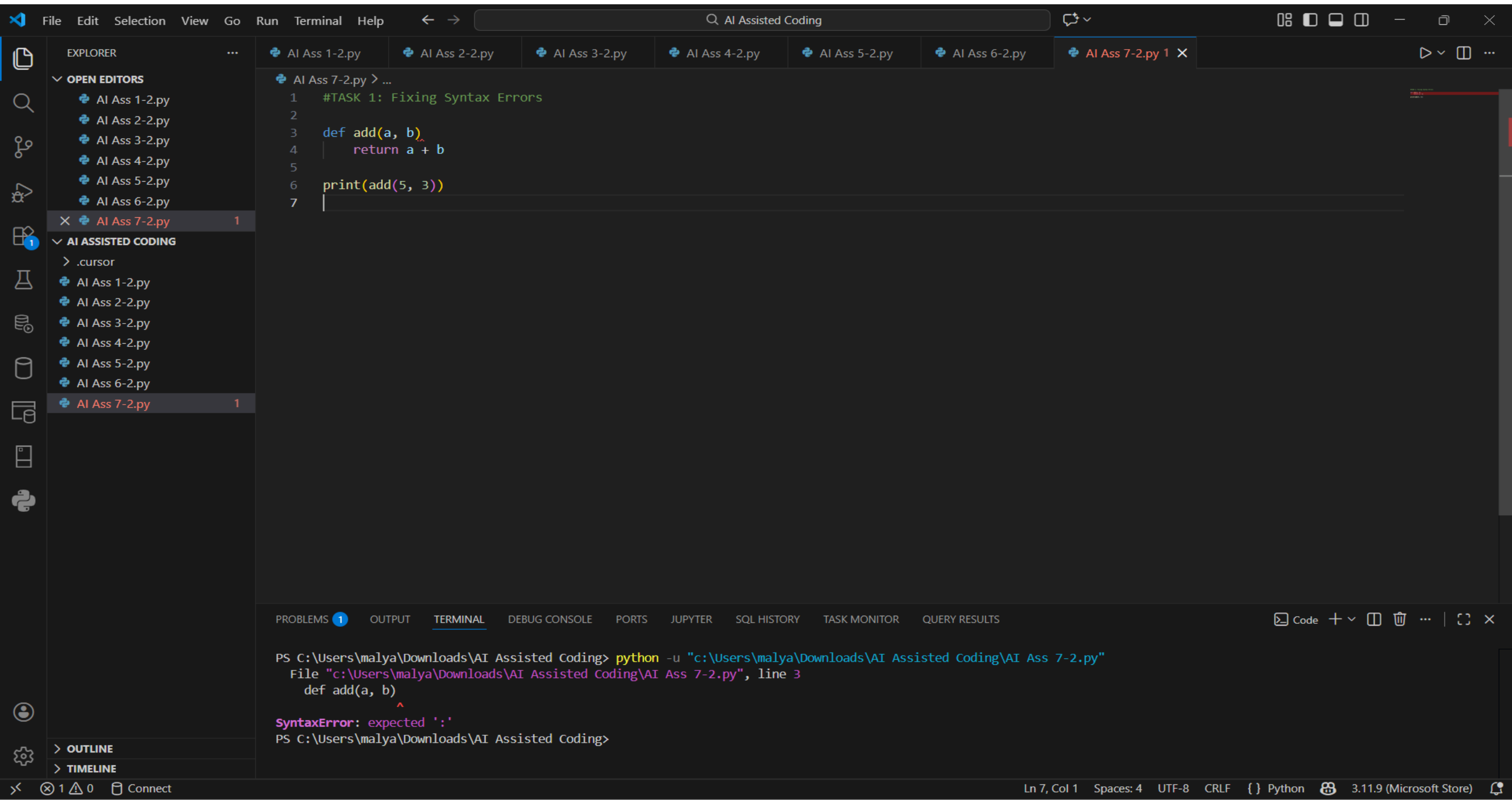
AI Assisted Coding ASSIGNMENT – 7

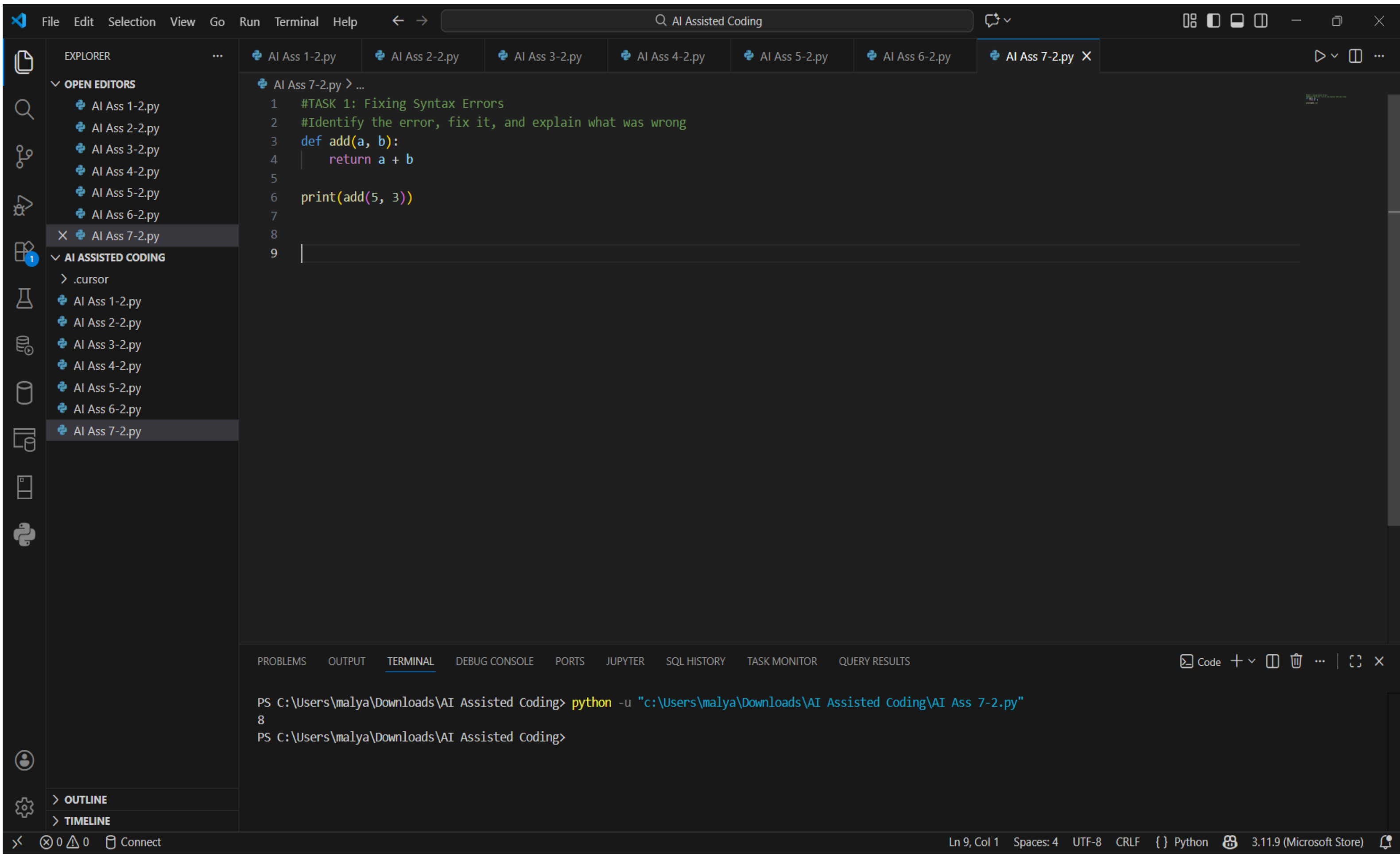
Roll.no : 2303a52443

M .Siddhartha
B-38


File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI Ass 1-2.py
AI Ass 2-2.py
AI Ass 3-2.py
AI Ass 4-2.py
AI Ass 5-2.py
AI Ass 6-2.py
AI Ass 7-2.py
AI ASSISTED CODING
.cursor
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 7, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

```python
#TASK 1: Fixing Syntax Errors

def add(a, b)
    return a + b

print(add(5, 3))
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
  File "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py", line 3
