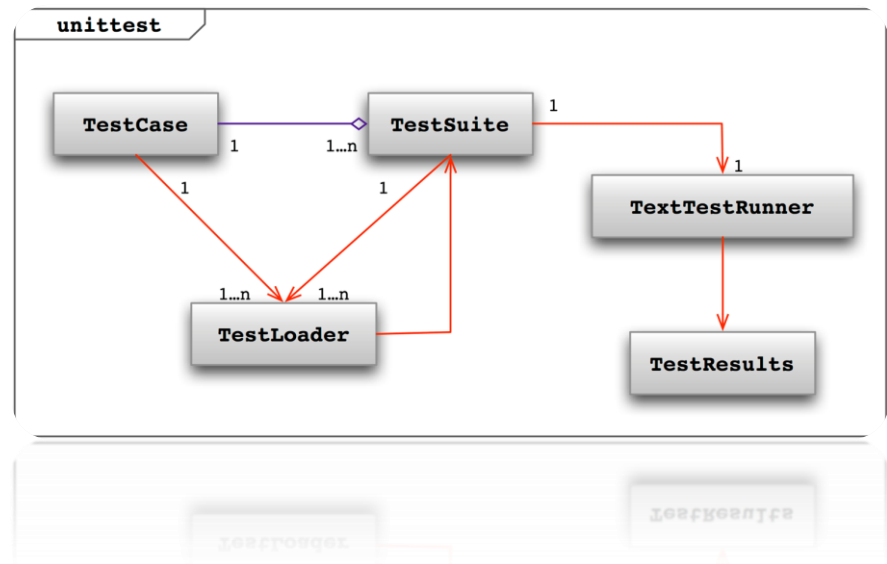


# Python Training

A basic overview

# Unit Testing

- The unittest module was earlier a third party module called “PyUnit” and later became default module in Python.
- 5 key classes as shown in fig.
  - TestCase
  - TestSuite
  - TestLoader
  - TextTestRunner
  - TestResults
- unittest.TestCase methods
  - setUp(): runs before every test
  - tearDown(): runs after every test
  - skipTest(msg:string):
  - fail(msg:string):
  - id(): returns a string containing the name of the TestCase object and of the test routine
  - shortDescription(): returns the docstr comment



# Unit Testing

- Designing a test routine
  - Each test routine must have the prefix "test" in its name.
  - To perform a test, the test routine should use an assert method.
- Basic boolean asserts

Assert	Complement Assert	Operation
<code>assertTrue(a, M)</code>	<code>assertFalse(a, M)</code>	<code>a = True; a = False</code>
<code>assertEqual(a, b, M)</code>	<code>assertNotEqual(a, b, M)</code>	<code>a = b; a ≠ b</code>
<code>assertIs(a, b, M)</code>	<code>assertIsNot(a, b, M)</code>	<code>a is b; a is not b</code>
<code>assertIsNone(a, M)</code>	<code>assertIsNotNone(a, M)</code>	<code>a = nil; a ≠ nil</code>
<code>assertIsInstance(a, b, M)</code>	<code>assertIsNotInstance(a, b, M)</code>	<code>isinstance(a,b); not isinstance(a,b)</code>

- Creating test suite
  - `unittest.TestLoader().loadTestsFromTestCase(TestCase1)`

# Unit Testing

## ■ Running the Tests

### ■ Two ways to run the tests

- `unittest.main`
- `unittest.TextTestRunner().run`

### ■ Regardless of approach, test cases and their routines run in alphanumeric order

### ■ Skipping a test is achieved using

- `unittest.skip()` method placed before the test routine with `@` token
  - `skipIf()` and `skipUnless()` conditional skip
- `skipTest()` method of `TestCase` class

## ■ Viewing the Test Results

- `unittest.TextTestRunner(stream=sys.stderr, descriptions=True, verbosity=1)`
- `TestResult` object

# Python 2 vs Python 3

Python 2	Python 3
<code>print x</code>	<code>print(x)</code>
<code>4/3 = 1</code>	<code>4/3 = 1.33333</code> <code>4//3 = 1</code>
<code>raw_input()</code>	<code>input()</code>
<code>file("my_file.txt")</code>	<code>open("my_file.txt")</code>
<code>xrange()</code>	<code>range()</code>
<code>except ExceptionType , e</code>	<code>except ExceptionType as e</code>
List pop function removes elements from end only	List pop function can remove elements at any index