| SCHOOLOFCOMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE | DEPARTMENTOFCOMPUTER SCIENCE ENGINEERING |
|---|---|

| ProgramName:B. Tech | AssignmentType: Lab | AcademicYear:2025-2026 |
|---|---|---|

| | |
|---|---|
| **CourseCoordinatorName** | Venkataramana Veeramsetty |
| **Instructor(s)Name** | Dr. V. Venkataramana (Co-ordinator)<br>Dr. T. Sampath Kumar<br>Dr. Pramoda Patro<br>Dr. Brij Kishor Tiwari<br>Dr.J.Ravichander<br>Dr. Mohammand Ali Shaik<br>Dr. Anirodh Kumar<br>Mr. S.Naresh Kumar<br>Dr. RAJESH VELPULA<br>Mr. Kundhan Kumar<br>Ms. Ch.Rajitha<br>Mr. M Prakash<br>Mr. B.Raju<br>Intern 1 (Dharma teja)<br>Intern 2 (Sai Prasad)<br>Intern 3 (Sowmya)<br>NS_2 ( Mounika) |

| | | | |
|---|---|---|---|
| **CourseCode** | 24CS002PC215 | **CourseTitle** | AI Assisted Coding |
| **Year/Sem** | II/I | **Regulation** | R24 |
| **DateandDay of Assignment** | Week4 - Thursday | **Time(s)** | |
| **Duration** | 2 Hours | **Applicableto Batches** | |

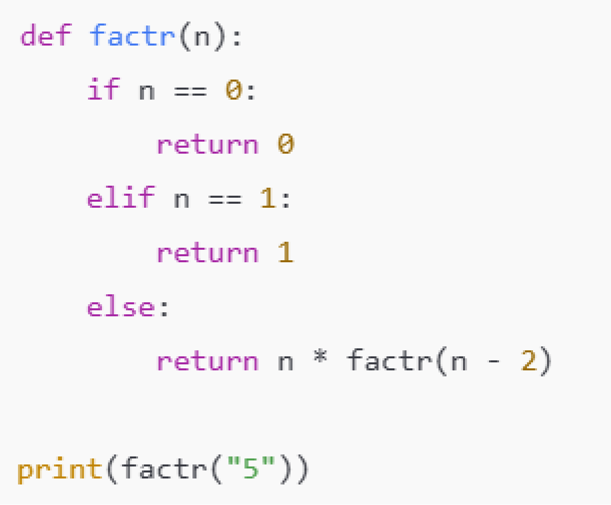
**AssignmentNumber:7.4(Presentassignmentnumber)/24(Totalnumberofassignments)**

| Q.No. | Question | *ExpectedTime to complete* |
|---|---|---|
| 1 | Lab 7: Error Debugging with AI – Systematic Approaches to Finding and Fixing Bugs<br>Lab Objectives:<br>• To identify and correct syntax, logic, and runtime errors in Python programs using AI tools. | Week4 - Thursday |

- To understand common programming bugs and AI-assisted debugging suggestions.
- To evaluate how AI explains, detects, and fixes different types of coding errors.
- To build confidence in using AI to perform structured debugging practices.
  Lab Outcomes (LOs):
  After completing this lab, students will be able to:
- Use AI tools to detect and correct syntax, logic, and runtime errors.
- Interpret AI-suggested bug fixes and explanations.
- Apply systematic debugging strategies supported by AI-generated insights.
- Refactor buggy code using responsible and reliable programming patterns.

**Task Description #1:**

• Introduce a buggy Python function that calculates the factorial of a number using recursion. Use Copilot or Cursor AI to detect and fix the logical or syntax errors.

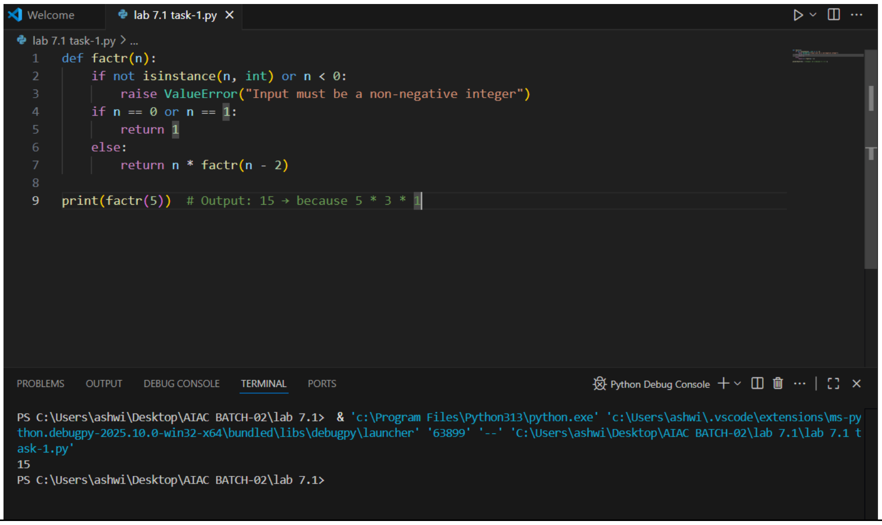


```python
def factr(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return n * factr(n - 2)


print(factr("5"))
```

