

**Name**: Abdul Ahad Butt

**Roll**: 231-485930

**Course**: Software Quality Assurance

**Section**: A

**Submitted to**: Saad Bin Saleem

**Automated Test Cases Using Unit Test**

**1)**

**Test**

**Positive Case:**

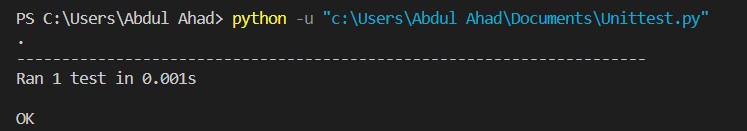
class Test(unittest.TestCase):

# Returns True if the string contains 4 '1'.

def test\_string(self):

self.assertEqual( '1'\*4, '1111')

**Output:**



**False Case:**

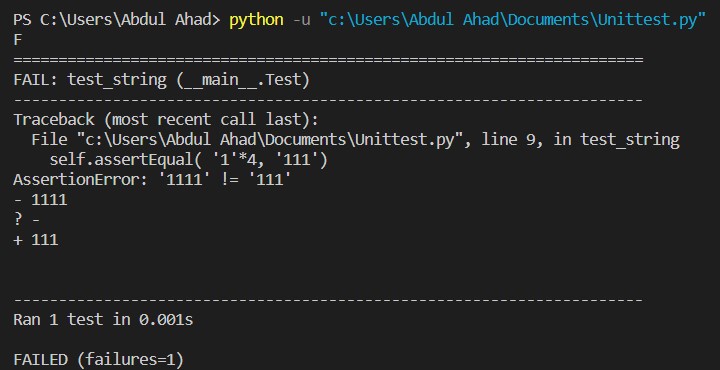
class Test(unittest.TestCase):

# Returns True if the string contains 4 '1'.

def test\_string(self):

self.assertEqual( '1'\*4, '111')

**Output:**



**2)**

**Test1**

**Positive Case:**

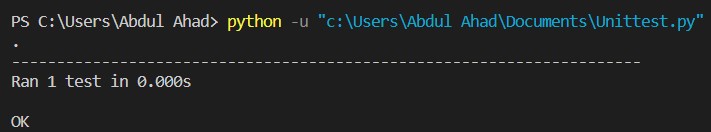
class Test1(unittest.TestCase):

# Returns True if the string is in upper case.

def test\_upper(self):

self.assertEqual('hopper'.upper(), 'HOPPER')

**Output:**

****

**False Case:**

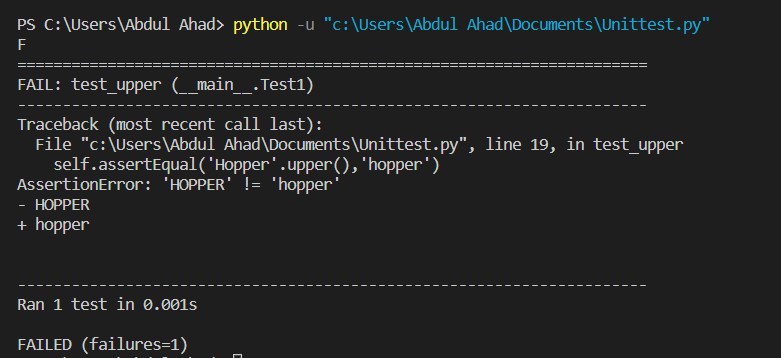
class Test1(unittest.TestCase):

# Returns True if the string is in upper case.

def test\_upper(self):

self.assertEqual('Hopper'.upper(),'hopper')

**Output:**

****

**3)**

**Test2**

**Positive Case:**

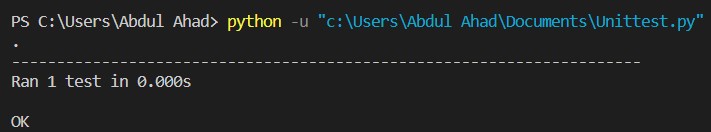
class Test2(unittest.TestCase):

