Please turn in answers to the following questions this Monday, in class.
For the next few questions, we will think about the implications of variables (like hopper and tapia) being references to immutable and mutable objects. Beside each print:, indicate what is printed.

1a. hopper = [19, 20, 21]  # some upcoming years
    tapia = [19, 20, 21]  # some classes of students
    print(hopper is tapia)  # prints:
    hopper.append(22)  ###
    print(hopper)  # prints:
    print(tapia)  # prints:

    Explain what is happening to hopper and tapia (if anything) on the statement marked ###.

1b. hopper = tapia = [22, 19, 20, 21]  # some upcoming graduating years
    print(hopper is tapia)  # prints:
    sorted(hopper)  ###
    print(hopper)  # prints:
    print(tapia)  # prints:
    tapia.sort()  ###
    print(hopper)  # prints:
    print(tapia)  # prints:

    Explain what is happening on the statements marked ###.

1c. hopper = tapia = 'upcoming years'
    print(hopper is tapia)  # prints:
    tapia.replace('upcoming', 'graduating')  ###
    print(tapia)  # prints:
    print(hopper)  # prints:
    hopper = hopper.replace('upcoming', 'pre-alumni')  ###
    print(hopper)  # prints:
    print(tapia)  # prints:

    Explain what is happening on the statements marked ###.
Suppose we’re interested in building a dominoes game. Two operations we would need to perform are (1) the construction of the set of tiles and (2) the shuffling of the tiles themselves. Let’s assume that a single domino is represented as a pair of numbers between 0 and 6 where the first element of the pair is never greater than the second. For example, (0,6) is a valid domino. A set of dominoes is simply all unique combinations of pairs of values.

2a. Here is some code that is supposed to generate a set of dominoes. Unfortunately, it has some logical errors. Edit the code to remove the errors while maintaining the general spirit of the logic. Please do not use list comprehensions.

```python
def tiles(pips=6):
    "Returns a full set of domino tiles. Each tile is a unique combination of pips between 0 and 'pips'. ex: calling tiles(2) should return: [(0, 0), (0, 1), (1, 1)]"
    results = []
    for front in range(pips):
        for back in range(0, pips):
            tile = tuple(front, back)
            results += list(tile)
    return results
```

2b. The following code shuffles a list of n tiles by performing a series of \(\frac{n}{2}\) exchanges. Each exchange randomly selects two items of the list and then swaps the items. Again, the code is not quite right. Please make changes that reflect the desired logic without using comprehensions.

```python
def shuffle(pile):
    """Shuffles a list of items."""
    n = len(pile)
    for _ in range(0,n//2):
        left = randint(0,n-1)
        right = randint(left,n)
        temp = pile[left]
        pile[right] = pile[left]
        pile[left] = temp
    return pile
```

2c. Extra: How might you distinguish a syntax error from a runtime error from a logic error? What is a common example of a syntax error? Of a runtime error?