**Expected Outcome #1:**

•Copilot or Cursor AI correctly identifies missing base condition or incorrect recursive call and suggests a functional factorial implementation.

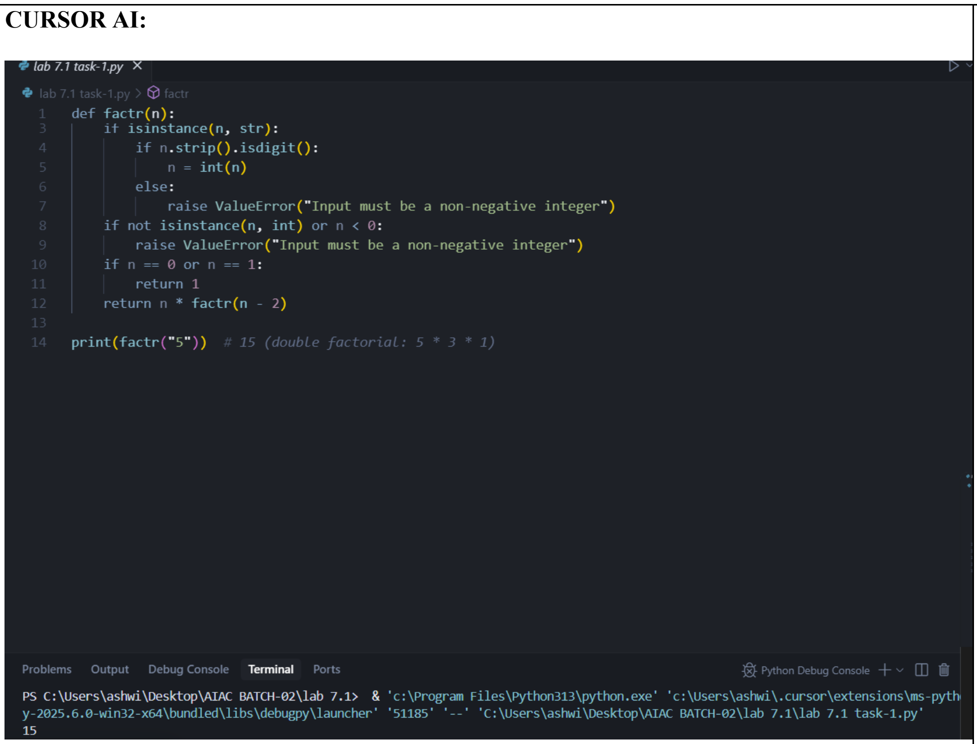
VS CODE:



```python
def factr(n):
    if not isinstance(n, int) or n < 0:
        raise ValueError("Input must be a non-negative integer")
    if n == 0 or n == 1:
        return 1
    else:
        return n * factr(n - 2)

print(factr(5))  # Output: 15 → because 5 * 3 * 1
```

```
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '63899' '--' 'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task-1.py'
15
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
```

**CURSOR AI:**

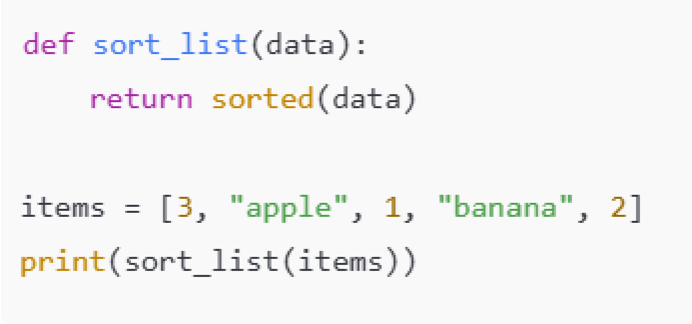


```python
def factr(n):
    if isinstance(n, str):
        if n.strip().isdigit():
            n = int(n)
        else:
            raise ValueError("Input must be a non-negative integer")
    if not isinstance(n, int) or n < 0:
        raise ValueError("Input must be a non-negative integer")
    if n == 0 or n == 1:
        return 1
    return n * factr(n - 2)

print(factr("5"))  # 15 (double factorial: 5 * 3 * 1)
```

```
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\ashwi\.cursor\extensions\ms-pyth
y-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '51185' '--' 'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task-1.py'
15
```

**Task Description #2:**

•Provide a list sorting function that fails due to a type error (e.g., sorting list with mixed integers and strings). Prompt AI to detect the issue and fix the code for consistent sorting.

