

AI Assisted Coding ASSIGNMENT – 7

Roll.no : 2303a52443

M .Siddhartha

B-38

The screenshot shows the Microsoft Visual Studio Code interface. The left sidebar has icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and various extensions. The Explorer sidebar shows files like AI Ass 1-2.py through AI Ass 7-2.py. The main editor window displays the following Python code:

```
1 #TASK 1: Fixing Syntax Errors
2
3 def add(a, b):
4     return a + b
5
6 print(add(5, 3))
7
```

The terminal at the bottom shows the command `python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"` and the resulting error:

```
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 ':'
```

This screenshot shows the Microsoft Visual Studio Code interface again, with the Explorer sidebar and main editor window identical to the previous one. However, the terminal output has changed:

```
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>
```

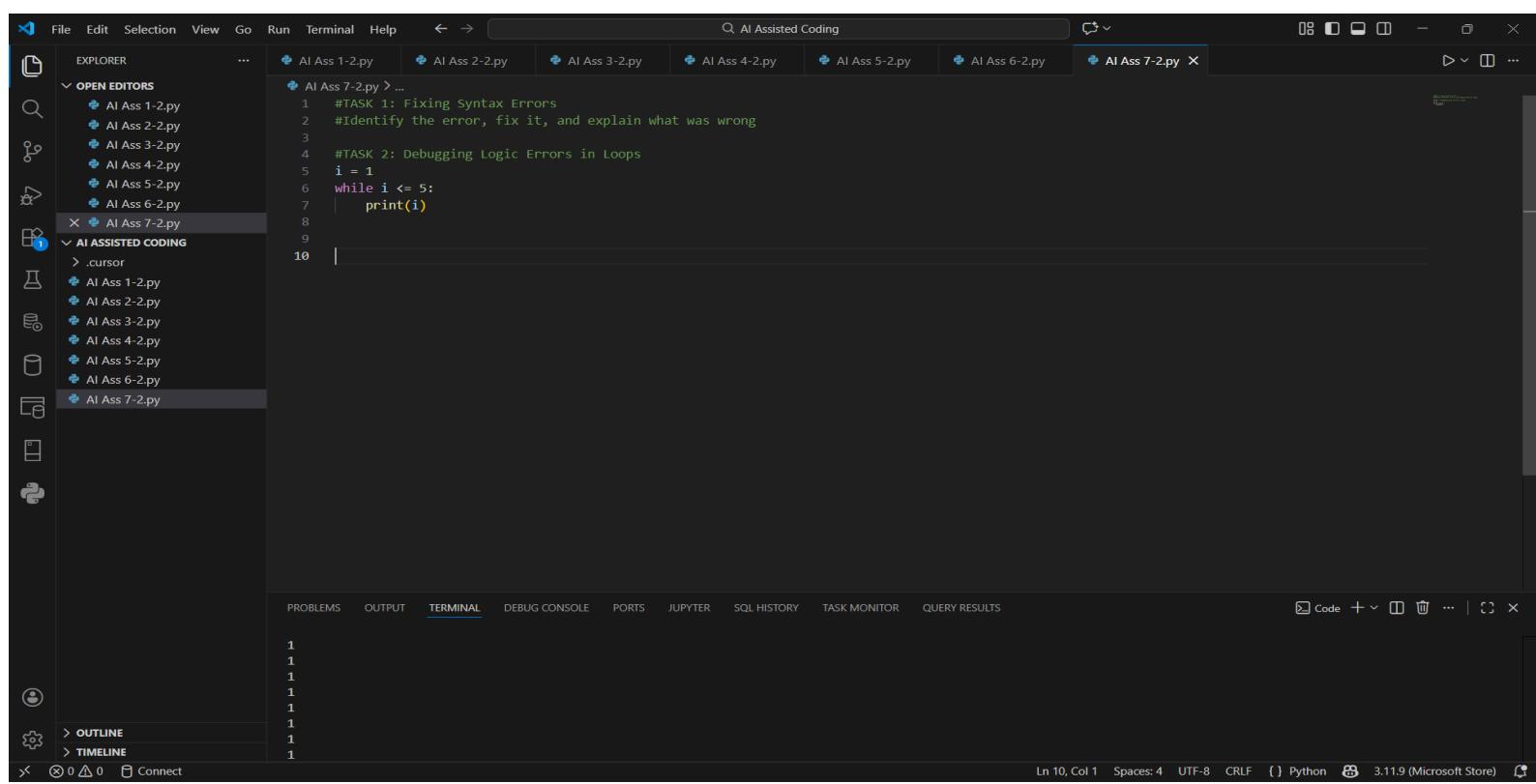
The number 8 likely represents the line number where the code was executed successfully.