    def add(a, b)
                 ^
SyntaxError: expected ':'
PS C:\Users\malya\Downloads\AI Assisted Coding>
```


File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI Ass 1-2.py
AI Ass 2-2.py
AI Ass 3-2.py
AI Ass 4-2.py
AI Ass 5-2.py
AI Ass 6-2.py
AI Ass 7-2.py
AI ASSISTED CODING
.cursor
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 9, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

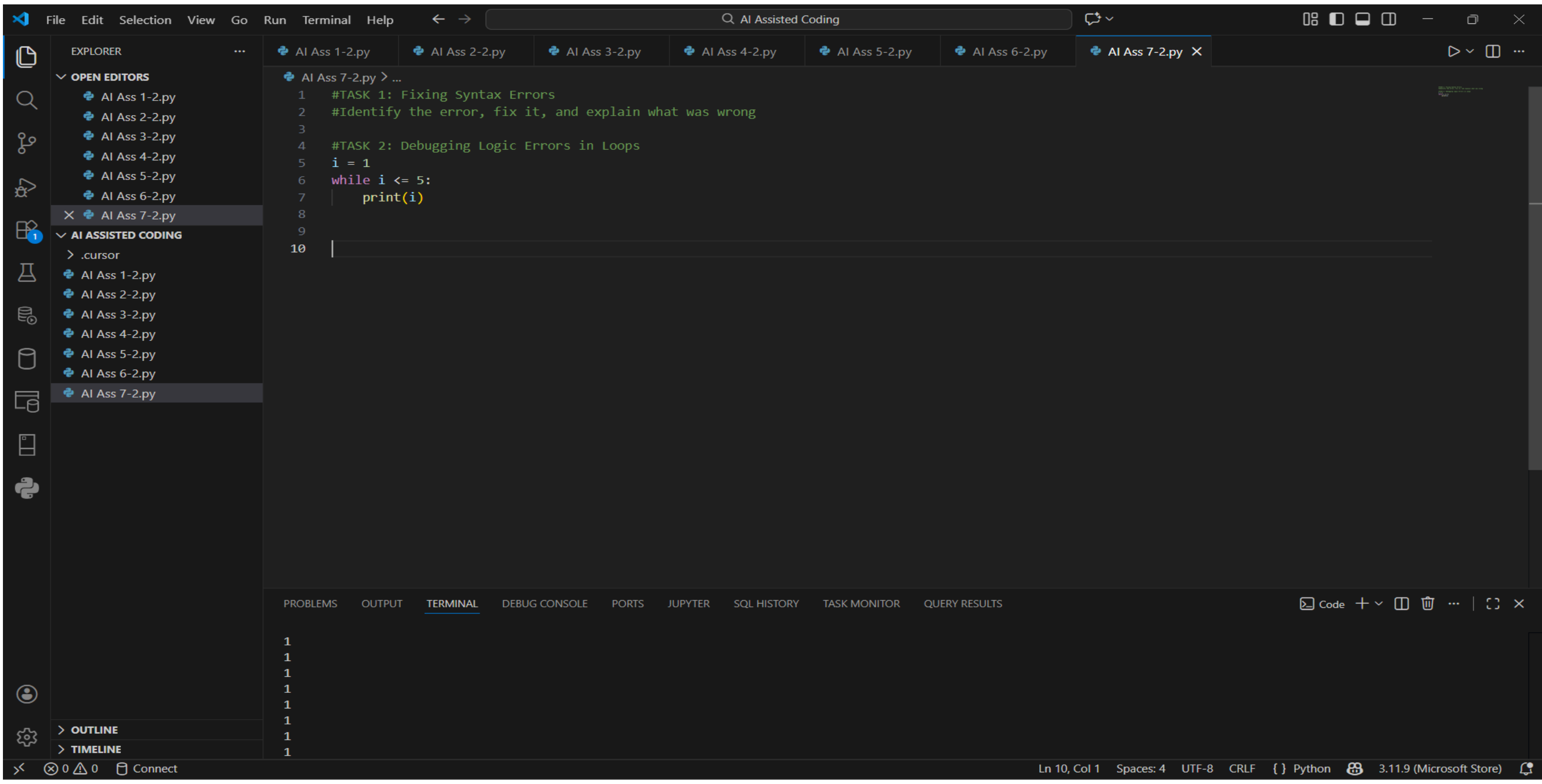
```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong
def add(a, b):
    return a + b

print(add(5, 3))
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
8
PS C:\Users\malya\Downloads\AI Assisted Coding>
```

