

AI ASSIGNMENT-7.4

NAME:MD NASRIN
2503a51l23

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.

Expected Outcome #1:

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

OUTPUT:

BUGGY CODE:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The Explorer sidebar on the left lists files: merge.py, binary.py, store.py, and function.py. The function.py file is open in the editor, showing the following code:

```
function.py > factorial
1 def factorial(n):
2     if n == 0:
3         return 0
4     else:
5         return n * factorial(n - 1)
```

The terminal at the bottom shows the command PS C:\Users\HASINI\Desktop\ai_code> & c:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/Desktop/ai_code/function.py" being run. The output shows the command was successful. A floating window titled "Ask about your code." is visible on the right side of the screen, containing the message "Ask about your code. AI responses may be inaccurate. Generate instructions to onboard AI onto your codebase." and a "Add Context..." button.

CORRECTED CODE:

The screenshot shows a code editor interface with a dark theme. The left sidebar is titled 'EXPLORER' and lists several files under 'AI CODE': merge.py, binary.py, store.py, function.py (which is currently selected), factorial.py, discount.py, list of number.py, list of students dictio..., merge.py, shopping cart.py, and store.py. The main area displays the contents of the function.py file:

```
function.py > factorial
1 def factorial(n):
2     if n < 0:
3         raise ValueError("factorial is not defined for negative numbers.")
4     elif n==0 or n==1:
5         return 1
6     else:
7         return n * factorial(n - 1)
8     try:
9         num=int(input('enter a non negative integer:'))
10        print(f"factorial of {num} is {factorial(num)}")
11    except ValueError as e:
12        print("error.",e)
```

The screenshot shows a terminal window with a dark theme. The command PS C:\Users\WASINI\OneDrive\Desktop\ai code> is followed by the output of the factorial program:

```
enter a non negative integer:6
factorial of 6 is 720
PS C:\Users\WASINI\OneDrive\Desktop\ai code>
```

OBSERVATION: Negative inputs cause infinite recursion and eventually a RecursionError. AI generated a corrected code with error exception which does not allow negative values.

- Returning 0 for $n == 0$ contradicts the mathematical definition of $\text{factorial}(0) = 1$
- **Correct Base Case:** Returns 1 for both 0 and 1, aligning with factorial rules

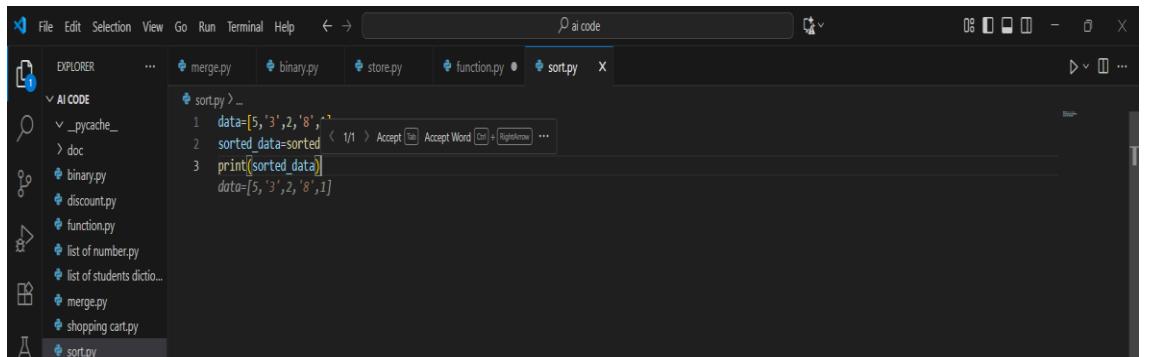
Task Description #2:

- 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.

Expected Outcome #2:

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

BUGGY CODE:



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files: merge.py, binary.py, store.py, function.py, and sort.py. The sort.py file is currently open in the editor. The code contains the following lines:

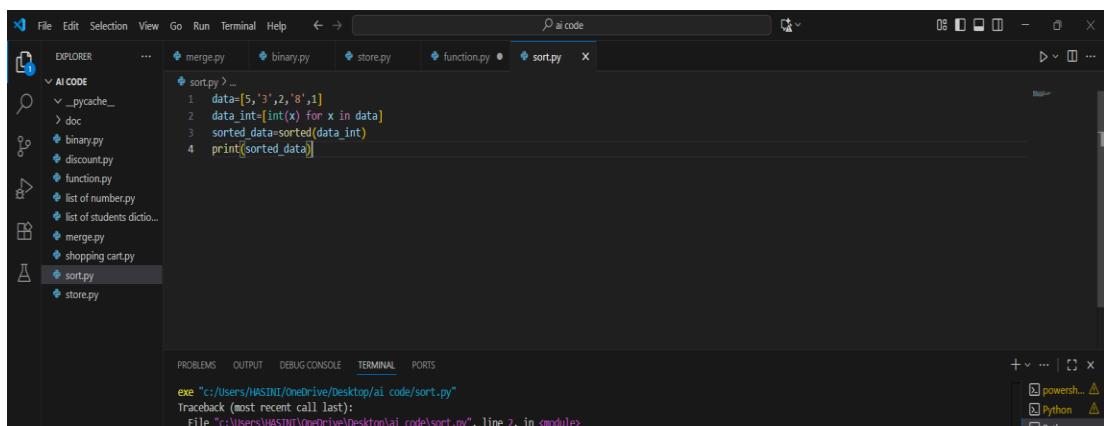
```
sort.py > ...
1 data=[5,'3',2,'8',1]
2 sorted_data=sorted(data)
3 print(sorted_data)
data=[5, '3', 2, '8', 1]
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & C:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
Traceback (most recent call last):
  File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 2, in <module>
    sorted_data=sorted(data)
TypeError: '<' not supported between instances of 'str' and 'int'
PS C:\Users\HASINI\OneDrive\Desktop\ai code> []
```

