

Lab 8
CPE101 Winter 2019
Due: March 15th

Peer Review

In this lab you are going to review the work of at least three of your classmates. By doing so, you are going to learn to read programs somebody else wrote, to know what clean and efficient programs look like, to write clean and efficient code, and to write a test using unittest.

1. Write a test to test program files for project 3 using unittest module.
2. To use unittest, create a new file lab8_test.py
3. At the top of the file, add this line: import unittest
4. Add this line below the previous line: from calcudoku import *
5. At the bottom of the file, add the following lines:

```
def main():  
    # execute unit tests  
    unittest.main()
```

```
if __name__ == '__main__':  
    main()
```

6. Above the “if __name__ == '__main__':” line, add the following class with functions:

```
class MyTest(unittest.TestCase):
```

```
    def test_validate_all(self):  
        #write 3 tests to test the validate_all() function  
        # to test, use this builtin function: self.assertEqual(test_val, expected_val)  
        #, where test_val is the value returned from the function you are testing,  
        # and expected_val is the value you are expecting from the function.
```

```
    def test_validate_rows(self):  
        #write 3 tests to test the validate_rows() function  
        # to test, use this builtin function: self.assertEqual(test_val, expected_val)  
        #, where test_val is the value returned from the function you are testing,  
        # and expected_val is the value you are expecting from the function.
```

```
    def test_validate_cols(self):  
        #write 3 tests to test the validate_cols() function  
        # to test, use this builtin function: self.assertEqual(test_val, expected_val)  
        #, where test_val is the value returned from the function you are testing,  
        # and expected_val is the value you are expecting from the function.
```

```
def test_validate_cages(self):  
    #write 3 tests to test the transpose_string() function  
    # to test, use this builtin function: self.assertEqual(test_val, expected_val)  
    #, where test_val is the value returned from the function you are testing,  
    # and expected_val is the value you are expecting from the function.
```

7. In addition to the unit tests described above, run the program with the supplied test inputs for the project 3 and see if the outputs from the program is correct.
8. Write a review noting if the tests described above fail or succeed.
9. Rate the program based on the test results with a score between 0 and 4.
10. Rate the readability (clean and concise program should receive a higher rating) of the program with a score between 0 and 4.
11. Rate the efficiency of the program with a score between 0 and 4.
12. Submit your test file and the reviews to polylearn as a zipped file.
13. When you are ready to do this lab, ask your instructor to provide you program files to be reviewed.