Aharouni Refaël **Explanation Code Final Exam Software Metrics**

st v2425-25

In this document, I will explain my function and the modifications.

- **Improvement of the Software metrics of the file with the code itself:**

At the beginning, I only wrote my function, so I had this:

```
1    def wrapper_round(x):
2        return round(x, 1)
```

When I first used Pylint, I had this:

```
PS C:\Users\rahar> .\py313\Scripts\activate
(py313) PS C:\Users\rahar> pylint "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"
************* Module aharouni_refael_software_metrics_final_exam
Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py:2:0: C0304: Final newline missing (missing-final-newline)
Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py:1:0: C0114: Missing module docstring (missing-module-docstring)
Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py:1:0: C0116: Missing function or method docstring (missing-function-docs
tring)

-----------------------------------
Your code has been rated at 0.00/10
```

The first thing I did was add a line at the end of the document, which erased the first error saying that the final line was missing:

```
1    def wrapper_round(x):
2        return round(x, 1)
3
```

The next thing I did was add a module docstring at the beginning of my code, which gave me this code:

```
1    """This function uses round to test some cases"""
2
3    def wrapper_round(x):
4        return round(x, 1)
5
```

and this Software Metrics:

```
(py313) PS C:\Users\rahar> pylint "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"
************* Module aharouni_refael_software_metrics_final_exam
Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py:3:0: C0116: Missing function or method docstring (missing-function-docs
tring)

--------------------------------------------------------------------
Your code has been rated at 5.00/10 (previous run: 0.00/10, +5.00)
```

After that, I understood that I also had to add a function docstring, just like this:

```
1    """This function uses round to test some cases"""
2
3    def wrapper_round(x):
4        """This function uses round to test some cases"""
5        return round(x, 1)
6
```

This gave me a software Metrics of 10/10:

```
(py313) PS C:\Users\rahar> pylint "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 5.00/10, +5.00)
```

However, in this function, I did not have all the cases (like the case where x is an integer or x is not a float), that is why I added these conditions in my function. The final function is:

```python
    """This function uses round to test some cases"""


    def wrapper_round(x):
        """This function uses round to test some cases"""
        if not isinstance(x, (int, float)):
            raise ValueError("Decimal number required")
        if isinstance(x, int):
            x = float(x)
        if x.is_integer():
            return int(x)
        return round(x, 1)

```

This finally gave me this Software Metrics:

```
(py313) PS C:\Users\rahar> pylint "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

At this moment, I began using Flake8 to improve my code. I only had this problem:

```
(py313) PS C:\Users\rahar> flake8 "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"
C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py:3:1: E302 expected 2 blank lines, found 1
```

This is why I added another line between the class docstring and the definition of the function.

This finally gave me this:

```
(py313) PS C:\Users\rahar> flake8 "C:\Users\rahar\Desktop\Test_exam\aharouni_refael_software_metrics_final_exam.py"
```

- **<u>Improvement of the Software metrics of the file with the tests:</u>**

At the beginning, I only had the code with all the tests, just like this:

```python
import unittest
from aharouni_refael_software_metrics_final_exam import wrapper_round

class TestRoundNumber(unittest.TestCase):

    def test_positive_number(self):
        self.assertEqual(wrapper_round(2.34567), 2.3)

    def test_zero(self):
        self.assertEqual(wrapper_round(0), 0.0)

    def test_negative_number(self):
        self.assertEqual(wrapper_round(-18.96), -19)

    def test_edge_upper_positive_case(self):
        self.assertEqual(wrapper_round(1.05), 1.1)

    def test_edge_lower_positive_case(self):
        self.assertEqual(wrapper_round(1.04), 1.0)

    def test_edge_upper_negative_case(self):
        self.assertEqual(wrapper_round(-4.05), -4.1)

    def test_edge_upper_negative_case(self):
        self.assertEqual(wrapper_round(-12.14), -12.1)

    def test_integer_positive_case(self):
        self.assertEqual(wrapper_round(5.0), 5)

    def test_integer_positive_case(self):
        self.assertEqual(wrapper_round(-5.0), -5)

    def test_extreme_case(self):
        self.assertEqual(wrapper_round(1.000001), 1.0)

    def test_one_decimal(self):
        self.assertEqual(wrapper_round(1.5), 1.5)

    def test_not_a_decimal_number(self):
        with self.assertRaises(ValueError):
            wrapper_round("shjkf")

