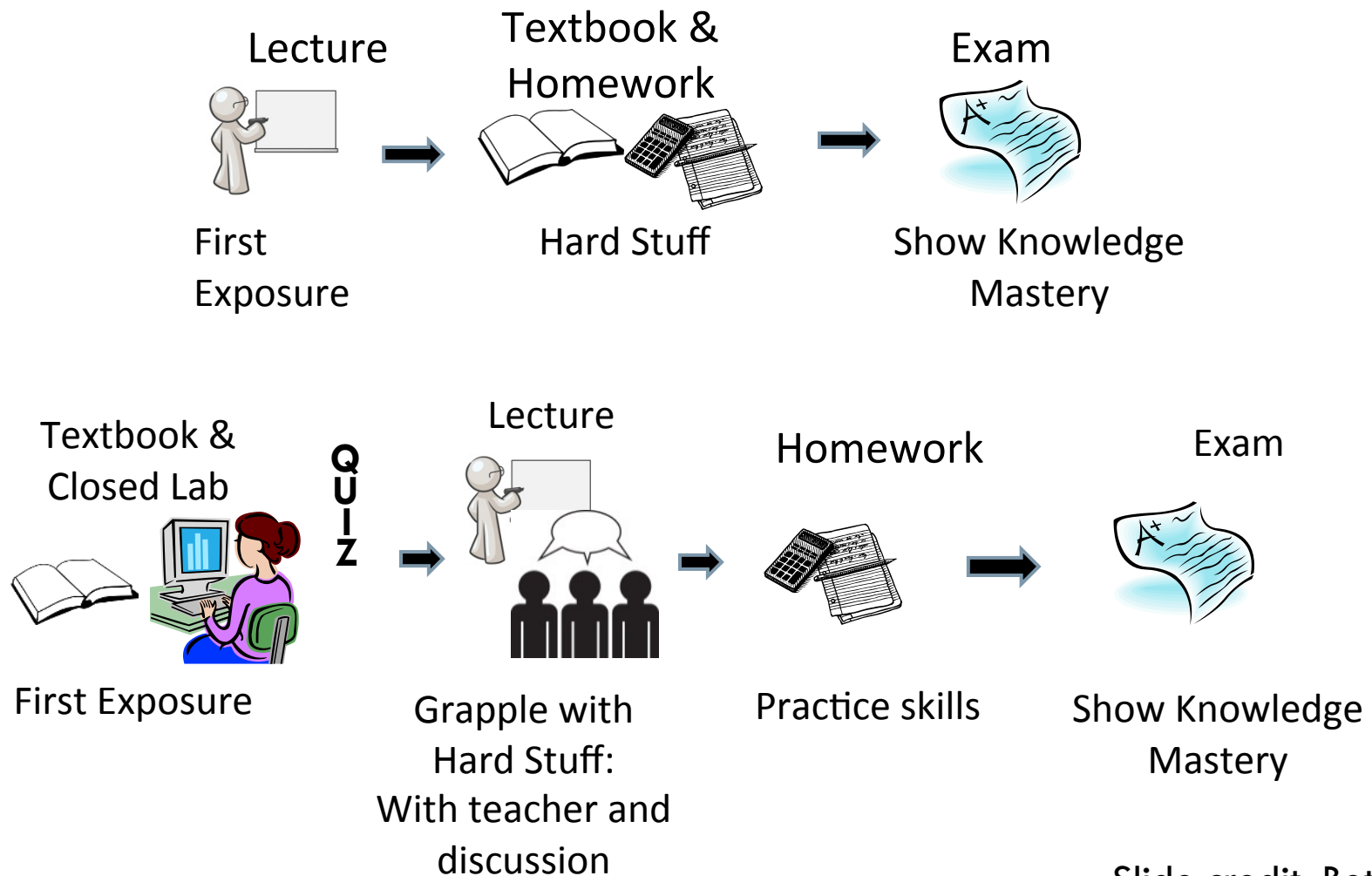
- Choose  $N$  that is comfortable for you ( $N$  between 1 and 3 seems to work well)
- Leave (some of) your class prep for the  $N$  hours right before class
- You will get better at fitting things into this limited time
- Then you can move the hours around! (I now typically prep well before class, but I am good at keeping the work within the bounded time)

# “Flip” your classroom! (Or, rather, just make it active)



- Specifically, consider a technique called Peer Instruction

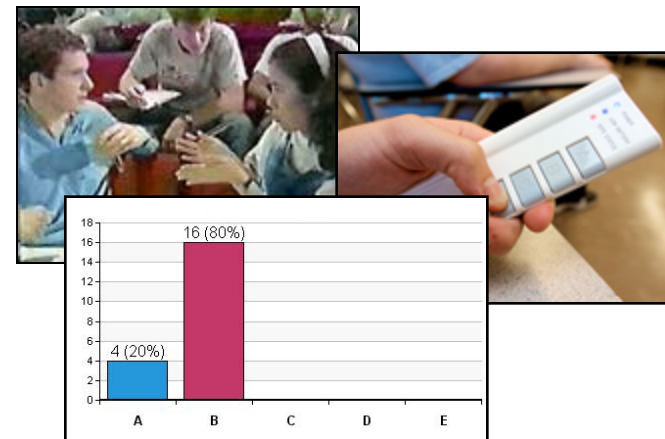
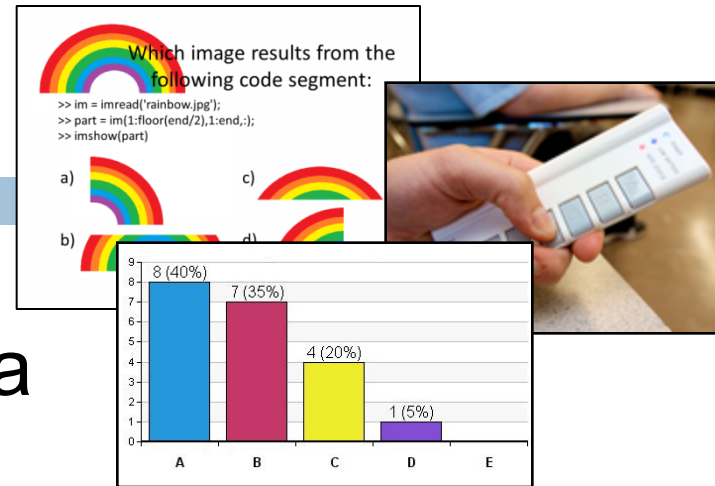
# Flipped classroom with Peer Instruction



Slide credit: Beth Simon

# Peer Instruction

1. Students individually consider and respond to a multiple choice question
2. Students discuss the *same* question in groups, then submit another response
3. Instructor guides students in a class-wide discussion



# Example: Peer Instruction

```
public static boolean contains( int[] a, int num )
{
    for ( int x : a )
    {
        if ( x == num )
        {
            return true;
            System.out.println( "Found it!" );
        }
    }
    return false;
}

public static void main( String[] args )
{
    int[] myA = {2, 4, 6, 1, 15};
    MyClass.contains( myA, 1 );
}
```

What will be printed?

- A. Nothing
- B. "Found it!"
- C. "true"
- D. "true"  
"Found it!"