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.



The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The code editor displays a file named "AI Ass 7-2.py" with the following content:

```
#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)
```

The terminal tab is active, showing the output of the script:

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

The status bar at the bottom right indicates the file is Python, has 3.11.9 (Microsoft Store) installed, and shows the current file path as "3.11.9 (Microsoft Store)".

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The title bar reads "AI Assisted Coding". The left sidebar has sections for "EXPLORER", "OPEN EDITORS" (listing files like "AI Ass 1-2.py" through "AI Ass 7-2.py"), "AI ASSISTED CODING" (with a file "AI Ass 7-2.py" selected), "OUTLINE", and "TIMELINE". The main editor area displays Python code for Task 2:

```
1 #TASK 1: Fixing Syntax Errors
2 #Identify the error, fix it, and explain what was wrong
3
4 #TASK 2: Debugging Logic Errors in Loops
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6 i = 1
7 while i <= 5:
8     print(i)
9     i += 1
10
11
12
```

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

```
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
```

Bottom status bar: Ln 12, Col 1 | Spaces: 4 | UTF-8 | CRLF | {} Python | 3.11.9 (Microsoft Store)

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.

The screenshot shows the Visual Studio Code interface with the title bar "AI Assisted Coding". The left sidebar has sections for "EXPLORER", "OPEN EDITORS" (listing files like AI Ass 1-2.py through AI Ass 7-2.py), and "AI ASSISTED CODING" (with a dropdown menu). The main editor area contains a Python script "AI Ass 7-2.py" with code related to handling division by zero. The terminal at the bottom shows the command "python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"" and its output, which includes a traceback and the message "Division by zero is not allowed".

```
1 #TASK 1: Fixing Syntax Errors
2 #Identify the error, fix it, and explain what was wrong
3
4 #TASK 2: Debugging Logic Errors in Loops
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6
7 #TASK 3: Handling Runtime Errors (Division by Zero)
8 def divide(a, b):
9     return a / b
10
11 print(divide(10, 0))
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS

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
 ~~~^~~~

