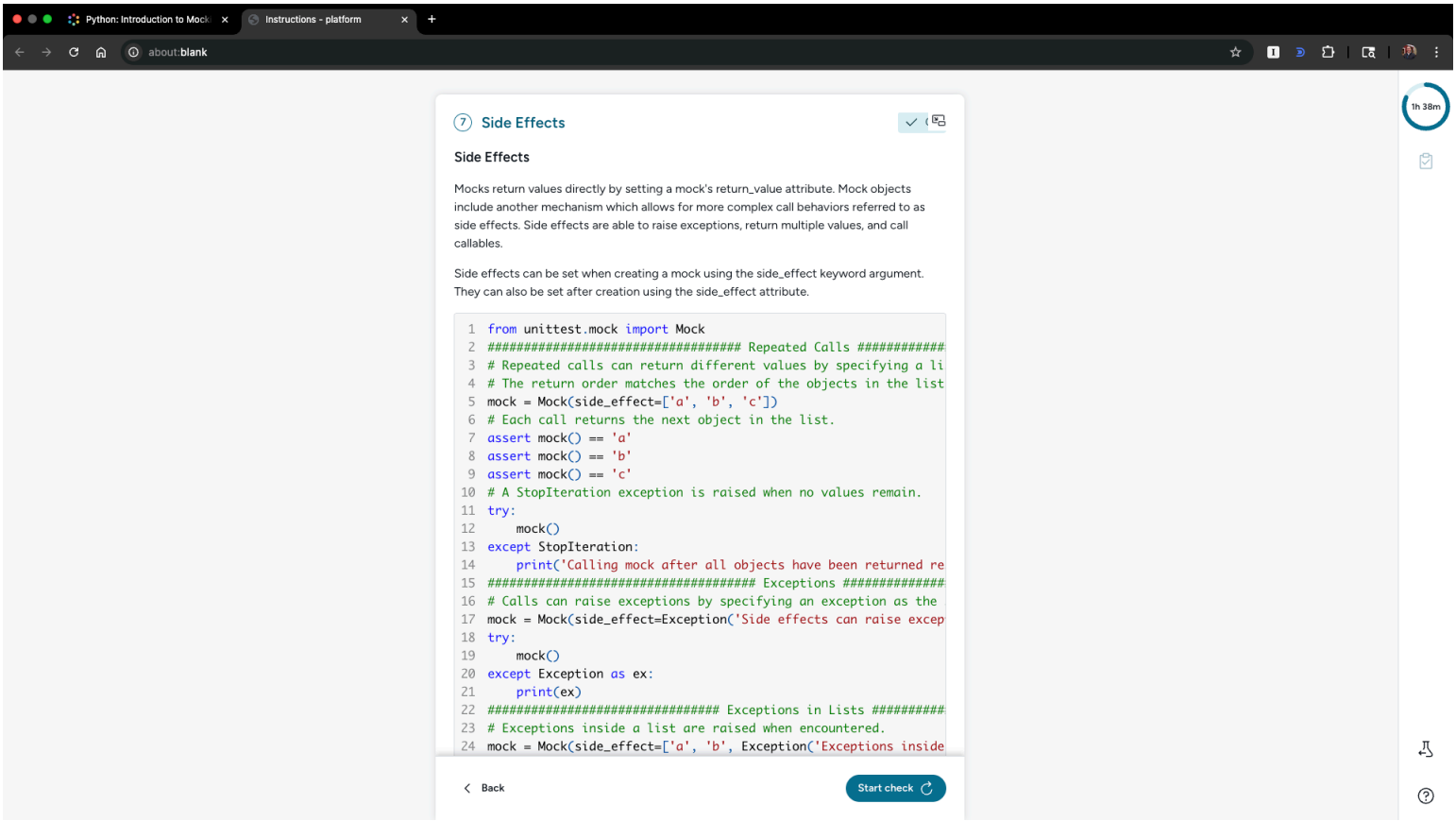
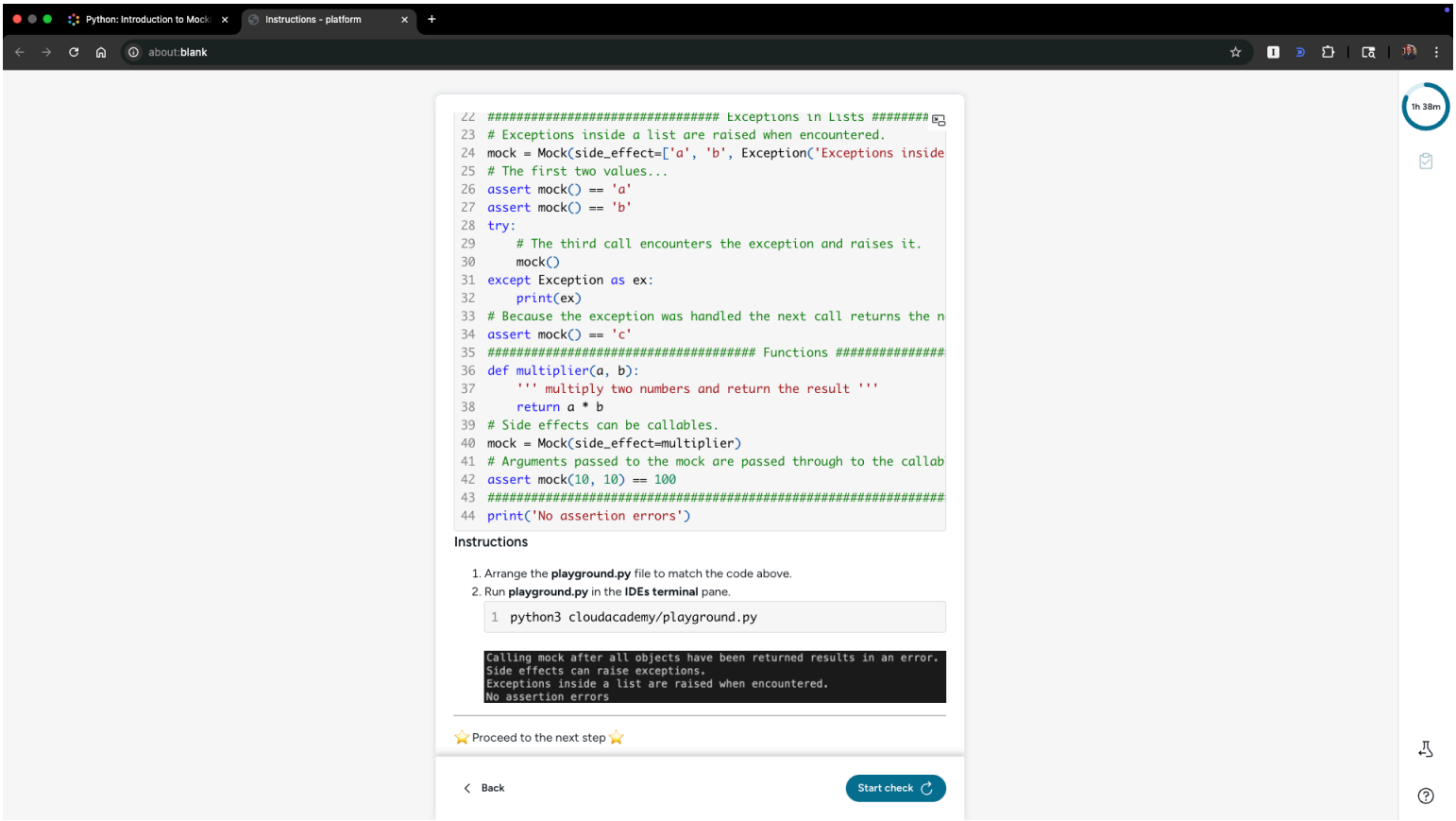

Python: Introduction to Mock
Instructions - platform
about:blank
7 Side Effects
Side Effects
Mocks return values directly by setting a mock's return_value attribute. Mock objects include another mechanism which allows for more complex call behaviors referred to as side effects. Side effects are able to raise exceptions, return multiple values, and call callables.
Side effects can be set when creating a mock using the side_effect keyword argument. They can also be set after creation using the side_effect attribute.
from unittest.mock import Mock
################################### Repeated Calls ############
# Repeated calls can return different values by specifying a li
# The return order matches the order of the objects in the list
mock = Mock(side_effect=['a', 'b', 'c'])
# Each call returns the next object in the list.
assert mock() == 'a'
assert mock() == 'b'
assert mock() == 'c'
# A StopIteration exception is raised when no values remain.
try:
mock()
except StopIteration:
print('Calling mock after all objects have been returned re
################################### Exceptions ##############
# Calls can raise exceptions by specifying an exception as the
mock = Mock(side_effect=Exception('Side effects can raise excep
try:
mock()
except Exception as ex:
print(ex)
############################### Exceptions in Lists ##########
