

# Integration Testing

---

ERIN KEITH

# Goals

---

1. Unit Test Reflection
2. Introduce integration testing
3. Explore methods for approaching integration testing

# Unit Test Reflection







---

- OOP
  - state
  - behavior
  - relationships
- TicTacToe

# Integration Tests

Test the combination of units

- The goal is to expose faults in the interactions between Units and Systems

	Unit Testing	Integration Testing
		
		

# Integration Test Example

---

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced and tested separately. When two or more units are ready, they are assembled, and Integration Testing is performed. For example, whether the cap fits into the body or not.

# Interactions to Test

---

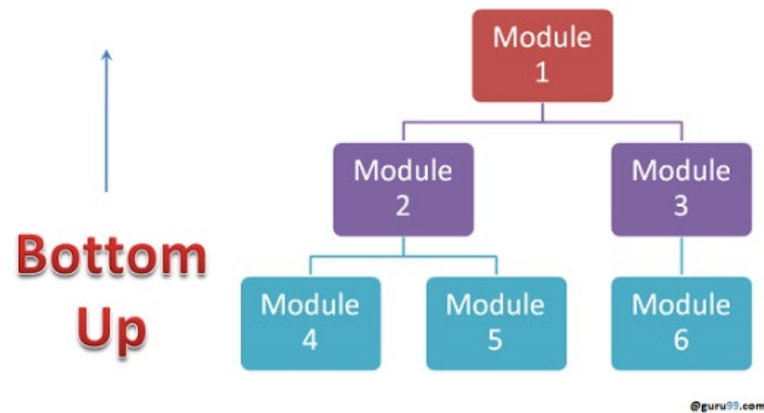
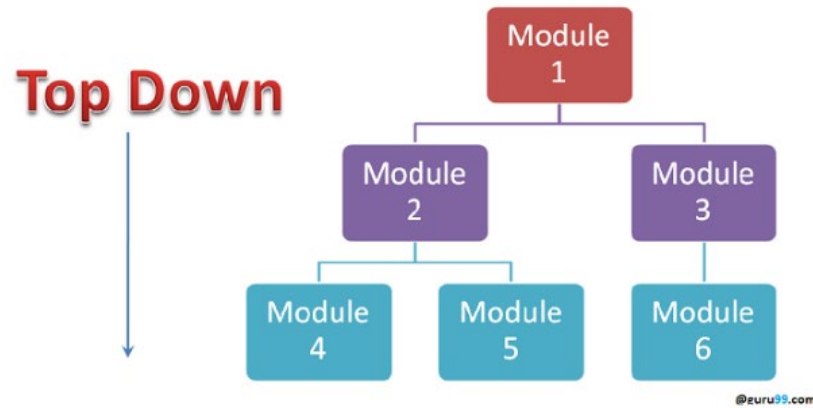
- Composition
- Dependency Injection
  - behavior in one class has side effects on another class
  - state in one class affects behavior in another class
- Exception Handling

If these components have outside dependencies (such as IO), you can “mock” them

- next week!

# Strategies

---



# Project Organization

Because there is a dependency on the units, unit tests should exist and run successfully before integration testing.

In teams or organizations, it can sometimes be unclear whose responsibility this is.





# Implementation

---

In many testing libraries (including unittests in Python), integration testing is an abstraction. Integration tests are implemented using the same techniques as unit tests, but their purpose is different.

```
class
TestBoardAndPlayer(unittest.TestCase):

    def test_player_X_move_row5_col5(self):
        pass
```

# Implementation

---

Because interactions are more complex and often multiple components are involved, there can be multiple assert statements.

- check the board state
- check the player state

```
class
```

```
TestBoardAndPlayer(unittest.TestCase) :
```

```
    def test_player_X_move_row5_col5(self) :  
        pass
```