Unit Testing 1 Solutions

1.

```python
def add_two_numbers(a, b):
  return a + b

def test_adds_two_whole_numbers():
  expected = 2
  actual = add_two_numbers(1, 1)
  assert expected == actual

test_adds_two_whole_numbers()
```

1.

```python
def add_two_numbers(a, b):
  return a + b

def test_adds_pos_neg_numbers():
  expected = 0
  actual = add_two_numbers(-100, 100)
  assert expected == actual

test_adds_pos_neg_numbers()
```

1.

```python
def add_two_numbers(a, b):
  return a + b

def test_adds_two_fp_numbers():
  expected = 0.9
  actual = add_two_numbers(0.5, 0.4)
  assert expected == actual

test_adds_two_fp_numbers()
```

1.

```python
def add_two_numbers(a, b):
  try:
    return a + b
  except:
    return "Invalid Input"
```

```python
def test_adds_string_to_number():
  expected = "Invalid Input"
  actual = add_two_numbers("test", 1)
  assert expected == actual

test_adds_string_to_number()
```

1.

```python
def add_two_numbers(a, b):
  if not instanceof(a, int) or not instanceof(b, int):
    return "Invalid input"

  return a + b

def test_adds_two_strings():
  expected = "Invalid Input"
  actual = add_two_numbers("test", "case")
  assert expected == actual

test_adds_two_strings()
```