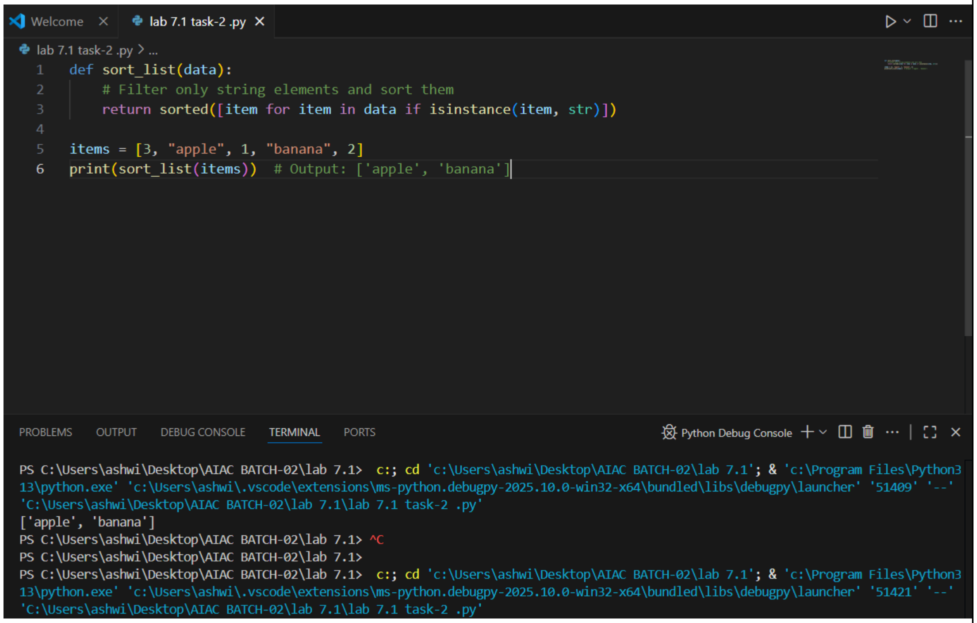


```python
def sort_list(data):
    return sorted(data)


items = [3, "apple", 1, "banana", 2]
print(sort_list(items))
```

**Expected Outcome #2:**

•AI detects the type inconsistency and either filters or converts list elements, ensuring successful sorting without a crash

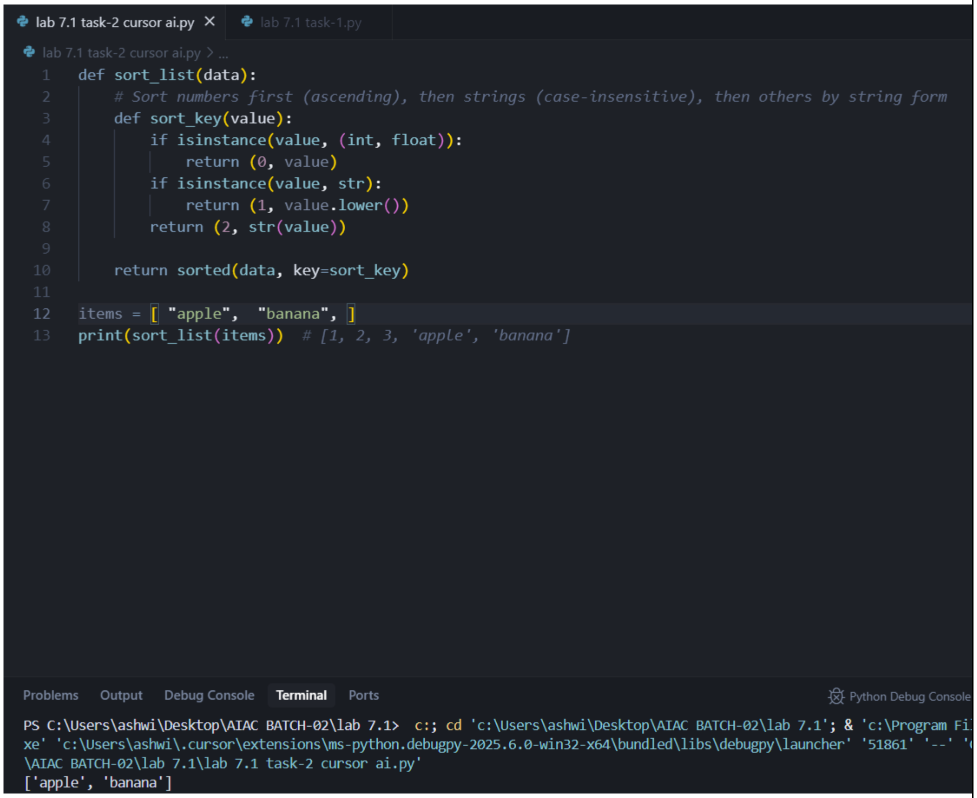
VS CODE:



```python
def sort_list(data):
    # Filter only string elements and sort them
    return sorted([item for item in data if isinstance(item, str)])

items = [3, "apple", 1, "banana", 2]
print(sort_list(items))  # Output: ['apple', 'banana']
```

```
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Files\Python3
13\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '51409' '--'
'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task-2 .py'
['apple', 'banana']
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1> ^C
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Files\Python3
13\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '51421' '--'
'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task-2 .py'
```