Task 1: Fixing Syntax Errors

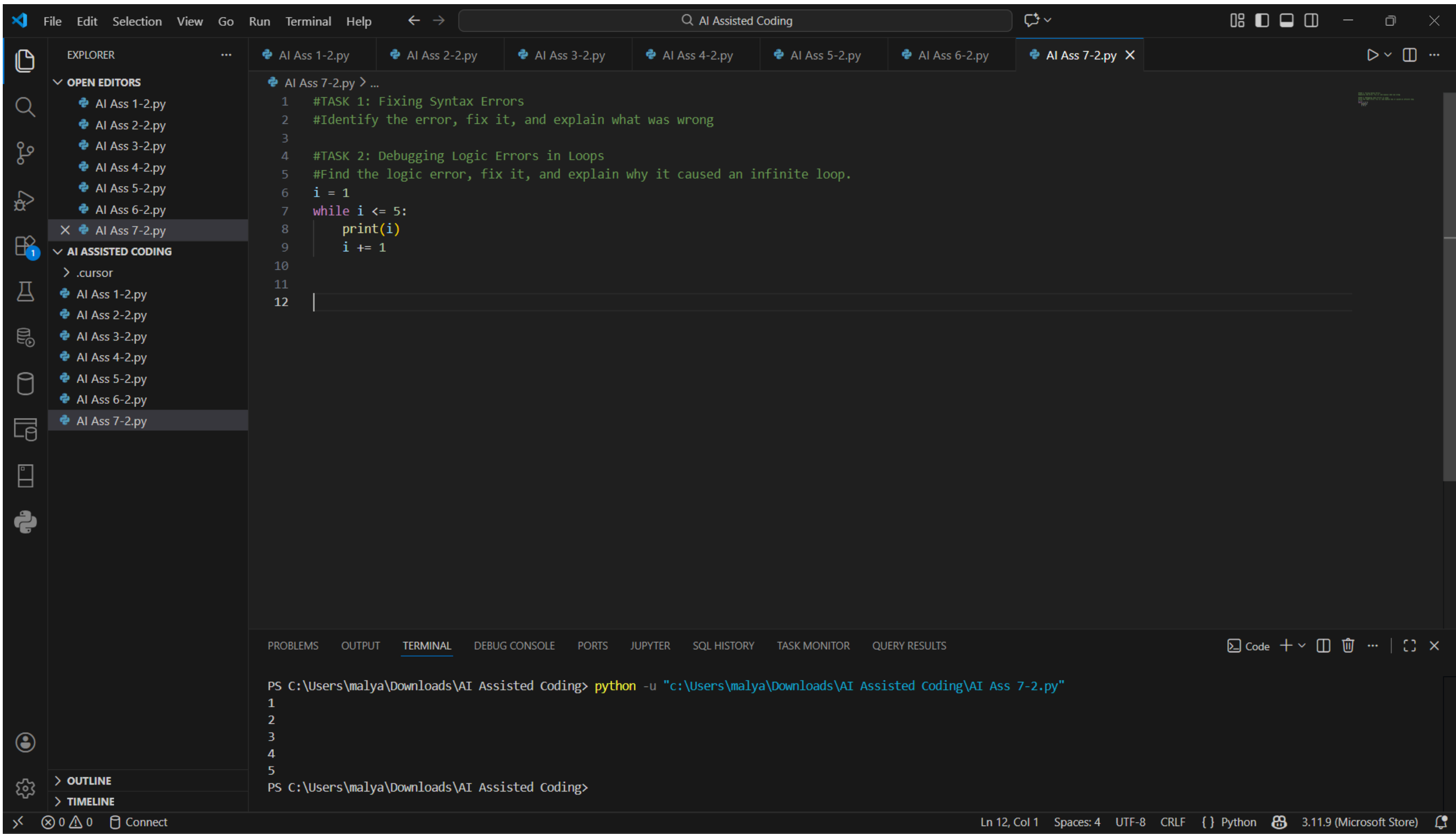
Explanation

The given program had a **syntax error** because the function definition was missing a **colon (:)** at the end of the `def` statement.
In Python, a colon is mandatory to indicate the start of a function block.
After adding the colon, the function executed correctly and returned the sum of two numbers.



```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
i = 1
while i <= 5:
    print(i)


