

AI ASSISTED CODING - ASSIGNMENT - 7

NAME : R.RONITH REDDY
HTNO : 2303A52280

TASK - 1 : FIXING SYNTAX ERRORS SCENARIO

PROMPT : DETECT THE SYNTAX ERRORS

The screenshot shows the Visual Studio Code interface. The left sidebar has icons for Run, Debug, Problems, and Breakpoints. The main editor window displays the following Python code:

```
1 # detect the syntax error in the following code
2 def multiply(a, b):
3     return a * b
4 multiply(2, 3)
```

The Problems view at the bottom shows several errors:

- Line 1: zsh: command not found: Radius = 9
- Line 2: zsh: command not found: Radius
- Line 3: zsh: command not found: Radius = 9
- Line 4: zsh: command not found: Radius
- Line 5: zsh: command not found: Radius
- Line 6: zsh: command not found: Radius = 9
- Line 7: zsh: command not found: Radius
- Line 8: zsh: command not found: Radius
- Line 9: zsh: command not found: Radius
- Line 10: zsh: command not found: Radius

A context menu is open on the right side of the editor, with the "Build with agent mode" option highlighted. A tooltip says "AI responses may be inaccurate." The status bar at the bottom indicates the code is in Python 3.13.9.

This screenshot is identical to the one above, showing the same Python code and error messages in the Problems view. The context menu on the right also shows the "Build with agent mode" option. The status bar at the bottom indicates the code is in Python 3.13.9.

EXPLANATION:

THE ERROR HAPPENED BECAUSE RETURN WAS WRITTEN OUTSIDE THE FUNCTION, WHICH PYTHON DOESN'T ALLOW. THERE WAS ALSO A SMALL INDENTATION MISTAKE. ONCE THESE WERE FIXED AND THE CODE WAS PROPERLY STRUCTURED, THE PROGRAM WORKED AS EXPECTED.

TASK - 2 : DEBUGGING LOGIC ERRORS IN LOOPS

PROMPT : FIX THE LOOP LOGIC

```
# Generate a Python function that prints.py
def count_down(n):
    n -= 1
    print("Blastoff!")
```

The code contains two errors:

- A syntax error in the print statement: "print("Blastoff!")". It should be "print("Blastoff!")".
- A logical error in the loop: "return n". This should be "return count_down(n)" to correctly implement a recursive countdown.

The terminal shows the following output:

```
ronithreddyrevuri@roniths-MacBook-Air ~ % 8,9
zsh: command not found: 8,9
ronithreddyrevuri@roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py"
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py", line 6
  1 = int(input("Enter second number: "))
  ^
IndentationError: unexpected indent
ronithreddyrevuri@roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py"
Enter first number: 9
Enter second number: 18
27
8,9ronithreddyrevuri@roniths-MacBook-Air ~ %
```

```
# Fix the loop logic
def count_down(n):
    while n > 0:
        print(n)
        n -= 1
    print("Blastoff!")
a = int(input("Enter a number: "))
print(a)
count_down(a)
```

The code has been corrected to fix the loop logic. The terminal shows the following output:

```
Enter a number: 12
12
11
10
9
8
7
6
5
4
3
2
1
Blastoff!
ronithreddyrevuri@roniths-MacBook-Air ~ %
```

EXPLANATION:

THE LOOP WAS RUNNING INFINITELY BECAUSE THE VALUE WAS INCREASING INSTEAD OF DECREASING. SINCE THE CONDITION NEVER BECAME FALSE, THE LOOP NEVER STOPPED. BY DECREMENTING THE VALUE CORRECTLY, THE LOOP NOW ENDS AS EXPECTED.

TASK - 3 : HANDLING RUNTIME ERRORS (DIVISION BY ZERO)

PROMPT : IDENTIFY THE RUNTIME ERROR AND FIX IT SO THAT THE FUNCTION EXECUTES SAFELY

The screenshot shows the VS Code interface with a dark theme. On the left is the sidebar with icons for Run and Debug, Terminal, Problems, and others. The main area shows a code editor with the following Python code:

```
# Identify the runtime error and fix it so that the function executes safely
def divide_numbers(a, b):
    if b == 0:
        print("Error: Cannot divide by zero")
    return None
    return a / b

print(divide_numbers(10, 2)) # Output: 5.0
```

Below the code editor is a terminal window showing the output of running the script:

```
print(divide_numbers(10, 0)) # This will raise a ZeroDivisionError
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py", line 3, in divide_numbers
    return a / b
ZeroDivisionError: division by zero
● ronithreddyrevuri@Roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py"
Error: Cannot divide by zero
● ronithreddyrevuri@Roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py"
5.0
● ronithreddyrevuri@Roniths-MacBook-Air ~ %
```

The Problems panel at the bottom shows one error: "ZeroDivisionError: division by zero".