**CURSOR AI:**



lab 7.1 task-2 cursor ai.py
lab 7.1 task-1.py
lab 7.1 task-2 cursor ai.py > ...

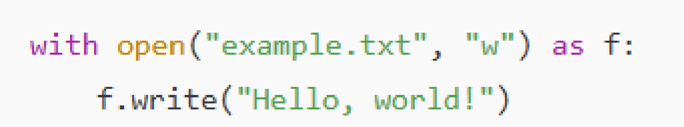
```python
def sort_list(data):
    # Sort numbers first (ascending), then strings (case-insensitive), then others by string form
    def sort_key(value):
        if isinstance(value, (int, float)):
            return (0, value)
        if isinstance(value, str):
            return (1, value.lower())
        return (2, str(value))

    return sorted(data, key=sort_key)

items = [ "apple",  "banana", ]
print(sort_list(items))  # [1, 2, 3, 'apple', 'banana']
```

```
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Fi
xe' 'c:\Users\ashwi\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '51861' '--' '
\AIAC BATCH-02\lab 7.1\lab 7.1 task-2 cursor ai.py'
['apple', 'banana']
```

**Task Description #3:**

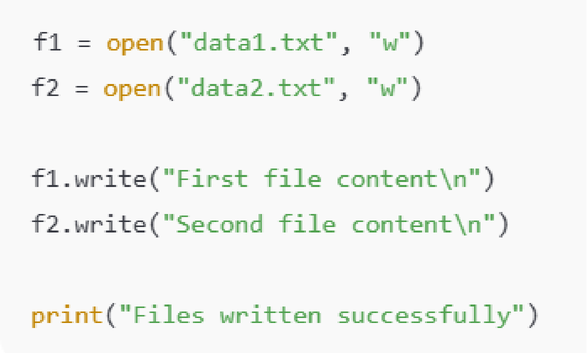
• Write a Python snippet for file handling that opens a file but forgets to close it. Ask Copilot or Cursor AI to improve it using the best practice (e.g., with open() block).

**Code1**



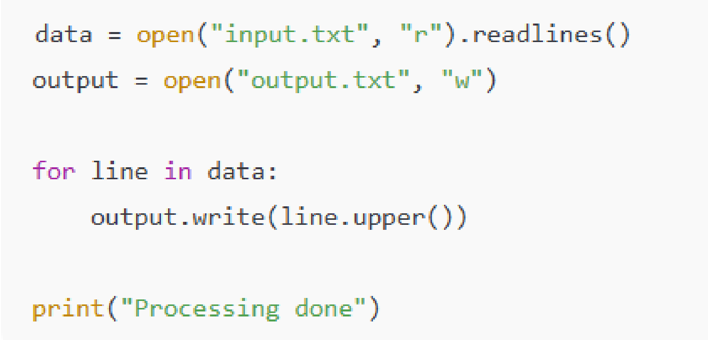
```python
with open("example.txt", "w") as f:
    f.write("Hello, world!")
```