```

```
1
1
1
1
1
1
1
1
```


File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
.cursor
AI Ass 1-2.py
AI Ass 2-2.py
AI Ass 3-2.py
AI Ass 4-2.py
AI Ass 5-2.py
AI Ass 6-2.py
AI Ass 7-2.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Code
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)
Connect

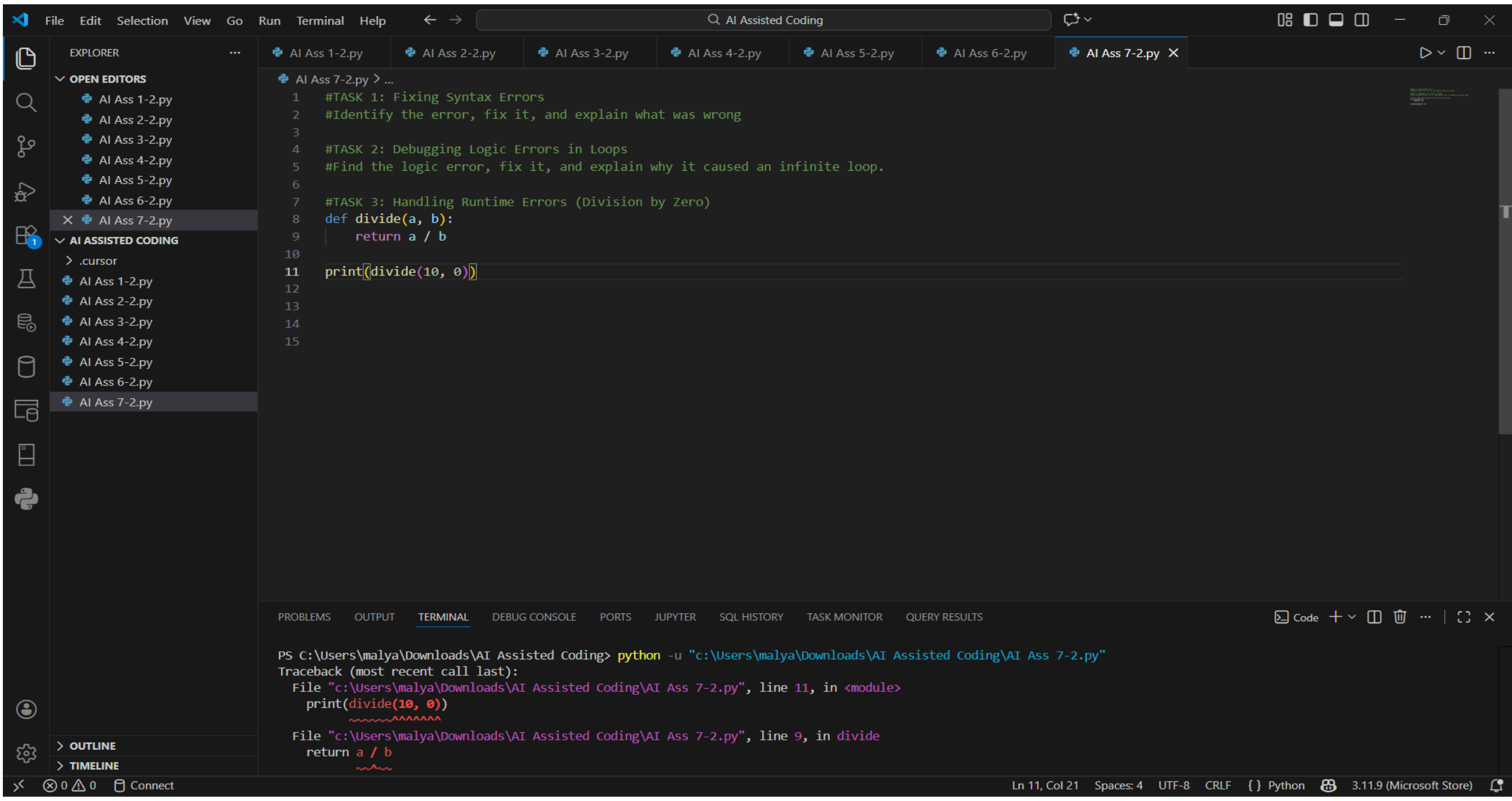
```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.
i = 1
while i <= 5:
    print(i)
    i += 1
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
1
2
3
4
5
PS C:\Users\malya\Downloads\AI Assisted Coding>
```