Ln 11, Col 21 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

This screenshot shows the same VS Code environment after the user has added a try-except block to handle the division by zero error. The terminal output now shows the corrected behavior: "Division by zero is not allowed".

```
1 #TASK 1: Fixing Syntax Errors
2 #Identify the error, fix it, and explain what was wrong
3
4 #TASK 2: Debugging Logic Errors in Loops
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6
7 #TASK 3: Handling Runtime Errors (Division by Zero)
8 #Identify the error and modify the code using try-except to handle it safely.
9 def divide(a, b):
10     try:
11         return a / b
12     except ZeroDivisionError:
13         return "Division by zero is not allowed"
14
15 print(divide(10, 0))
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS

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>

Ln 15, Col 21 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

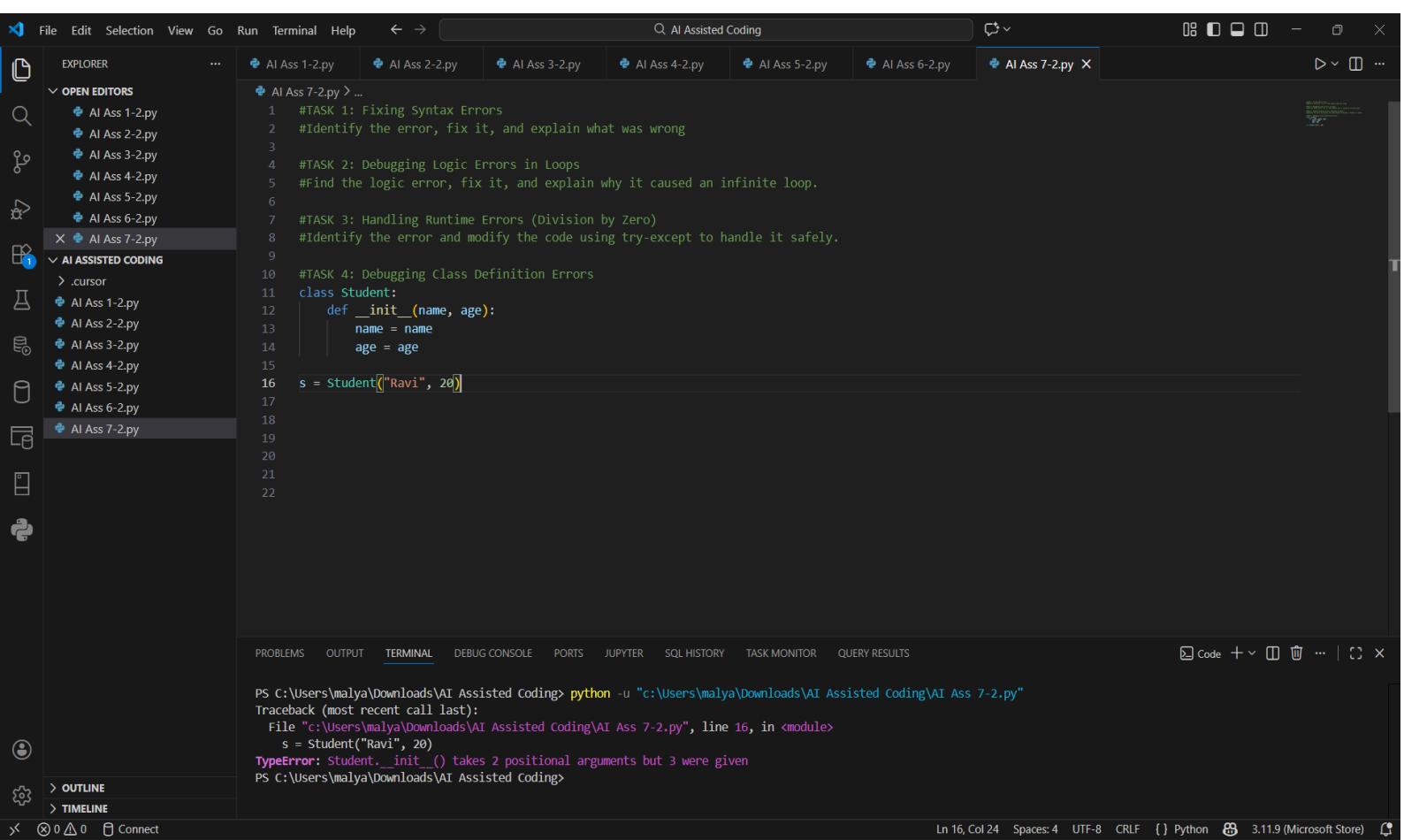
Task 3: Handling Runtime Errors (Division by Zero)

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.



The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The left sidebar has icons for Explorer, Search, Open Editors, and AI Assisted Coding. The main area shows a list of files under 'OPEN EDITORS' and the 'AI ASSISTED CODING' section, with 'AI Ass 7-2.py' currently selected. The code editor displays Python code for task 3, which includes a class definition for 'Student' and an instantiation line that causes a `TypeError`. The terminal at the bottom shows the command `python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"` and the resulting traceback indicating a `TypeError` due to incorrect arguments for the `__init__` method. The status bar at the bottom right shows the file is at line 16, column 24, with 3.11.9 (Microsoft Store) Python installed.

