Code Understanding Report

Generated: 2025-05-06 16:00:01

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

File: pasted_code.py

Summary

- :

log_calls

Refactored Solution

class Solution(object): def **init**(self): pass

```
def is_prime(self, nums):
    """
    :type nums: List[int]
    :rtype: bool
    """
    if not nums:
        return True
    if len(nums) < 2:
        return False

    for i in range(2, len(nums)):
        if nums[i] % nums[i - 1] == 0:
            return False
```

- ###

def subtract(a, b): return a - b

This is a stricter version of the code.

You may find another difference of the two code with different cases in each file.

def add(a, b): return a + b

add(1, 2) add(1, 2)

Docstring

- :

def log_calls(func): def wrapper(*args, *kwargs): print(f"Calling {func.**name**} with arguments: {args}, {kwargs}") result = func(*args, - ###

def add(a, b): return a + b

def add(a, b): return a + b

def add(a, b): return a + b

def add(

Code Quality

Tool: `pylint`
Issues: 2`

```text [AST Parse Error] expected an indented block after function definition on line 1 (line 2)

---

[AST Parse Error] expected an indented block after function definition on line 1 (line 2) ```

# Conclusion

Add( add(1,2) add(1,2), add(1,2) add(1,2) ) # this is the same as add() but this time it should only be used for # adding # arguments add(1,2), add(1,2), add(1,2), add(1,2), # again this time it should only be used for # adding arguments add(1,2), add(1,2), add(1,2), # add(1,2)
```