Task 2: Debugging Logic Errors in Loops

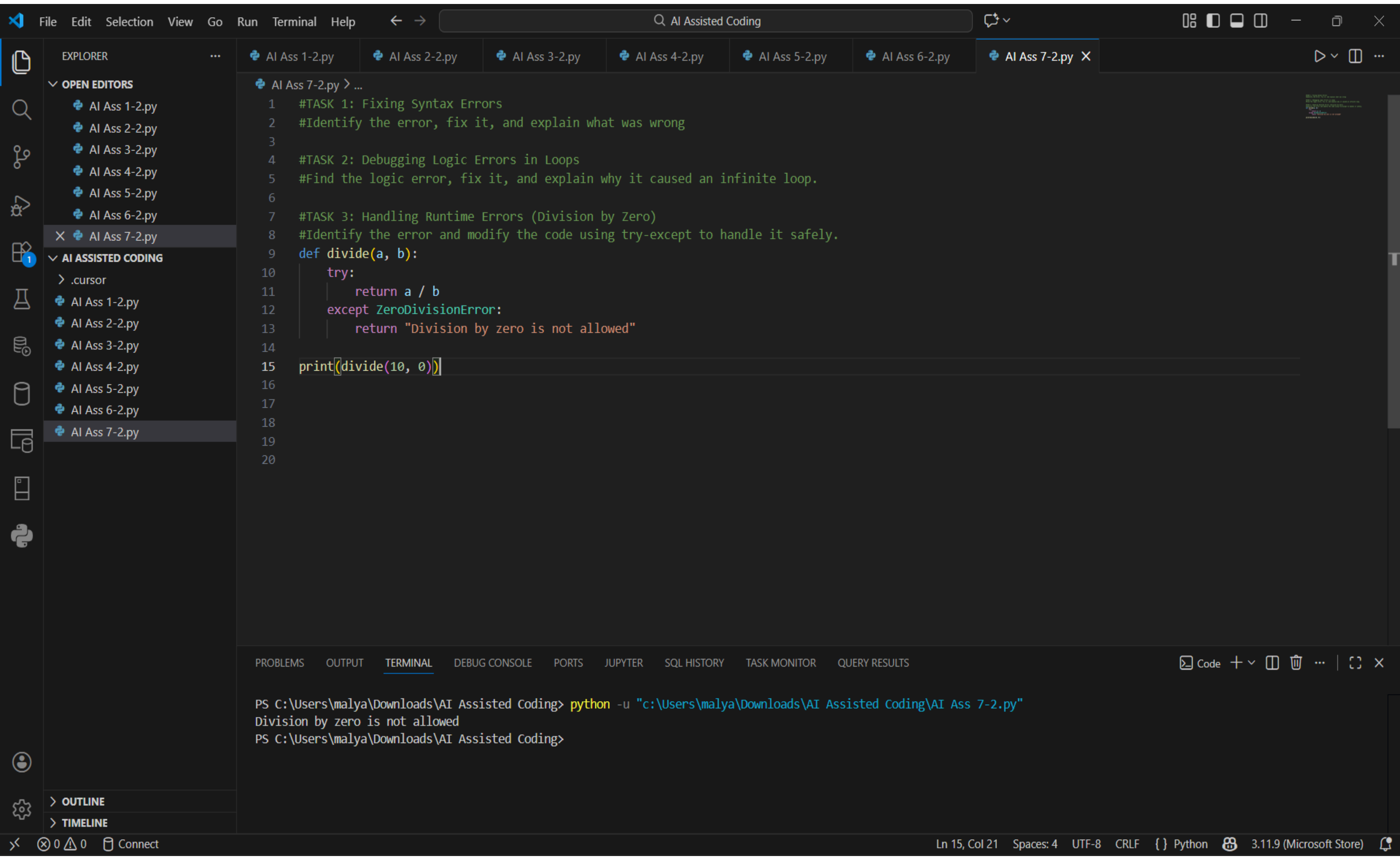
Explanation

The loop ran infinitely because the loop control variable was **never updated**.
Since the value of `i` did not change, the condition `i <= 5` always remained true.
By adding an increment statement (`i += 1`), the loop progressed correctly and terminated after reaching the limit.


File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
.cursor
AI Ass 1-2.py
AI Ass 2-2.py
AI Ass 3-2.py
AI Ass 4-2.py
AI Ass 5-2.py
AI Ass 6-2.py
AI Ass 7-2.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 11, Col 21 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
def divide(a, b):
    return a / b

print(divide(10, 0))
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Traceback (most recent call last):
  File "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py", line 11, in <module>
    print(divide(10, 0))
          ~~~~~~^^^^^^^

  File "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py", line 9, in divide
    return a / b
           ~~^~~
```


File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 15, Col 21 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

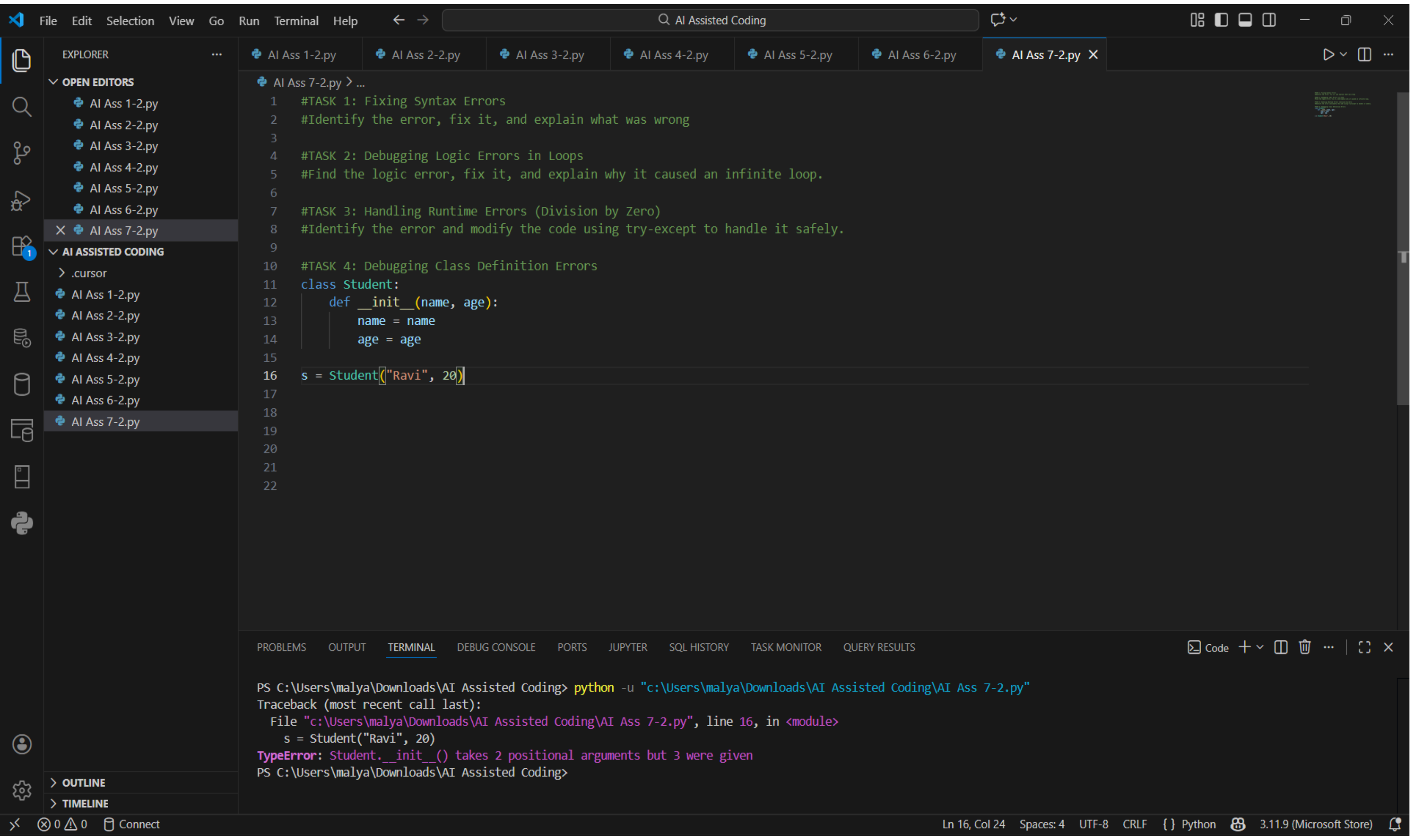
```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
#Identify the error and modify the code using try-except to handle it safely.
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Division by zero is not allowed"

