

Instructor Notes:

A presentation slide with a blue header and a white footer. The header contains the text "Python Training" in a large, bold, white font, and "Lesson 05: Unit Testing" in a smaller, white font below it. The footer contains the Capgemini logo and tagline "People matter, results count." on the left, and the Capgemini logo and tagline "CONSULTING. TECHNOLOGY. OUTSOURCING" on the right. At the bottom right, there is a small copyright notice: "Copyright © Capgemini 2015. All Rights Reserved" and a page number "1".

Python Training
Lesson 05: Unit Testing

People matter, results count.
 **Capgemini**
CONSULTING. TECHNOLOGY. OUTSOURCING

 **Capgemini**
CONSULTING. TECHNOLOGY. OUTSOURCING

Copyright © Capgemini 2015. All Rights Reserved 1

Instructor Notes:

Lesson Objectives

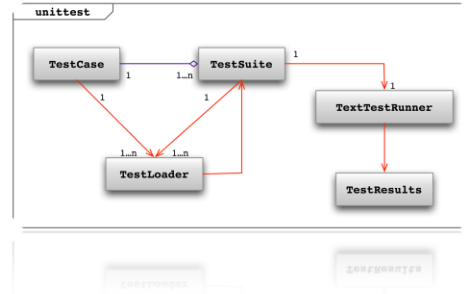
- After completing this lesson, you will learn about:
 - Unit Test Classes
 - Test Suite
 - Python 2 vs 3



Instructor Notes:

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



Instructor Notes:

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)`

Instructor Notes:

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

Instructor Notes:

Python 2 vs Python 3

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

`a=int(input())` is possible in python 3 but not with `raw_input()` in Python 2

Instructor Notes:

Summary

- In this lesson, you learnt:

- Unit Test Classes
- Test Suite
- Python 2 vs 3