**Code2**



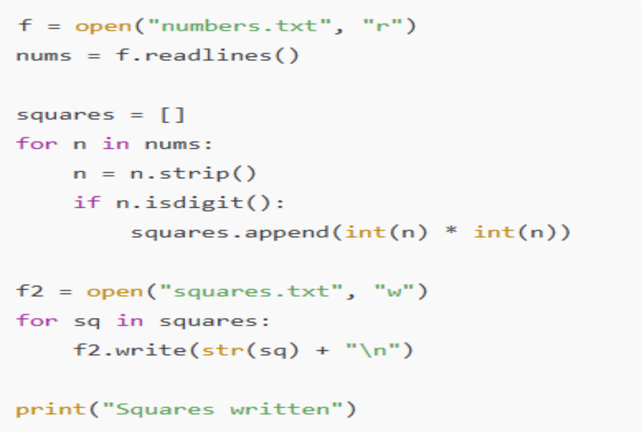
```python
f1 = open("data1.txt", "w")
f2 = open("data2.txt", "w")

f1.write("First file content\n")
f2.write("Second file content\n")

print("Files written successfully")
```

**Code3**



```python
data = open("input.txt", "r").readlines()
output = open("output.txt", "w")

for line in data:
    output.write(line.upper())

print("Processing done")
```

**Code4:**



```python
f = open("numbers.txt", "r")
nums = f.readlines()

squares = []
for n in nums:
    n = n.strip()
    if n.isdigit():
        squares.append(int(n) * int(n))

f2 = open("squares.txt", "w")
for sq in squares:
    f2.write(str(sq) + "\n")

print("Squares written")
```

**Expected Outcome #3:**

- AI refactors the code to use a context manager, preventing resource leakage and runtime warnings.

**Task Description #4:**

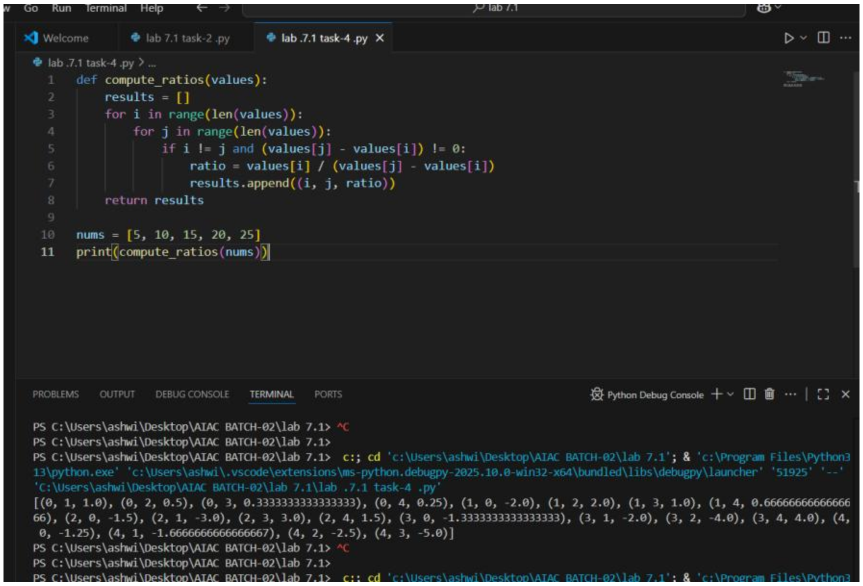
- Provide a piece of code with a ZeroDivisionError inside a loop. Ask AI to add error handling using try-except and continue execution safely.

```python
def compute_ratios(values):
    results = []
    for i in range(len(values)):
        for j in range(i, len(values)):
            ratio = values[i] / (values[j] - values[i])
            results.append((i, j, ratio))
    return results


nums = [5, 10, 15, 20, 25]
print(compute_ratios(nums))
```

**Expected Outcome #4:**

• Copilot adds a try-except block around the risky operation, preventing crashes and printing a meaningful error message.

