

6.149 Checkoff 4

<http://web.mit.edu/6.149/www/materials.html>

What to complete

Due: Tuesday, January 20, 2015 at 5 p.m.

Lectures 1-5 on MITx

`checkoff4_nims.py`, which contains your code for 4.1 The game of Nims/Stones.

4.0 Review: MITx, Week 1

Complete the exercises on MITx from Lectures 1-5. If you have not received an activation email from MITx, email Michelle (mszucs@mit.edu) **immediately** so we can get you access in a timely manner.

Ensure you are trying to access an MITx URL starting with `lms`; the `staging` URL will not work.

These exercises should be review. If you find they are taking you more than 30 minutes per lecture, please review the readings and lecture notes before attempting 4.1.

4.1 The game of Nims/Stones

In this game, two players sit in front of a pile of 100 stones. They take turns, each removing between 1 and 5 stones (assuming there are at least 5 stones left in the pile). The person who removes the last stone(s) wins.

Download `nims.py` from the website and open it up. In this problem, you'll write a function to play this game; we've outlined it for you. It may seem tricky, so break it down into parts. Like many programs, we have to use nested loops (one loop inside another). In the outermost loop, we want to keep playing until we are out of stones. Inside that, we want to keep alternating players. You have the option of either writing two blocks of code, or keeping a variable that tracks the current player. The second way could be slightly trickier, but it's definitely do-able!

Finally, we might want to have an innermost loop that checks if the user's input is valid. Is it a number? (Hint: try making a list of valid inputs or look up the string method `isdigit()`). Is it a valid number (e.g. between 1 and 5)? Are there enough stones in the pile to take off this many? If any of these answers are no, we should tell the user and re-ask them the question.

Don't forget to print who won, Player 1 or Player 2.

If you choose to write two blocks of code, the basic outline of the program should be something like this:

```
while [pile is not empty]:  
    while [player 1's answer is not valid]:  
        [ask player 1]  
        [execute player 1's move]  
  
    [same as above for player 2]
```

Be careful with the validity checks. Specifically, we want to keep asking player 1 for their choice as long as their answer is not valid, BUT we want to make sure we ask them at least ONCE. So, for example, we will want to keep a variable that tracks whether their answer is valid, and set it to False initially.

When you're finished, test each other's programs by playing them!