NOVEMBER 30, 2024

ADVANCE PYTHON

MODULE 7.2: TEST CASES

STEVE STYLIN
BELELVUE UNIVERSITY

Code Explanation

city_functions.py

1. **Function Definition**:
   - The function is defined using the def keyword, followed by the function name and its parameters.
   - A docstring is included to describe the function's purpose.
2. **Return Statement**:
   - The function returns a formatted string that combines the city and country, separated by a comma.
3. **Function Calls**:
   - The function is called three times with different city and country pairs: "Santiago, Chile", "Paris, France", and "Montreal, Canada".
   - Each call to the function is printed to the console, allowing us to see the output directly.

```python
# city_functions.py


def city_country(city, country):
    """Return a formatted string of the form 'City, Country'."""
    return f"{city}, {country}"

# Calling the function with different city and country values
print(city_country("Santiago", "Chile"))
print(city_country("Paris", "France"))
print(city_country("Montreal", "Canada"))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe c:/csd/csd-325/module-7/city_functions.py
Santiago, Chile
Paris, France
Montreal, Canada
```

test_cities.py

1. **Importing Modules**: We import the unittest module for testing and the city_country function from the city_functions module.
2. **Test Class**: We define a class TestCityCountry that inherits from unittest.TestCase. This class contains our test methods.
3. **Test Method**: The method test_city_country() checks if the city_country function returns the expected formatted strings for the given inputs. We use assertEqual to compare the output of the function with the expected result.

4. **Running Tests**: The if __name__ == '__main__': block ensures that the tests run when the script is executed directly.

```
Welcome    city_functions.py    test_cities.py ×
test_cities.py > ...
3
4   class TestCityCountry(unittest.TestCase):
5       def test_city_country(self):
6           """Test the city_country function with various inputs."""
7           self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chil
8           self.assertEqual(city_country("Paris", "France"), "Paris, France")
9           self.assertEqual(city_country("Montreal", "Canada"), "Montreal, Can
10
11  if __name__ == '__main__':
12      unittest.main()
13
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   SQL CONSOLE   Python
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe
d/csd-325/module-7/city_functions.py
Santiago, Chile
Paris, France
Montreal, Canada
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe
d/csd-325/module-7/test_cities.py
Santiago, Chile
Paris, France
Montreal, Canada
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```

Modify city_functions.py

The function *city_functions* is modified to accept a third parameter, population, and the return string is updated accordingly. This change allows us to provide more detailed information about the city.

Welcome | city_functions.py | test_cities.py

city_functions.py > ...

```python
# city_functions.py



def city_country(city, country,population):
    """Return a formatted string of the form 'City, Country, population'."""
    return f"{city}, {country},{population}"

# Calling the function with different city and country values
print(city_country("Santiago", "Chile", 5000000))
print(city_country("Paris", "France", 2148000))
print(city_country("Montreal", "Canada", 1780000))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.ex
d/csd-325/module-7/city_functions.py
Santiago, Chile,5000000
Paris, France,2148000
Montreal, Canada,1780000
```

We rerun the test_cities.py. If the test does not match the expected output, it will fail, demonstrating how to handle changes in function behavior.

Welcome | city_functions.py | test_cities.py

test_cities.py > ...

```python
import unittest
from city_functions import city_country

class TestCityCountry(unittest.TestCase):
    def test_city_country(self):
        """Test the city_country function with various inputs."""
        self.assertEqual(city_country("Santiago", "Chile"), "Santi
        self.assertEqual(city_country("Paris", "France"), "Paris,
        self.assertEqual(city_country("Montreal", "Canada"), "Mont

if __name__ == '__main__':
    unittest.main()
```

PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SQL CONSOLE | Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Pytho
d/csd-325/module-7/test_cities.py
Santiago, Chile,5000000
Paris, France,2148000
Montreal, Canada,1780000
E
======================================================================
ERROR: test_city_country (__main__.TestCityCountry.test_city_country)
Test the city_country function with various inputs.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:\csd\csd-325\module-7\test_cities.py", line 7, in test_city_country
    self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chile")
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: city_country() missing 1 required positional argument: 'population'

----------------------------------------------------------------------
Ran 1 test in 0.000s

FAILED (errors=1)
PS C:\csd\csd-325\module-7>
```

city_functions.py

1. **Function Definition**: The city_country function takes three parameters: city, country, and an optional population.
2. **String Formatting**: It checks if the population parameter is provided. If it is, the function returns a string formatted as "City, Country - population xxx". If not, it returns "City, Country".

city_functions.py > ...