```
#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)
```

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar labeled 'Q AI Assisted Coding'. The left sidebar has sections for Explorer, Open Editors, and AI Assisted Coding, with 'AI Ass 7-2.py' currently selected. The main editor area displays Python code related to class definitions and debugging. Below the editor is a terminal window showing command-line output for running the script. The bottom status bar shows file details like Ln 18, Col 21, and file statistics.

```
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6
7 #TASK 3: Handling Runtime Errors (Division by Zero)
8 #Identify the error and modify the code using try-except to handle it safely.
9
10 #TASK 4: Debugging Class Definition Errors
11 #Identify the issue in the constructor and correct the class definition
12 class Student:
13     def __init__(self, name, age):
14         self.name = name
15         self.age = age
16
17 s = Student("Ravi", 20)
18 print(s.name, s.age)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS JUPYTER SQL HISTORY TASK MONITOR QUERY RESULTS

```
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>
```

Ln 18, Col 21 Spaces: 4 UTF-8 CRLF { } Python 3.11.9 (Microsoft Store)

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.

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q AI Assisted Coding
- Sidebar:** Explorer, Open Editors, AI Assisted Coding (with tasks 1-7 listed), Outline, Timeline.
- Editor:** The main editor window displays Python code for task 7. The code includes comments for tasks 1 through 5, followed by task 7's code:

```
1 #TASK 1: Fixing Syntax Errors
2 #Identify the error, fix it, and explain what was wrong
3
4 #TASK 2: Debugging Logic Errors in Loops
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6
7 #TASK 3: Handling Runtime Errors (Division by Zero)
8 #Identify the error and modify the code using try-except to handle it safely.
9
10 #TASK 4: Debugging Class Definition Errors
11 #Identify the issue in the constructor and correct the class definition
12
13 #TASK 5: Resolving Index Errors in Lists
14 nums = [10, 20, 30]
15 print(nums[5])
16 |
```
- Terminal:** The terminal shows the command `python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"` being run, which results in an `IndexError: list index out of range`.
- Status Bar:** PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, PORTS, JUPYTER, SQL HISTORY, TASK MONITOR, QUERY RESULTS, Code +, etc.
- Bottom:** Ln 16, Col 1, Spaces: 4, UTF-8, CRLF, Python 3.11.9 (Microsoft Store).

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q AI Assisted Coding
- Sidebar:** Explorer, Open Editors, AI Assisted Coding (with tasks 1-7 listed), Outline, Timeline.
- Editor:** The main editor window displays Python code for task 7, now including error handling:

```
1 #TASK 1: Fixing Syntax Errors
2 #Identify the error, fix it, and explain what was wrong
3
4 #TASK 2: Debugging Logic Errors in Loops
5 #Find the logic error, fix it, and explain why it caused an infinite loop.
6
7 #TASK 3: Handling Runtime Errors (Division by Zero)
8 #Identify the error and modify the code using try-except to handle it safely.
9
10 #TASK 4: Debugging Class Definition Errors
11 #Identify the issue in the constructor and correct the class definition
12
13 #TASK 5: Resolving Index Errors in Lists
14 #Identify the problem and modify the code to safely access list elements
15 nums = [10, 20, 30]
16
17 try:
18     print(nums[5])
19 except IndexError:
20     print("Index out of range")
21
22
23
24
25
26
27 |
```
- Terminal:** The terminal shows the command `python -u "c:\Users\malya\Downloads\AI Assisted Coding\AI Ass 7-2.py"` being run, which results in the message `Index out of range`.
- Status Bar:** PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, PORTS, JUPYTER, SQL HISTORY, TASK MONITOR, QUERY RESULTS, Code +, etc.
- Bottom:** Ln 27, Col 1, Spaces: 4, UTF-8, CRLF, Python 3.11.9 (Microsoft Store).

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.