Code Understanding Report

Generated: 2025-05-06 16:09:06

This report presents automated insights based on large language models and code analysis tools.

File: pasted_code.py

Summary

- class MyMeta(type): def **new**(cls, name, bases, attrs): attrs['custom_attribute'] = 'Added by class' return super().**new**(cls, name, bases, attrs)

Description:

def metaclass*attribute*is_added(meta, attrname): return 'Custom Attribute Added by metaclass' in str(meta)

class MyMeta(type): def **new**(cls, name, bases, attrs): attrs['custom_attribute'] = 'Added by class' return super - :

Your MyMeta class

class MyMeta: def **init**(self): self.name = 'Walder'

Actions:

def do_something(): print('Do something')

def do_something2(): print('Do something2')

def do_something3(): print('Do something3')

def do_something4(): print('Do something4')

Constraints:

class MyClass(metaclass=MyMeta): #...

class MyClass2(metaclass=

Docstring

- : class MyMeta(type): def **new**(cls, name, bases, attrs): attrs['custom_attribute'] = 'Added by metaclass' return super().**new**(cls, name,
- ###

class MyMeta(object): def **init**(self, name): self.name = name

class MyClass(object): def **init**(self, metaclass=My

Code Quality

Tool: `pylint`
Issues: 0`

```text
************* Module tmpo8dmgux0 C:
\Users\nmoha\AppData\Local\Temp\tmpo8dmgux0.py:10:10: E1101:
Class 'MyClass' has no 'custom_attribute' member (no-member)
```

Conclusion

def add*a*new_metaclass(cls): #...

class MyMeta(type): def **new**(cls, name, bases, attrs): attrs['my_attribute'] = 'My value' return super().**new**(cls, name, bases, attrs)

Your MyMeta class

class MyMeta(object): def **init**(self, name): self.name = name

class MyClass2(object): def **init**(self, metaclass=MyMeta): #...

def add*a*new_metaclass2(cls): #...

def add*a*new_object(