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

I had many problems in Pylint:

```
Desktop\Test_exam\test_wrapper_round.py:4:0: C0115: Missing class docstring (missing-class-docstring)
Desktop\Test_exam\test_wrapper_round.py:6:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:9:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:12:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:15:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:18:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:21:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:24:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:24:4: E0102: method already defined line 21 (function-redefined)
Desktop\Test_exam\test_wrapper_round.py:27:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:30:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:30:4: E0102: method already defined line 27 (function-redefined)
Desktop\Test_exam\test_wrapper_round.py:27:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:30:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:30:4: E0102: method already defined line 27 (function-redefined)
Desktop\Test_exam\test_wrapper_round.py:30:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:30:4: E0102: method already defined line 27 (function-redefined)
Desktop\Test_exam\test_wrapper_round.py:33:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:36:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:39:4: C0116: Missing function or method docstring (missing-function-docstring)

-----------------------------------
Desktop\Test_exam\test_wrapper_round.py:33:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:36:4: C0116: Missing function or method docstring (missing-function-docstring)
Desktop\Test_exam\test_wrapper_round.py:39:4: C0116: Missing function or method docstring (missing-function-docstring)

-----------------------------------
-----------------------------------
Your code has been rated at 1.00/10
```

I first erased all the trailing whitespaces and then added docstring for the class, the module and the functions, which finally gave me a Software Metrics of 10/10:

```
(py313) PS C:\Users\rahar> pylint "C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py"

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 9.67/10, +0.33)
```

After that, I used Flake8 to also improve my code:

```
(py313) PS C:\Users\rahar> flake8 "C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py"
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:6:1: E302 expected 2 blank lines, found 1
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:22:80: E501 line too long (91 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:26:80: E501 line too long (91 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:30:80: E501 line too long (91 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:34:80: E501 line too long (91 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:38:80: E501 line too long (88 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:42:80: E501 line too long (88 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:46:80: E501 line too long (86 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:50:80: E501 line too long (93 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:54:80: E501 line too long (80 > 79 characters)
C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py:58:1: E305 expected 2 blank lines after class or function definition, found 1
```

I first reduced some lines that were too long, added another blank lines between the import and the definition of the class and after the class. At the end, it gave me this for flake8:

```
(py313) PS C:\Users\rahar> flake8 "C:\Users\rahar\Desktop\Test_exam\test_wrapper_round.py"
(py313) PS C:\Users\rahar>
```

And my code for the tests was:

```python
"""This module allows to do unitary tests"""

import unittest
from aharouni_refael_software_metrics_final_exam import wrapper_round


class TestRoundNumber(unittest.TestCase):
    """This class contains the unitary tests for wrapper_round"""

    def test_positive_number(self):
        """This function tests the case where x is a positive number"""
        self.assertEqual(wrapper_round(2.34567), 2.3)

    def test_zero(self):
        """This function tests the case where x is zero"""
        self.assertEqual(wrapper_round(0), 0.0)

    def test_negative_number(self):
        """This function tests the case where x is a negative number"""
        self.assertEqual(wrapper_round(-18.96), -19)

    def test_edge_upper_positive_case(self):
        """This function tests an edge case (x positive)"""
        self.assertEqual(wrapper_round(1.05), 1.1)

    def test_edge_lower_positive_case(self):
        """This function tests an edge case (x positive)"""
        self.assertEqual(wrapper_round(1.04), 1.0)

    def test_edge_upper_negative_case(self):
        """This function tests an edge case (x negative)"""
        self.assertEqual(wrapper_round(-4.05), -4.1)

    def test_edge_lower_negative_case(self):
        """This function tests an edge case (x negative)"""
        self.assertEqual(wrapper_round(-12.14), -12.1)

    def test_integer_positive_case(self):
        """This function tests the case: x is in |N"""
        self.assertEqual(wrapper_round(5.0), 5)

    def test_integer_negative_case(self):
        """This function tests the case: x is in |Z"""
        self.assertEqual(wrapper_round(-5.0), -5)

    def test_extreme_case(self):
        """This function tests a case where x tends to zero"""
        self.assertEqual(wrapper_round(1.000001), 1.0)

    def test_one_decimal(self):
        """This function tests the case with one number in the decimal part"""
        self.assertEqual(wrapper_round(1.5), 1.5)

    def test_not_a_decimal_number(self):
        """This function tests the case where x is a string"""
        with self.assertRaises(ValueError):
            wrapper_round("shjkf")


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

```