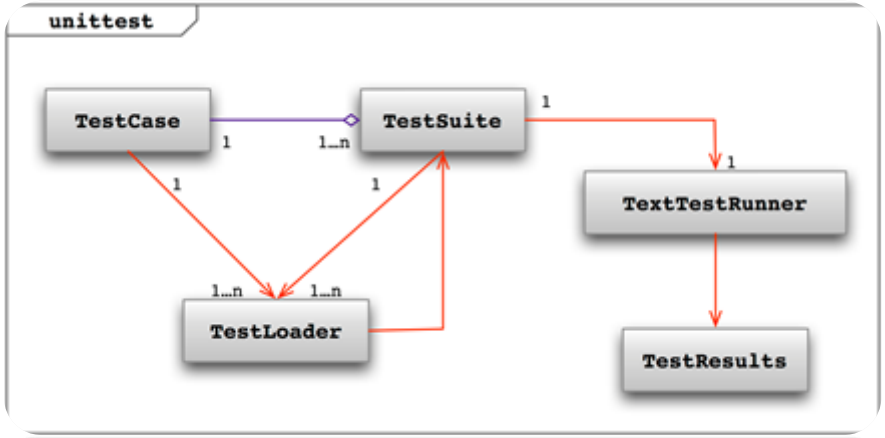
Python Training

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
assertTrue(a, M)	assertFalse(a, M)	a = True; a = False
assertEqual(a, b, M)	assertNotEqual(a, b, M)	a = b; a ≠ b
assertIs(a, b, M)	assertIsNot(a, b, M)	a is b; a is not b
assertIsNone(a, M)	assertIsNotNone(a, M)	a = nil; a ≠ nil
assertIsInstance(a, b, M)	assertIsNotInstance(a, b, M)	isinstance(a,b); not isinstance(a,b)

- **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

- Code written in python 3 is not backward compatible.
- Most of the current Linux distributions and Macs still use python 2.x as default.
- Some of the 2.x modules are still not compatible with 3.x
- Some comparisons:

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

Case Study: Frequently used modules [Self Study]

- **sys**
  - E.g. sys.stdout.write() : print something without formatting. Useful if newline is not needed to be printed at the end, change path of modules etc.
- **time**
  - E.g. Access current time, clock ticks between events etc.
- **os**
  - E.g. Unix like system calls e.g. create directory, processes, change file permissions etc.