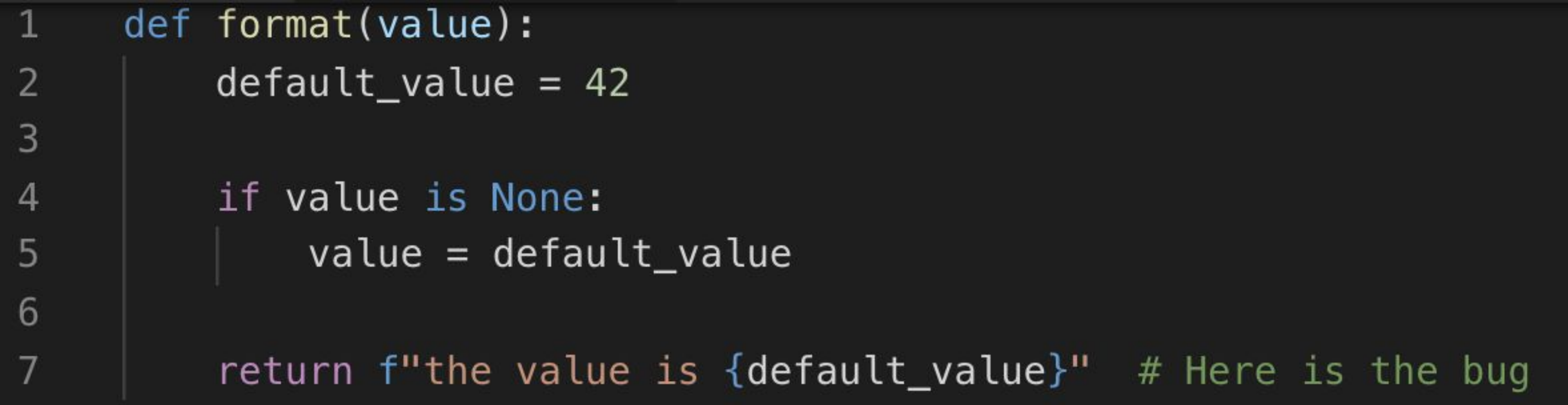
Coverage Criteria

Kai Böhrnsen - Boran Sahindal

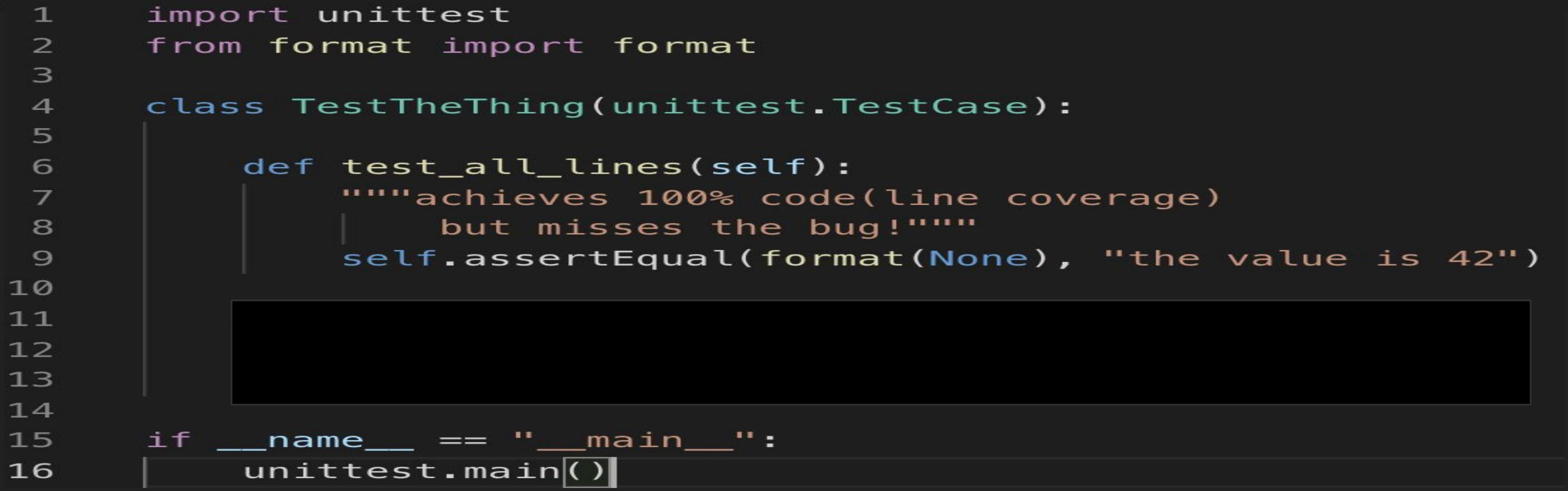
Code coverage vs Functional coverage

- Code coverage
  - How well your code has been exercised by your test bench?
  - Collected automatically.
  - Easy to measure.