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases

# city_functions.py



def city_country(city, country, population=None):
    """Return a formatted string of the form 'City, Country - population xx
    if population:
        return f"{city}, {country} - population {population}"
    else:
        return f"{city}, {country}"

# Function calls
print(city_country("Santiago", "Chile"))
print(city_country("Paris", "France"))
print(city_country("Montreal", "Canada"))

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python
d/csd-325/module-7/city_functions.py
Santiago, Chile
Paris, France
Montreal, Canada
```

test_cities.py

1. **Unit Testing**: This file uses the unittest framework to test the city_country function.
2. **Test Method**: The test_city_country method verifies that the function returns the expected strings for various inputs, including the population.

Modifications

1. **Adding Population**: The function was initially modified to include a population parameter, which caused the tests to fail if it was not provided.
2. **Making Population Optional**: The population parameter was then made optional, allowing the function to pass all tests again.

test_cities.py > ...

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases
# test_cities.py

import unittest
from city_functions import city_country

class TestCityCountry(unittest.TestCase):
    def test_city_country(self):
        """Test the city_country function with various inputs."""
        self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chil
        self.assertEqual(city_country("Paris", "France"), "Paris, France")
        self.assertEqual(city_country("Montreal", "Canada"), "Montreal, Can

if __name__ == '__main__':
    unittest.main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python
d/csd-325/module-7/test_cities.py
Santiago, Chile
Paris, France
Montreal, Canada
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```

city_functions.py

We modified the city_country function and added a new parameter called language. This allows us to specify the language spoken in the respective city. The function now constructs a formatted string that includes the city name, country name, population (formatted with commas for readability), and language.

city_functions.py > ...

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases

# city_functions.py



def city_country(city, country, population, language):
    """Return a formatted string of city, country, population, and language
    return f"{city}, {country} - population {population:,}, {language}"

# Example usage
if __name__ == "__main__":
    print(city_country("Santiago", "Chile", 5000000, "Spanish"))
    print(city_country("Paris", "France", 2148000, "French"))
    print(city_country("Montreal", "Canada", 1780000, "English/French"))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe
d/csd-325/module-7/city_functions.py
Santiago, Chile - population 5,000,000, Spanish
Paris, France - population 2,148,000, French
Montreal, Canada - population 1,780,000, English/French
```

test_cities.py

When you run the test_cities.py file after making these changes, it is expected to fail. This is likely because the test cases in test_cities.py may not account for the new language parameter.

test_cities.py > TestCityCountry > test_city_country

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases
# test_cities.py

import unittest
from city_functions import city_country

class TestCityCountry(unittest.TestCase):
    def test_city_country(self):
        """Test the city_country function with various inputs."""
        self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chil
        self.assertEqual(city_country("Paris", "France"), "Paris, France")
        self.assertEqual(city_country("Montreal", "Canada"), "Montreal, Can

if __name__ == '__main__':
    unittest.main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe c
d/csd-325/module-7/test_cities.py
E
======================================================================
ERROR: test_city_country (__main__.TestCityCountry.test_city_country)
Test the city_country function with various inputs.
----------------------------------------------------------------------
Traceback (most recent call last):
  File "c:\csd\csd-325\module-7\test_cities.py", line 12, in test_city_country
    self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chile")
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: city_country() missing 2 required positional arguments: 'population' and 'language'

----------------------------------------------------------------------
Ran 1 test in 0.000s

FAILED (errors=1)
```

city_functions.py

The city_country function takes four parameters: city, country, population, and language. The last two parameters are optional.

city_functions.py > city_country

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases

# city_functions.py


def city_country(city, country, population=None, language=None):
    """Return a formatted string of the form 'City, Country - population xx
    if population and language:
        return f"{city}, {country} - population {population}, {language}"
    elif population:
        return f"{city}, {country} - population {population}"
    elif language:
        return f"{city}, {country}, {language}"
    else:
        return f"{city}, {country}"

# Function calls
print(city_country("Santiago", "Chile"))
print(city_country("Paris", "France", 2148000))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe
d/csd-325/module-7/city_functions.py
Santiago, Chile
Paris, France - population 2148000
Montreal, Canada - population 1780000, French
```

Initially, the function was modified to include a population parameter, which caused the tests to fail. After making population optional, the tests passed again. The function was

then updated to include a language parameter, which also caused the tests to fail. After making language optional, the tests passed successfully.

test_cities.py > ...

```python

# Steve Stylin@Bellevue University
#Module 7.2 Test Cases
# test_cities.py

import unittest
from city_functions import city_country

class TestCityCountry(unittest.TestCase):
    def test_city_country(self):
        """Test the city_country function with various inputs."""
        self.assertEqual(city_country("Santiago", "Chile"), "Santiago, Chil
        self.assertEqual(city_country("Paris", "France"), "Paris, France")
        self.assertEqual(city_country("Montreal", "Canada"), "Montreal, Can

if __name__ == '__main__':
    unittest.main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE Python

```
PS C:\csd\csd-325\module-7> & C:/Users/Steve/AppData/Local/Programs/Python/Python312/python.exe
d/csd-325/module-7/test_cities.py
Santiago, Chile
Paris, France - population 2148000
Montreal, Canada - population 1780000, French
.
----------------------------------------------------------------------
Ran 1 test in 0.000s

OK
```