print(divide(10, 0))
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Division by zero is not allowed
PS C:\Users\malya\Downloads\AI Assisted Coding>
```

Task 3: Handling Runtime Errors (Division by Zero)

Explanation

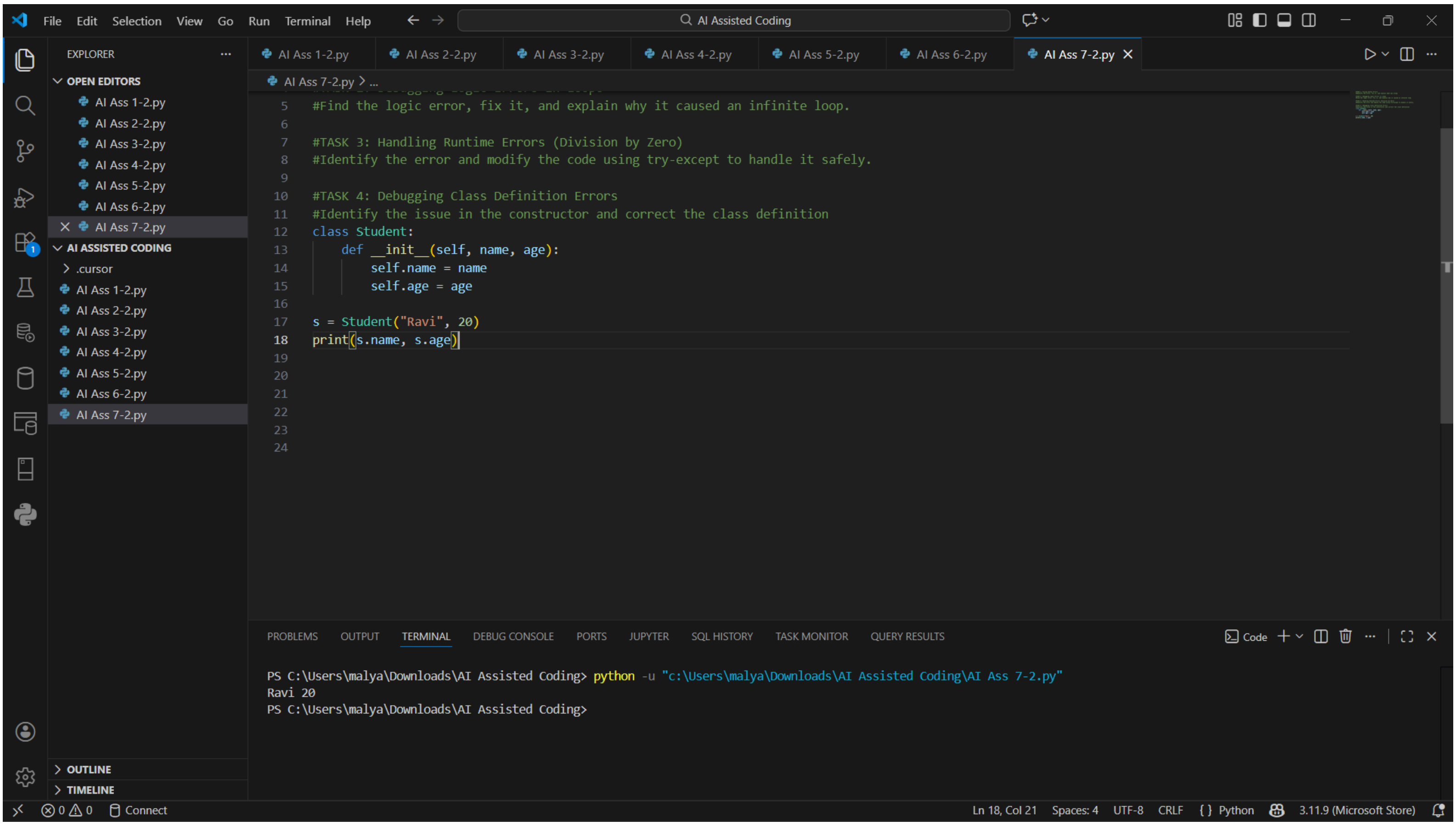
The program caused a **runtime error** when division by zero occurred.
Python raises a `ZeroDivisionError` when a number is divided by zero.
To prevent the program from crashing, a `try-except` block was used to handle the error gracefully and display a meaningful message.



File Edit Selection View Go Run Terminal Help
AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
.cursor
AI Ass 1-2.py
AI Ass 2-2.py
AI Ass 3-2.py
AI Ass 4-2.py
AI Ass 5-2.py
AI Ass 6-2.py
AI Ass 7-2.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Code
Ln 16, Col 24 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)
0 0 Connect