# Exceptions inside a list are raised when encountered.
mock = Mock(side_effect=['a', 'b', Exception('Exceptions inside
Back
Start check
1h 38m


Python: Introduction to Mock
Instructions - platform
about:blank
1h 38m
22 ################################ Exceptions in Lists ########
23 # Exceptions inside a list are raised when encountered.
24 mock = Mock(side_effect=['a', 'b', Exception('Exceptions inside
25 # The first two values...
26 assert mock() == 'a'
27 assert mock() == 'b'
28 try:
29 # The third call encounters the exception and raises it.
30 mock()
31 except Exception as ex:
32 print(ex)
33 # Because the exception was handled the next call returns the n
34 assert mock() == 'c'
35 ################################### Functions ###############
36 def multiplier(a, b):
37 ''' multiply two numbers and return the result '''
38 return a * b
39 # Side effects can be callables.
40 mock = Mock(side_effect=multiplier)
41 # Arguments passed to the mock are passed through to the callab
42 assert mock(10, 10) == 100
43 ###################################################################
44 print('No assertion errors')
Instructions
1. Arrange the playground.py file to match the code above.
2. Run playground.py in the IDEs terminal pane.
1 python3 cloudacademy/playground.py
Calling mock after all objects have been returned results in an error.
Side effects can raise exceptions.
Exceptions inside a list are raised when encountered.
No assertion errors
Proceed to the next step
Back
Start check