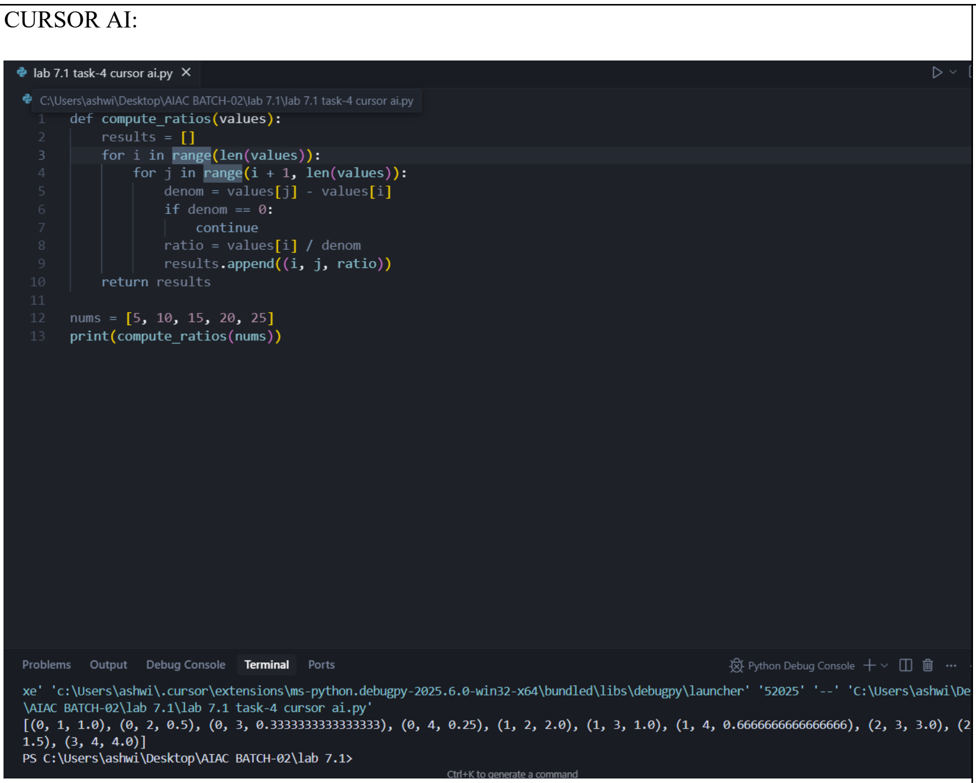
VS CODE:



```python
def compute_ratios(values):
    results = []
    for i in range(len(values)):
        for j in range(len(values)):
            if i != j and (values[j] - values[i]) != 0:
                ratio = values[i] / (values[j] - values[i])
                results.append((i, j, ratio))
    return results

nums = [5, 10, 15, 20, 25]
print(compute_ratios(nums))
```

```
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1> ^C
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Files\Python3
13\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '51925' '--'
'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab .7.1 task-4 .py'
[(0, 1, 1.0), (0, 2, 0.5), (0, 3, 0.3333333333333333), (0, 4, 0.25), (1, 0, -2.0), (1, 2, 2.0), (1, 3, 1.0), (1, 4, 0.66666666666666
66), (2, 0, -1.5), (2, 1, -3.0), (2, 3, 3.0), (2, 4, 1.5), (3, 0, -1.3333333333333333), (3, 1, -2.0), (3, 2, -4.0), (3, 4, 4.0), (4,
 0, -1.25), (4, 1, -1.6666666666666667), (4, 2, -2.5), (4, 3, -5.0)]
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1> ^C
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>  c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Files\Python3
```

CURSOR AI:


lab 7.1 task-4 cursor ai.py
C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task-4 cursor ai.py
Problems
Output
Debug Console
Terminal
Ports
Python Debug Console
Ctrl+K to generate a command

```python
def compute_ratios(values):
    results = []
    for i in range(len(values)):
        for j in range(i + 1, len(values)):
            denom = values[j] - values[i]
            if denom == 0:
                continue
            ratio = values[i] / denom
            results.append((i, j, ratio))
    return results

nums = [5, 10, 15, 20, 25]
print(compute_ratios(nums))
```

```
xe' 'c:\Users\ashwi\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '52025' '--' 'C:\Users\ashwi\De
\AIAC BATCH-02\lab 7.1\lab 7.1 task-4 cursor ai.py'
[(0, 1, 1.0), (0, 2, 0.5), (0, 3, 0.3333333333333333), (0, 4, 0.25), (1, 2, 2.0), (1, 3, 1.0), (1, 4, 0.6666666666666666), (2, 3, 3.0), (2
1.5), (3, 4, 4.0)]
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
```

