



```
import unittest as ut

# Check for the Valid Mail Id
def chkMailId(mail_id):
    status = ""

    if (len(mail_id) > 0):
        if (mail_id.count('@') == 1):
            status = "Valid Mail Id"
        else:
            status = "Invalid Mail Id"
    else:
        status = "Mail Id is not Passed"

    return status

# Addition of 2 Nos
add_num = lambda num1, num2: num1 + num2

# Create the Class for Unit Test Cases
class ut1(ut.TestCase):
    # This method will be executed before each and every test cases
    def setUp(self):
        print ("Executing ->", self.shortDescription())

    # This method will be executed after each and every test cases
    def tearDown(self):
        print (self.shortDescription(), "has been executed successfully")

    def testAdd_1(self):
        'Add_1'
        self.assertEqual(add_num(5,3), 8)

    def testAdd_2(self):
        'Add_2'
        self.assertNotEqual(add_num(5,3), -8)

    def testCheckMail_1(self):
        'CheckMail_1'
        self.assertEqual(chkMailId(''), "Mail Id is not Passed")

    def testCheckMail_2(self):
        'CheckMail_2'
        self.assertEqual(chkMailId('mohan@gmail.com'), 'Valid Mail Id')

    @ut.skip("Skipping the Test Cases")
    def testCheckMail_3(self):
        'CheckMail_3'
        self.assertEqual(chkMailId('mohangmail.com'), 'Invalid Mail Id')
```



```
# Create the Test Suite
def create_suite():
    suite = ut.TestSuite()

    # Add all the test cases
    # suite.addTest(ut.makeSuite(ut1))

    # Add only the specific test cases to the suite
    suite.addTest(ut1('testAdd_1'))
    suite.addTest(ut1('testCheckMail_1'))
    suite.addTest(ut1('testCheckMail_3'))

    return(suite)

if (__name__ == '__main__'):
    # ut.main()

    # Running the Test Suite
    runner = ut.TextTestRunner()
    test_suite = create_suite()
    runner.run(test_suite)
```