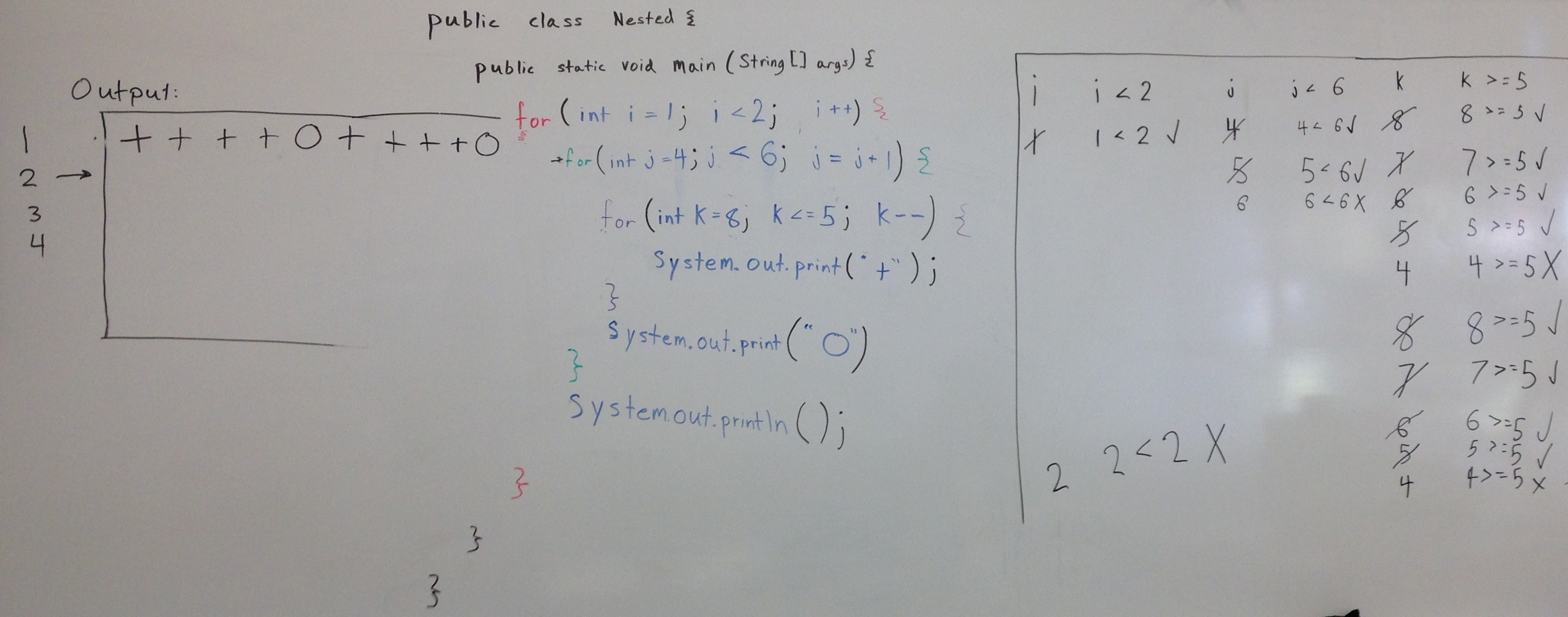
**Plans for today:**

* Talk to students about how the class is structured
  + We are going to be using a method over learning programming called the box method.
  + Both professors **do not** **go** **over this in class.** They use it loosely, but deviate from it.
  + The book cannot be read for context like you can with a history book. **Everything** from the book needs to be taken as fact. Important rules are often introduced in only a single sentence in an entire chapter, and although only introduced once in the whole book, these rules are so fundamental to programing that they keep coming back in every example program from that point on.
    - Students are not going to be able to memorize all of these rules on the first go.
    - Solution: Create a comprehensive handout with all of these rules. It’s difficult enough for students to have to memorize the rules and then apply them based on context. Having them search for it in the book has been proving to be too much work.
    - Suggestion: Read book and take notes of what you think is most important. Reiterate that the book is not a surefire way of learning everything in this course. Use it only to learn about rules.
  + Practice problems from the book are tricky.
    - The book examples attempt to teach multiple concepts at once, in a matter that can easily become confusing to students who are having trouble understanding how one rule alone works, let alone trying to figure out two in the same problem.
    - The book contains many trick questions!
    - Solution:
* Go over a single for loop
  + Talk about the three parts: Initialization, Condition and Update
  + Use the whiteboard to do this.
    - Show the entire box method for this problem
    - Help students to understand that, if followed correctly, the box method will always work, 100% of the time.
  + Afterwards, type the solution on the computer for students to see if it actually runs as we expect.
* Go over nested for loops
  + Ensure students that they can solve this much more daunting problem with all of the simple rules that were just introduced in the easier problem
  + Get to class 20 minutes early to write the whole problem on the board. I everybody is in class by then, start class early if possible.
  + Have the students work with the professor to solve the problem using the box method, which they just learned about.
  + This problem will take so long that students will naturally see the importance of writing every single change in the program down in an effort to reduce the amount of memorizing needed to finish the whole problem.
  + After the 10th to 20th or so iteration, students will start to get comfortable with the whole routine.
  + Go over scope if time permits
* Discuss other types of variations to students, while emphasizing the fact that they are now prepared to tackle **any problem** involving for loops using what we learned today
* Throughout using the whiteboard, gauge the students’ understanding. Are people getting it?
  + Stop and ask for questions on every single event that takes place.
* Go slowly. We don’t have that much time, so compensate by writing and talking quickly while making sure to pause long enough for students to understand what is going on in each step. Give them the idea that the professor is not in a hurry to finish.

**How it went:**

I think our meetup went really well today. I feel that students are now ready for the next material that is being covered in class. Some students have already been asking questions about content that will be covered next time we meet. I wrote up a comprehensive for loop guide for students to use to review just in case they forget what we did in class today.

I want to ask the professors if they would consider allowing a special “formula sheet,” including all of the interpreter rules, all syntax specifications and documentation for all methods learned in class. Students shouldn’t be expected to memorize these things, but should be able to successfully trace output on their own and should also be capable of producing code correctly, so that it produces the correct output. I am currently working on this comprehensive guide, and will be giving it to all of my students to use as a reference guide, as opposed to having to search for the book or the internet for help on a single specific concept in programming. The last problem students need to worry about is having to memorize a whole new language, including its unique sentence structure through syntax, and all of it’s foreign words, or code statements. Including this comprehensive guide should get students to naturally memorize these constructs over time, while using them in their programs and when interpreting code. The way the course is set up now is the equivalent of asking students to memorize a list of 100+ math formulas in an unusual language that is strictly composed up of shapes.



Still, I have high hopes for this semester! I had a lot of students stay long after class to ask me questions. Thank goodness there isn’t another class after ours! I have sent out a comprehensive email including everything we’ve learned this week, and a preview of what we will be doing over the next week.