**Task Description #5:**

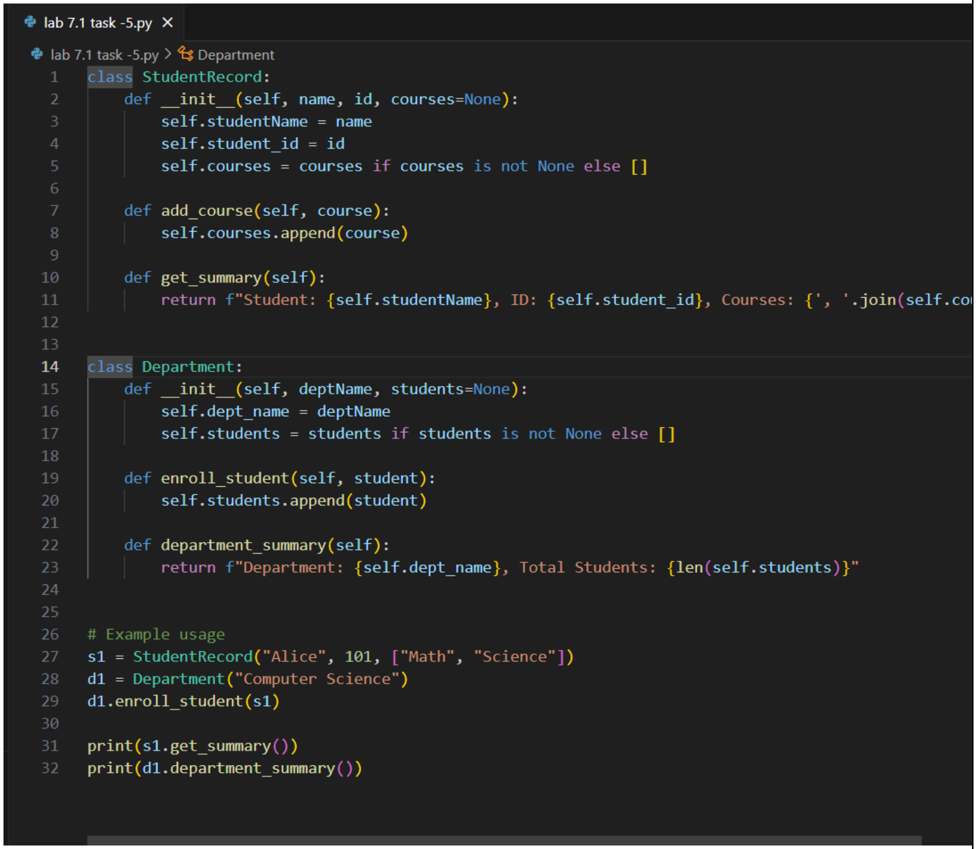
• Include a buggy class definition with incorrect __init__ parameters or attribute references. Ask AI to analyze and correct the constructor and attribute usage.

```
class StudentRecord:
    def __init__(self, name, id, courses=[]):
        self.studentName = names
        self.student_id = id
        self.courses = courseList

    def add_course(self, course):
        self.courses.append(course)

    def get_summary(self):
        return f"Student: {self.studentName}, ID: {self.student_id}, Courses: {', '.join(self.courses)}"

class Department:
    def __init__(self, deptName, students=None):
        self.dept_name = deptName
        self.students = students

    def enroll_student(self, student):
        self.students.append(student)

    def department_summary(self):
        return f"Department: {self.dept_name}, Total Students: {len(self.student)}"


s1 = StudentRecord("Alice", 101, ["Math", "Science"])
d1 = Department("Computer Science")
d1.enroll_student(s1)
print(s1.get_summary())
print(d1.department_summary())
```

**Expected Outcome #5:**

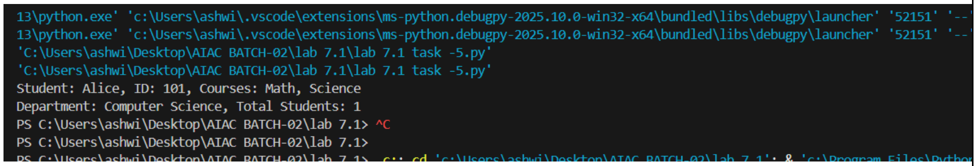
- Copilot identifies mismatched parameters or missing self references and rewrites the class with accurate initialization and usage.