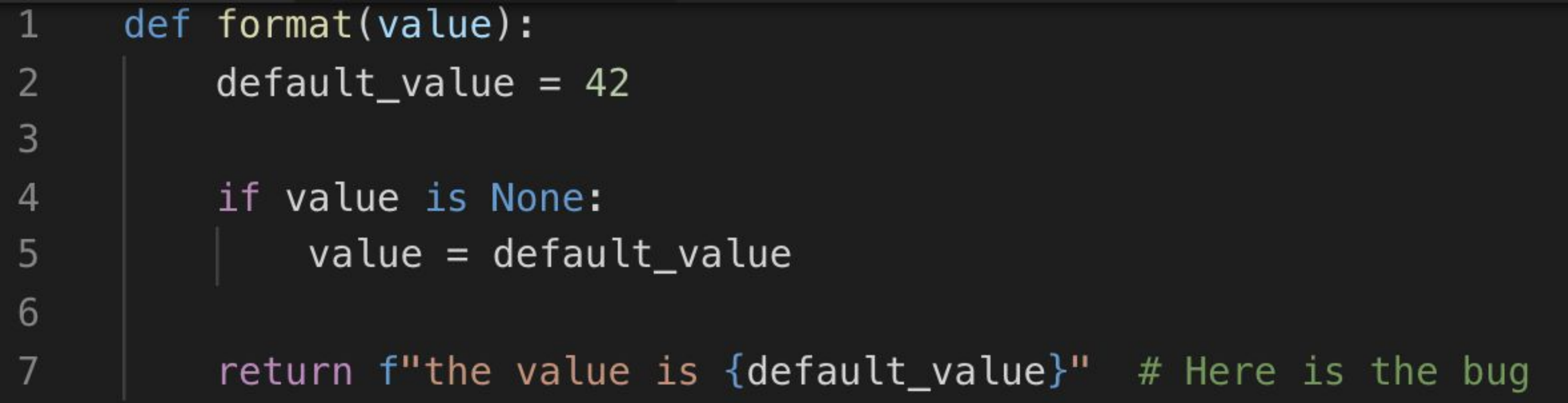
- Functional coverage
  - How well the functionality of the design has been covered by your test bench?
  - User defined functionality.
  - Hard to measure