```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
#Identify the error and modify the code using try-except to handle it safely.

#TASK 4: Debugging Class Definition Errors
class Student:
    def __init__(name, age):
        name = name
        age = age

s = Student("Ravi", 20)
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Traceback (most recent call last):
  File "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py", line 16, in <module>
    s = Student("Ravi", 20)
TypeError: Student.__init__() takes 2 positional arguments but 3 were given
PS C:\Users\malya\Downloads\AI Assisted Coding>
```



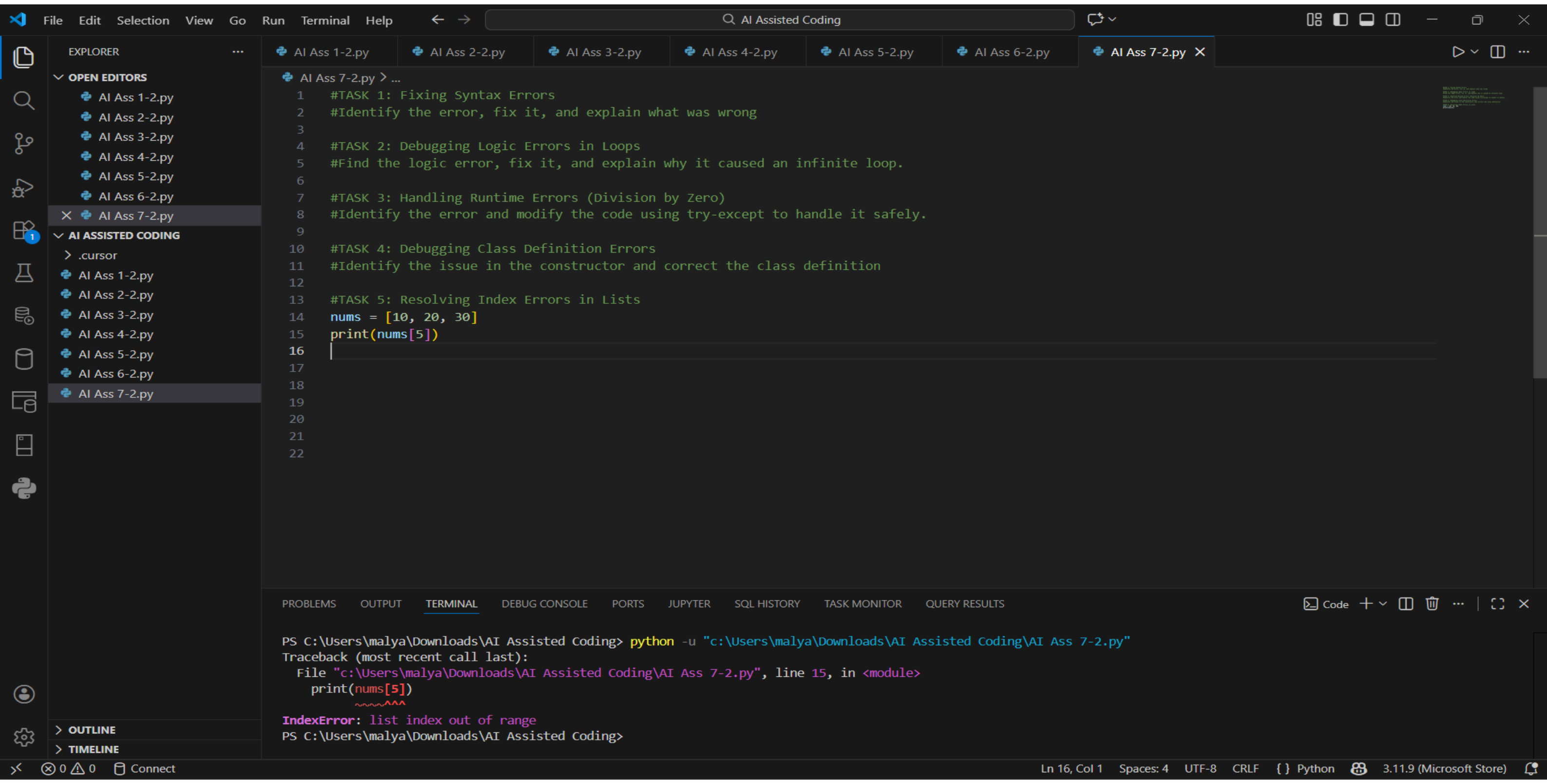
```python
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
#Identify the error and modify the code using try-except to handle it safely.