# Returns True if the string is in lower case.

def test\_lower(self):

self.assertEqual('ALIO'.lower(), 'alio')

**Output:**

****

**False Case:**

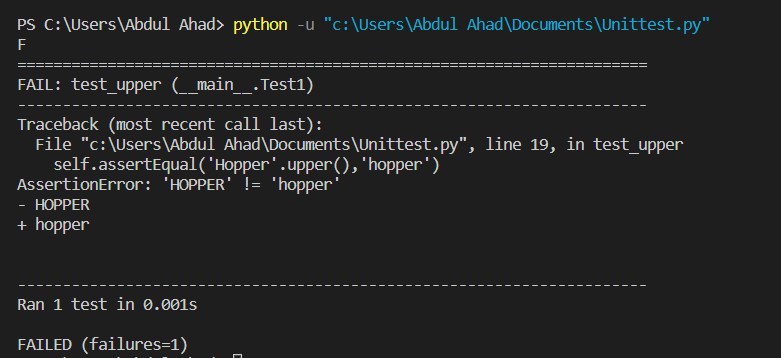
class Test2(unittest.TestCase):

# Returns True if the string is in lower case.

def test\_lower(self):

self.assertEqual('ten'.lower(),'TEN')

**Output:**

****

**4)**

**Test3**

**Positive Case:**

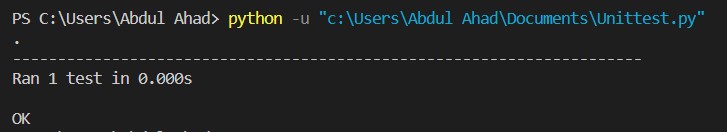
class Test3(unittest.TestCase):

# Returns True if the Square of Number 4 is 16

def test\_square(self):

self.assertEqual(4\*4,16)

**Output:**

****

**False Case:**

class Test3(unittest.TestCase):

# Returns True if the Square of Number 4 is 16

def test\_square(self):

self.assertEqual(4\*4,24)

**Output:**

Text

Description automatically generated

**5)**

**Positive Case:**

class Test4(unittest.TestCase):

# Returns true if the string splits and matches

# the given output.

def test\_split(self):

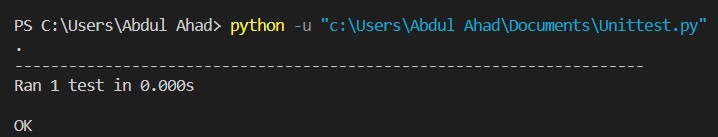
s = 'Quality Assurance'

self.assertEqual(s.split(), ['Quality', 'Assurance'])

with self.assertRaises(TypeError):

s.split(2)

**Output:**

****

**False Case:**

class Test4(unittest.TestCase):

# Returns true if the string splits and matches

# the given output.

def test\_split(self):

s = 'Quality Assurance'

##self.assertEqual(s.split(), ['Quality', 'Measure'])

with self.assertRaises(TypeError):

s.split(2)

**Output:**

**Text

Description automatically generated**

**6)**

**Positive Case:**

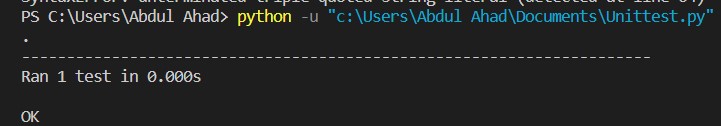
class Test5(unittest.TestCase):

# Return True if Greatest Common Divisor of 5 and 10 is 5

def test\_gcd(self):

self.assertEqual(5%10,5)

**Output:**

****

**False Case:**

class Test5(unittest.TestCase):

# Return True if Greatest Common Divisor of 5 and 10 is 5

def test\_gcd(self):

self.assertEqual(5%10,4)

**Output:**

**Text

Description automatically generated**