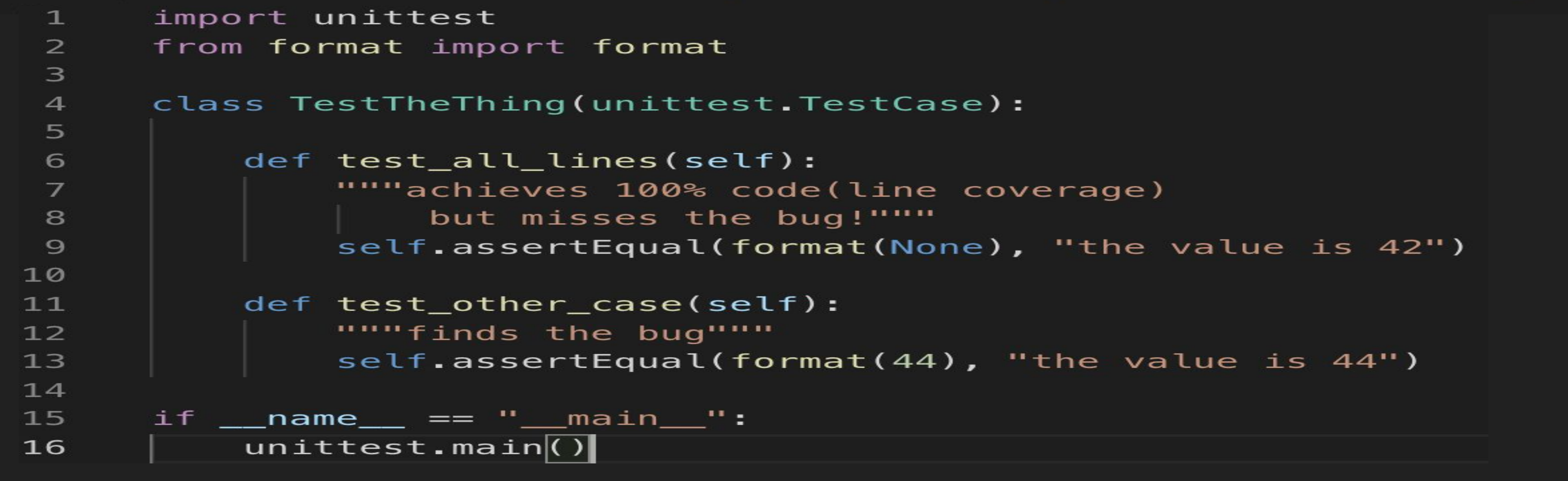
```python
def format(value):
    default_value = 42

    if value is None:
        value = default_value

    return f"the value is {default_value}"  # Here is the bug
```



```python
import unittest
from format import format

class TestTheThing(unittest.TestCase):

    def test_all_lines(self):
        """achieves 100% code(line coverage)
            but misses the bug!"""
        self.assertEqual(format(None), "the value is 42")

    [redacted]

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



```python
def format(value):
    default_value = 42

    if value is None:
        value = default_value

    return f"the value is {default_value}"  # Here is the bug
```



```python
import unittest
from format import format

class TestTheThing(unittest.TestCase):

    def test_all_lines(self):
        """achieves 100% code(line coverage)
            but misses the bug!"""
        self.assertEqual(format(None), "the value is 42")

    def test_other_case(self):
        """finds the bug"""
        self.assertEqual(format(44), "the value is 44")

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

Function coverage

- All accessible functions, methods and subroutines are covered by tests
- No agreement among developers if private methods should be included
    - Testing private methods is not possible in all programming language
- Good start for testing

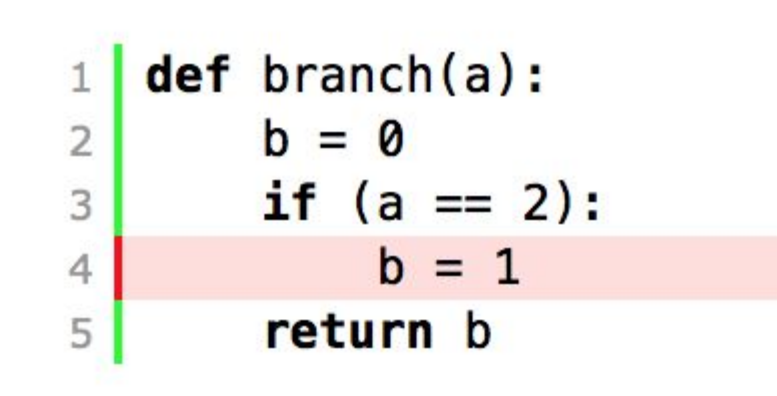
Statement coverage

- Line coverage
- White-box testing

Detects:

- Unused statements
- Dead code
- Unused branches

Statement coverage example



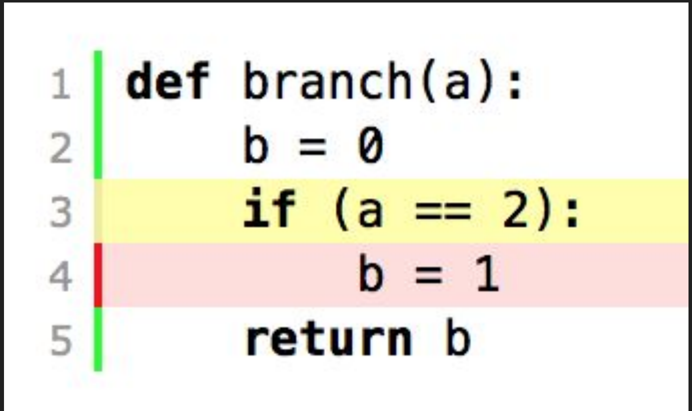
```
def branch(a):
    b = 0
    if (a == 2):
        b = 1
    return b
```