The status bar at the bottom indicates the code is 3.13.7 (Microsoft Store) and the date is 18-09-2025.

CORRECTED CODE:



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files. The sort.py file is currently open in the editor. The code has been modified to:

```
sort.py > ...
1 data=[5,'3',2,'8',1]
2 data_int=[int(x) for x in data]
3 sorted_data=sorted(data_int)
4 print(sorted_data)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & C:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
Traceback (most recent call last):
  File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 2, in <module>
    sorted_data=sorted(data)
TypeError: 'int' object is not iterable
PS C:\Users\HASINI\OneDrive\Desktop\ai code> [1, 2, 3, 5, 8]
```

The status bar at the bottom indicates the code is 3.13.7 (Microsoft Store) and the date is 18-09-2025.

OBSERVATION: Mixed datatypes are given as input. AI has converted them into one datatype and sorted the data.

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).

Expected Outcome #3:

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

BUGGY CODE:

The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows files in the workspace, including `merge.py`, `binary.py`, `store.py`, `function.py`, and `sort.py`.
- CODE EDITOR:** Displays the following Python code:

```

sort.py > ...
1 file=open("file1.txt",'r')
2 content=file.read()
3 print(content)

```
- TERMINAL:** Shows the output of running the code, which results in a `FileNotFoundException` because the file does not exist:

```

FileNotFoundException: [Errno 2] No such file or directory: 'file1.txt'
PS C:\Users\HASINI\OneDrive\Desktop\ai code & C:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
Traceback (most recent call last):
  File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 1, in <module>
    file=open("file1.txt",'r')
FileNotFoundError: [Errno 2] No such file or directory: 'file1.txt'
PS C:\Users\HASINI\OneDrive\Desktop\ai code & C:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
Traceback (most recent call last):
  File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 1, in <module>
    file=open("file1.txt",'r')
FileNotFoundError: [Errno 2] No such file or directory: 'file1.txt'
PS C:\Users\HASINI\OneDrive\Desktop\ai code> []

```
- STATUS BAR:** Shows the Python extension is active, along with other status information like battery level (32°C) and system icons.

CORRECTED CODE:

```

# Safely open and read the file
with open("file1.txt", "r") as file:
    content = file.read()

print(content)

```

```
PS C:\today> c;; cd 'c:\today'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\ADHARSH\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '53058' '--' 'c:\today\seven-3.py'
print("hello world")
print("this is me")hello widget
```

OBSERVATION:

The file.close() is missing so, file remains open after reading, which can:

Leak system resources, Lock the file (especially on Windows),

Trigger runtime warnings or errors in larger applications

No Exception Handling: If the file doesn't exist or can't be read, the code will crash without a fallback.

Using a Context Manager (with open):

Automatically closes the file when the block exits—even if an error occurs

Prevents resource leakage and improves reliability.

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.

Expected Outcome #4:

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

BUGGY CODE:

```

1 numbers=[10,5,0,2]
2 for num in numbers:
3     result=100/num
4     print(f"100 divided by {num} is {result}")
5

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
Traceback (most recent call last):
File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 1, in <module>
with open("file1.txt", "r") as file:

FileNotFoundException: [Errno 2] No such file or directory: 'file1.txt'
PS C:\Users\HASINI\OneDrive\Desktop\ai code & C:/Users/HASINI/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py"
100 divided by 10 is 10.0
100 divided by 5 is 20.0
Traceback (most recent call last):
File "c:/Users/HASINI/OneDrive/Desktop/ai code/sort.py", line 3, in <module>
result=100/num
~~~~~  
ZeroDivisionError: division by zero  
PS C:\Users\HASINI\OneDrive\Desktop\ai code> []

Air: Satisfactory  
Tomorrow

In 4, Col 47 Spaces: 4 UTF-8 CRLF {} Python Chat quota reached 3.13.7 (Microsoft Store) Go Live ENG IN 14:44 18-09-2025

## CORRECTED CODE:

```

numbers = [10, 5, 0, 2]

for num in numbers:
    try:
        result = 100 / num
        print(f"100 divided by {num} is {result}")
    except ZeroDivisionError:
        print(f"Cannot divide by zero when num = {num}. Skipping...")

