

## SI 206 Data-Oriented Programming

### Homework Name: Unit Testing

#### Homework Objective:

Demonstrate the ability to:

- Complete partial unit tests
- Create unique unit test

#### Deliverables and Submission Process:

Submit a modified version of the file `206_UnitTests.py` via Canvas. The code must be executable! Code that does not run will be given a score of 0. You can receive partial credit for working programs.

#### Supporting Material:

Files can be accessed via the course GitHub site: <https://github.com/cvanlent/SI206-Fall2017>

Files: `206_UnitTests.py`

#### Background:

In order to complete this assignment you will need to familiarize yourself with the Card class covered in lecture. You will also want to review any readings on Unit Testing in Python.

Then you will include tests for the cases described below. There are a few notes though:

- You may create as many or few `unittest.TestCase` subclasses as you like.
- You must arrange your code so that if you were to fail a test, that would not cause any other tests to fail as a result of that test failing.
- IF YOU INVOKE THE `play_war_game` FUNCTION IN A TEST CASE, YOU MUST INVOKE IT WITH THE PARAMETER `testing=True`, like this:  
    `play_war_game(testing=True)` AS OPPOSED TO `play_war_game()`  
If you do not do that, we will not be able to grade your homework properly!
- You may assume that other programmers will NOT invoke these functions with unacceptable inputs (e.g. no one will try to create a card with rank 0). You just need to ensure that the code works as intended.

#### Steps:

1. Test that if you create a card with rank 12, its rank will be "Queen"
2. Test that if you create a card with rank 1, its rank will be "Ace"

3. Test that if you create a card instance with rank 3, its rank will be 3
4. Test that if you create a card instance with suit 1, it will be suit "Clubs"
5. Test that if you create a card instance with suit 2, it will be suit "Hearts"
6. Test that if you create a card instance, it will have access to a variable `suit_names` that contains the list ["Diamonds", "Clubs", "Hearts", "Spades"]
7. Test that if you invoke the `__str__` method of a card instance that is created with `suit=2`, `rank=7`, it returns the string "7 of Hearts"
8. Test that if you create a deck instance, it will have 52 cards in its `cards` instance variable
9. Test that if you invoke the `pop_card` method on a deck, it will return a card instance.
10. Test that the return value of the `play_war_game` function is a tuple with three elements, the first of which is a string. (This will probably require multiple test methods!)
11. (and 12) Write at least 2 additional tests (not repeats of the above described tests). Make sure to include a descriptive message in these two so we can easily see what you are testing!

**Integrity Policy:**

All materials submitted by students must be their own work - you may not submit material from previous semesters or examples taken from class or the Internet. Students may discuss the homework with others, but should not share code. If you work with others, make sure to indicate their name and the nature of the collaboration. Any instances of cheating will receive a 0 on the assignment and one letter grade deduction in the final course grade. If you are unsure about the integrity of your submission, you have 48 hours after submission to withdraw your submission.

**No assignments are accepted more than 24 hours late.**