#TASK 4: Debugging Class Definition Errors
#Identify the issue in the constructor and correct the class definition
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

s = Student("Ravi", 20)
print(s.name, s.age)
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Ravi 20
PS C:\Users\malya\Downloads\AI Assisted Coding>
```

Task 4: Debugging Class Definition Errors

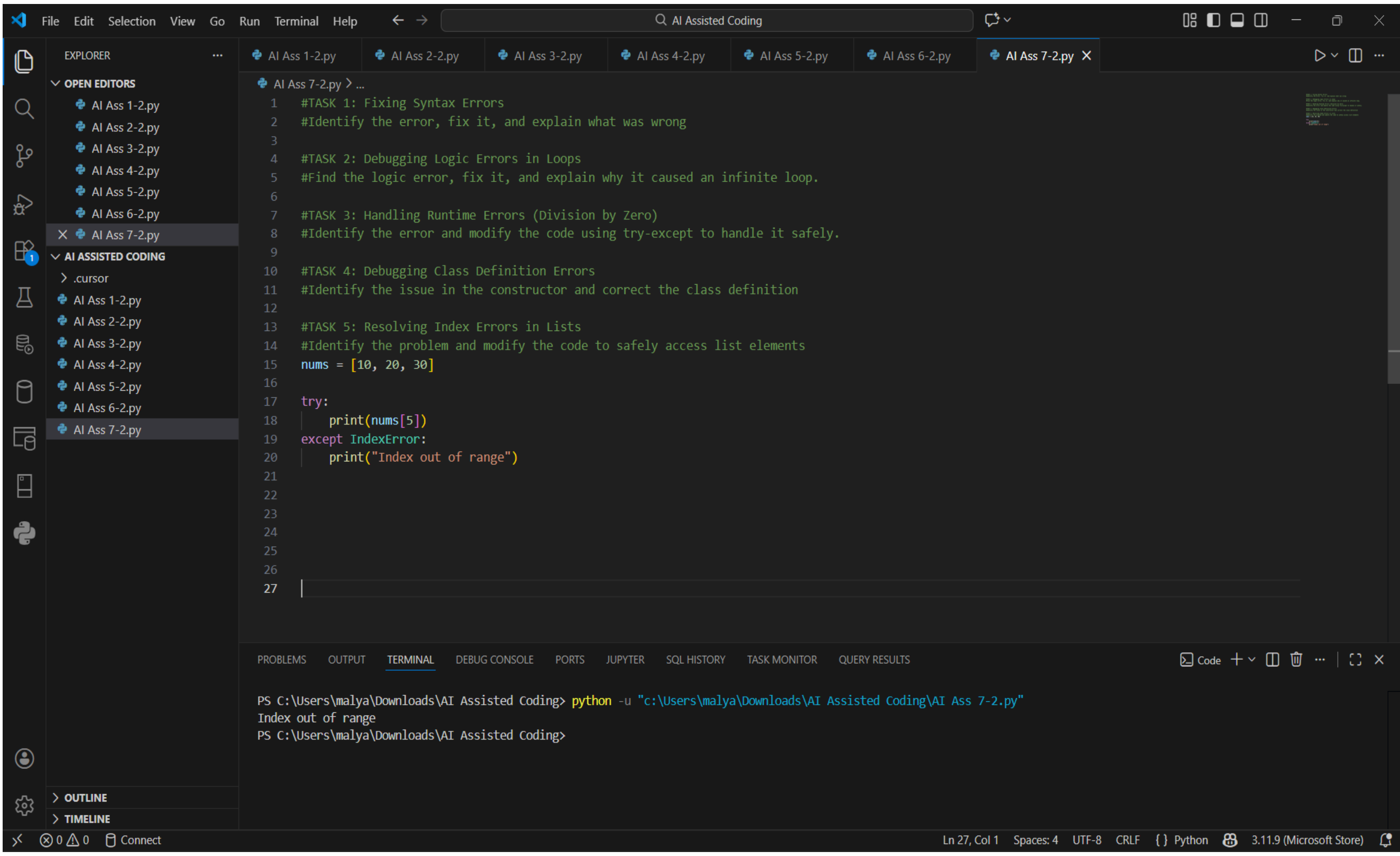
Explanation

The constructor method `__init__()` was defined without the `self` parameter.
In Python, `self` refers to the current object and is required to access instance variables.
Adding `self` and properly assigning values to instance variables corrected the class definition.


AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
AI Ass 7-2.py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
#Identify the error and modify the code using try-except to handle it safely.

#TASK 4: Debugging Class Definition Errors
#Identify the issue in the constructor and correct the class definition

#TASK 5: Resolving Index Errors in Lists
nums = [10, 20, 30]
print(nums[5])
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Traceback (most recent call last):
  File "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py", line 15, in <module>
    print(nums[5])
          ~~~~^^^
IndexError: list index out of range
PS C:\Users\malya\Downloads\AI Assisted Coding>
```


AI Assisted Coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
AI Ass 7-2.py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS
Ln 27, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

```python
#TASK 1: Fixing Syntax Errors
#Identify the error, fix it, and explain what was wrong

#TASK 2: Debugging Logic Errors in Loops
#Find the logic error, fix it, and explain why it caused an infinite loop.

#TASK 3: Handling Runtime Errors (Division by Zero)
#Identify the error and modify the code using try-except to handle it safely.

#TASK 4: Debugging Class Definition Errors
#Identify the issue in the constructor and correct the class definition

#TASK 5: Resolving Index Errors in Lists
#Identify the problem and modify the code to safely access list elements
nums = [10, 20, 30]

try:
    print(nums[5])
except IndexError:
    print("Index out of range")
```

```
PS C:\Users\malya\Downloads\AI Assisted Coding> python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"
Index out of range
PS C:\Users\malya\Downloads\AI Assisted Coding>
```

Task 5: Resolving Index Errors in Lists

Explanation

The program attempted to access an index that does not exist in the list, causing an `IndexError`.
Lists in Python have fixed index ranges starting from 0.
Using exception handling ensured safe access and prevented the program from crashing when an invalid index was used.