```

```

PS C:\today> c;; cd 'c:\today'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\ADHARSH\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '54730' '--' 'c:\today\seven-4.py'
100 divided by 10 is 10.0
100 divided by 5 is 20.0
Cannot divide by zero when num = 0. Skipping...
100 divided by 2 is 50.0

```

## OBSERVATION:

**There is no Error Handling.**

**Poor User Feedback:** No indication of what went wrong or which value caused the issue.

**Try Except block added:** isolates risky operation and catches the specific error.

**Improved Feedback:** Prints a clear message when an error occurs, aiding debugging and user understanding.

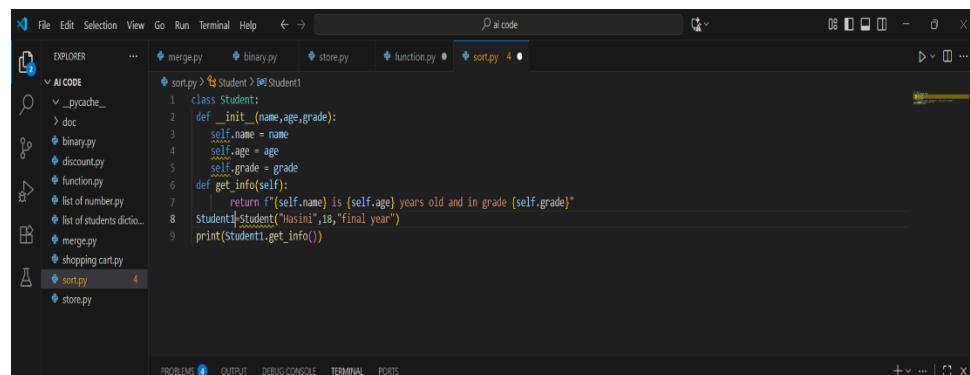
### Task Description #5:

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

### Expected Outcome #5:

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

## BUGGY CODE:



The screenshot shows the VS Code interface with the 'AI CODE' extension active. The Explorer sidebar shows files like merge.py, binary.py, store.py, and function.py. The sort.py file is open in the editor, containing the following code:

```
sort.py > Student > Student
1 class Student:
2     def __init__(name,age,grade):
3         self.name = name
4         self.age = age
5         self.grade = grade
6     def get_info(self):
7         return f"{self.name} is {self.age} years old and in grade {self.grade}"
8 Student = Student("Hasini",18,"final year")
9 print(Student.get_info())
```

The code has several issues: it uses lowercase variable names ('name', 'age', 'grade') instead of 'self.name', 'self.age', and 'self.grade'; it uses 'Student' as both a class name and an instance variable; and it uses 'Student' as a function name.



The terminal window shows the following stack trace and error message:

```
Traceback (most recent call last):
File "C:\Users\ASUS\OneDrive\Desktop\ai code\sort.py", line 1, in <module>
    class Students:
          ^
SyntaxError: invalid syntax
PS C:\Users\ASUS\OneDrive\Desktop\ai code>
```

The terminal also shows the command PS C:\Users\ASUS\OneDrive\Desktop\ai code> at the bottom.

The screenshot shows a Windows desktop environment with VS Code open. The left sidebar displays a file tree with various Python files and a database file. The main editor area has tabs for multiple files, with 'seven-5.py' currently active. The code in 'seven-5.py' is as follows:

```
seven-5.py ...
1 class Student:
2     def __init__(self, name, age, grade):
3         self.name = name
4         self.age = age
5         self.grade = grade # Renamed to match usage in get_info
6
7     def get_info(self):
8         return f'{self.name} is {self.age} years old and in grade {self.grade}'
9 student1 = Student("Anikitha", 20, "second Year")
10 print(student1.get_info())
11
```

The terminal below shows the execution of the script and an error message:

```
PS C:\today> c; cd 'c:\today'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\ADHARSH\vscode\extensions\ms-python.python.debug-2025.10.0-win32-x64\bundledLibs\debugpy\launcher' '49730' ... 'c:\today\seven-5.py'
TypeError: Student.__init__() takes 3 positional arguments but 4 were given
PS C:\today> c; cd 'c:\today'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\ADHARSH\vscode\extensions\ms-python.python.debug-2025.10.0-win32-x64\bundledLibs\debugpy\launcher' '49748' ... 'c:\today\seven-5.py'
Adharsh is 21 years old and in grade Final Year
PS C:\today> c; cd 'c:\today'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\ADHARSH\vscode\extensions\ms-python.python.debug-2025.10.0-win32-x64\bundledLibs\debugpy\launcher' '49774' ... 'c:\today\seven-5.py'
Anikitha is 20 years old and in grade second Year
○ PS C:\today>
```

A floating panel on the right contains AI-generated code for a cursor operation and a note about missing indentation. A status bar at the bottom indicates 'Chat quota reached'.

## OBSERVATION:

- Missing self in `__init__` parameters: Python requires `self` as the first argument in instance methods to refer to the object itself.
  - `self` is the reference to the current instance—required in all instance methods.