VS CODE:



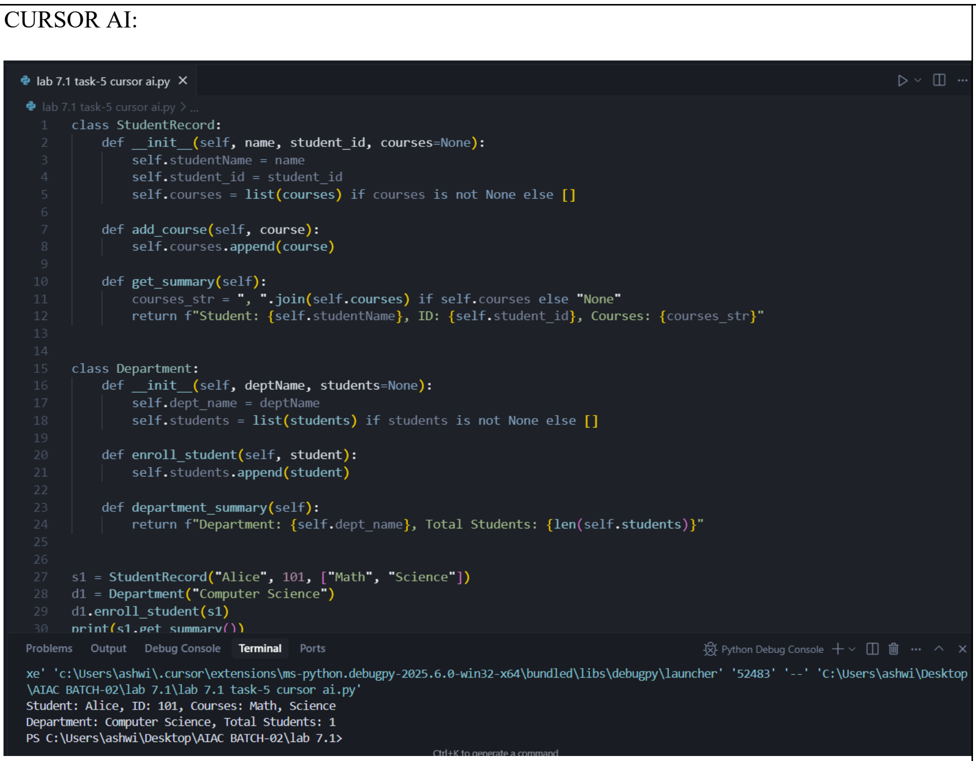
```python
class StudentRecord:
    def __init__(self, name, id, courses=None):
        self.studentName = name
        self.student_id = id
        self.courses = courses if courses is not None else []

    def add_course(self, course):
        self.courses.append(course)

    def get_summary(self):
        return f"Student: {self.studentName}, ID: {self.student_id}, Courses: {', '.join(self.co


class Department:
    def __init__(self, deptName, students=None):
        self.dept_name = deptName
        self.students = students if students is not None else []

    def enroll_student(self, student):
        self.students.append(student)

    def department_summary(self):
        return f"Department: {self.dept_name}, Total Students: {len(self.students)}"


# Example usage
s1 = StudentRecord("Alice", 101, ["Math", "Science"])
d1 = Department("Computer Science")
d1.enroll_student(s1)

print(s1.get_summary())
print(d1.department_summary())
```

OUTPUT:



```
13\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '52151' '--'
13\python.exe' 'c:\Users\ashwi\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '52151' '--'
'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task -5.py'
'C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1\lab 7.1 task -5.py'
Student: Alice, ID: 101, Courses: Math, Science
Department: Computer Science, Total Students: 1
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1> ^C
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1> c:; cd 'c:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1'; & 'c:\Program Files\Pythor
```

CURSOR AI:



lab 7.1 task-5 cursor ai.py
lab 7.1 task-5 cursor ai.py > ...

```python
class StudentRecord:
    def __init__(self, name, student_id, courses=None):
        self.studentName = name
        self.student_id = student_id
        self.courses = list(courses) if courses is not None else []

    def add_course(self, course):
        self.courses.append(course)

    def get_summary(self):
        courses_str = ", ".join(self.courses) if self.courses else "None"
        return f"Student: {self.studentName}, ID: {self.student_id}, Courses: {courses_str}"


class Department:
    def __init__(self, deptName, students=None):
        self.dept_name = deptName
        self.students = list(students) if students is not None else []

    def enroll_student(self, student):
        self.students.append(student)

    def department_summary(self):
        return f"Department: {self.dept_name}, Total Students: {len(self.students)}"


s1 = StudentRecord("Alice", 101, ["Math", "Science"])
d1 = Department("Computer Science")
d1.enroll_student(s1)
print(s1.get_summary())
```

```
xe' 'c:\Users\ashwi\.cursor\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '52483' '--' 'C:\Users\ashwi\Desktop
\AIAC BATCH-02\lab 7.1\lab 7.1 task-5 cursor ai.py'
Student: Alice, ID: 101, Courses: Math, Science
Department: Computer Science, Total Students: 1
PS C:\Users\ashwi\Desktop\AIAC BATCH-02\lab 7.1>
```