- 80% coverage
- Possible dead code, unused branch

Branch coverage

- Coverage of decision outcomes taken
- Find unused code segments
- Find independent fragments for refactoring

Branch coverage example



```
1 def branch(a):
2     b = 0
3     if (a == 2):
4         b = 1
5     return b
```

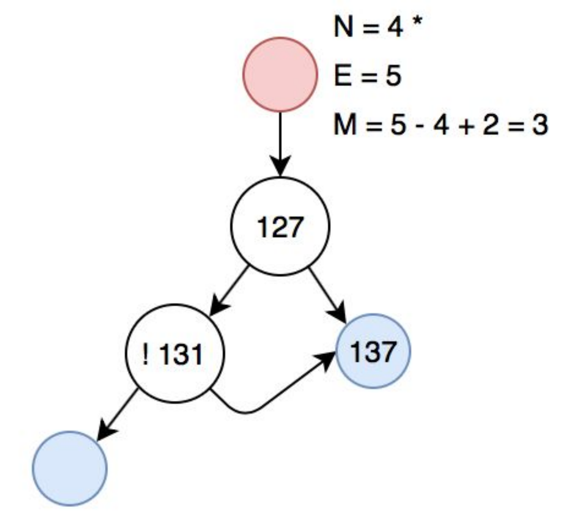
- Partial coverage, line 3
- Two branches, one taken (50%)

Common tools

- Codecov.io
- Coveralls.io
- Codeclimate.com
- JaCoCo
- Cobertura
- Clover
- GNU Gcov
- And many more...

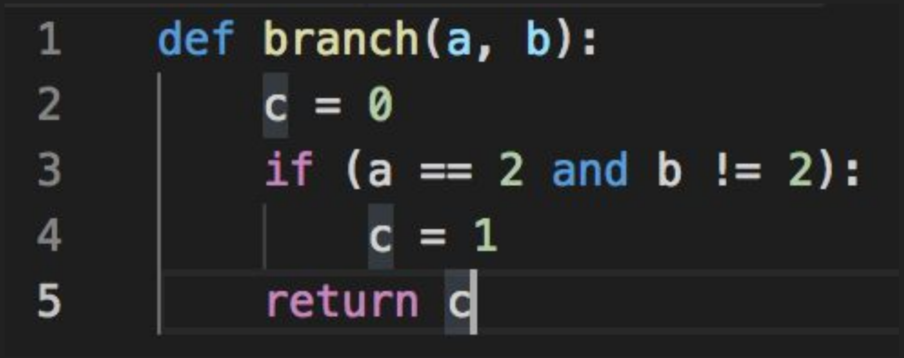
Edge coverage

- Has every edge in Control flow graph being executed



Condition coverage

- Reveal how the variables and subexpressions are evaluated

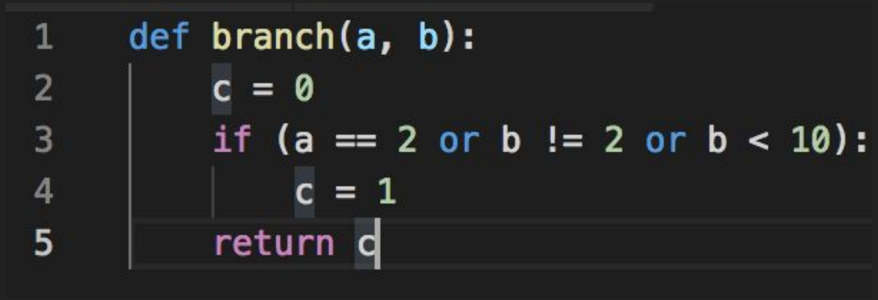


```python
def branch(a, b):
    c = 0
    if (a == 2 and b != 2):
        c = 1
    return c
```

a == 2	b != 2
T	T
T	F
F	T
F	F

Clause coverage

- Checks that each condition in a statement is reachable and evaluates its outcomes



```python
def branch(a, b):
    c = 0
    if (a == 2 or b != 2 or b < 10):
        c = 1
    return c
```

- 6 cases for 100%:
  - T/F, doesn’t matter, doesn’t matter
  - F, T/F, doesn’t matter
  - F, F, T/F