

Homework: Lesson 9 – Unit Testing

Question 1

Simple ATM program unit tests – Example Solution

Note that students can implement this task in any way they want, however, they must include at five unit tests in total covering various scenarios / inputs.

```
from unittest import TestCase, mock
from . import homework
```

```
# These are only a handful of tests for the ATM functions.
# (to be used as examples / for guidance)
```

```
class TestVerifyPin(TestCase):
```

```
    def test_good_pin(self):
        self.assertTrue(homework.verify_pin(pin='1234'))

    def test_bad_pin(self):
        self.assertFalse(homework.verify_pin(pin='1111'))
```

```
class TestLogIn(TestCase):
```

```
    def test_bad_login(self):
        with mock.patch('builtins.input', side_effect=['111', '222', '333', '444']):
            result = homework.log_in()
            self.assertFalse(result)

    def test_good_login(self):
        with mock.patch('builtins.input', side_effect=['1234']):
            result = homework.log_in()
            self.assertTrue(result)
```

```
class TestRunAtm(TestCase):
```

```
    @mock.patch.object(homework, 'log_in')
    @mock.patch.object(homework, 'log_in')
    def test_run_atm_success_withdrawal(self, mock_verify_pin, mock_log_in):
        mock_verify_pin.return_value = True
        mock_log_in.return_value = True
        result = homework.run_atm(withdrawal_amount=80)
        self.assertEqual(result, 20)

    @mock.patch.object(homework, 'log_in')
    @mock.patch.object(homework, 'log_in')
    def test_run_atm_raises_error(self, mock_verify_pin, mock_log_in):
        mock_verify_pin.return_value = True
        mock_log_in.return_value = True
        with self.assertRaises(ValueError):
            homework.run_atm(withdrawal_amount=500)
```