The screenshot shows the VS Code interface with a dark theme. The sidebar and code editor are identical to the previous screenshot. The terminal window now shows the corrected code and its execution:

```
# Identify the runtime error and fix it so that the function executes safely
def divide_numbers(a, b):
    if b == 0:
        return "Error: Division by zero is not allowed."
    return a / b
```

```
Enter a number: 12
12
11
10
9
8
7
6
5
4
3
2
1
Blastoff!
● ronithreddyrevuri@Roniths-MacBook-Air ~ %
```

The terminal output shows the user entering numbers from 12 down to 1, followed by the word "Blastoff!". The script correctly handles the division by zero case for the value 0, returning the error message "Error: Division by zero is not allowed." for all other inputs.

EXPLANATION :

THE ERROR OCCURRED BECAUSE DIVISION BY ZERO IS NOT PERMITTED IN PYTHON. BY CHECKING THE DIVISOR BEFORE PERFORMING THE OPERATION, THE FUNCTION AVOIDS CRASHING AND EXECUTES SAFELY.

TASK - 4 : DEBUGGING CLASS DEFINITION ERRORS

PROMPT :

IDENTIFY THE ISSUE IN THE `__INIT__` CONSTRUCTOR AND CORRECT THE CLASS DEFINITION

```
1 # Identify the runtime error and fix it so that the function executes safely
2 def divide_numbers(a, b):
3     if b == 0:
4         print("Error: Cannot divide by zero")
5     return None
6
7
8 print(divide_numbers(10, 0))
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
Traceback (most recent call last):
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py", line 4, in <module>
    print(divide_numbers(10, 0)) # This will raise a ZeroDivisionError
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py", line 3, in divide_numbers
    return a / b
ZeroDivisionError: division by zero
ronithreddyrevuri@Roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6/# Generate a Python function that prints.py"
Error: Cannot divide by zero
None
```

Ln 8, Col 27 Spaces: 4 UTF-8 LF Python Python 3.13.9

EXPLANATION :

THE CONSTRUCTOR WAS MISSING THE `SELF` PARAMETER, WHICH IS REQUIRED TO ACCESS OBJECT VARIABLES. ADDING `SELF` FIXES THE CLASS DEFINITION AND ALLOWS THE OBJECT TO STORE VALUES PROPERLY.

TASK - 5 : RESOLVING INDEX ERRORS IN LISTS

PROMPT :

IDENTIFY AND FIX THE INDEX ERROR CAUSED BY ACCESSING AN OUT-OF-RANGE LIST INDEX

The screenshot shows the Visual Studio Code interface. On the left is the sidebar with icons for Run, Debug, Problems, Output, and Terminal. The main area is a code editor with the following Python code:

```
## Identify and fix the index error caused by accessing an out-of-range list index
numbers = [1, 2, 3, 4, 5]
print(numbers[5])
```

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

```
ronithreddyrevuri@Roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingny - 6/# Generate a Python function that prints.py"
Traceback (most recent call last):
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingny - 6/# Generate a Python function that prints.py", line 3, in <module>
    print(numbers[5])
IndexError: list index out of range
```

A tooltip on the right says "Build with agent mode. AI responses may be inaccurate."

This screenshot shows the same VS Code environment after the code has been fixed. The code in the editor now includes a comment explaining the error:

```
## Identify and fix the index error caused by accessing an out-of-range list index
numbers = [1, 2, 3, 4, 5]
# This will cause an IndexError because the index is out of range
print(numbers[5])
```

The terminal output now shows a different error message:

```
ronithreddyrevuri@Roniths-MacBook-Air ~ % /usr/local/bin/python3 "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingny - 6/# Generate a Python function that prints.py"
Traceback (most recent call last):
File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingny - 6/# Generate a Python function that prints.py", line 3, in <module>
    print(numbers[5])
IndexError: list index out of range
```

A tooltip on the right says "Build with agent mode. AI responses may be inaccurate."

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface in dark mode. On the left is the sidebar with icons for Run and Debug, Terminal, Problems, Output, and more. The main area is a code editor with the following Python code:

```
# Generate a Python function that prints.py
1  ## Identify and fix the index error caused by accessing an out-of-range list index
2  numbers = [1, 2, 3, 4, 5]
3  print(numbers[4]) # This will cause an IndexError because the index is out of range
```

An AI assistance panel on the right suggests fixes:

- Build with agent mode.
- AI responses may be inaccurate.

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

```
zsh
zsh
IndexError: list index out of range
ronithreddyrevuri@Roniths-MacBook-Air:~/Documents/AI Asstd Coding/assingy - 6$ python3 prints.py
Traceback (most recent call last):
  File "/Users/ronithreddyrevuri/Documents/AI Asstd Coding/assingy - 6", line 3, in <module>
    print(numbers[5]) # This will cause an IndexError because the index is out of range
IndexError: list index out of range
ronithreddyrevuri@Roniths-MacBook-Air:~/Documents/AI Asstd Coding/assingy - 6$ python3 prints.py
5
ronithreddyrevuri@Roniths-MacBook-Air:~/Documents/AI Asstd Coding/assingy - 6$
```

Bottom status bar: Ln 3, Col 16 Spaces: 4 UTF-8 LF {} Python Python 3.13.9

EXPLANATION :

THE ERROR OCCURRED BECAUSE THE PROGRAM TRIED TO ACCESS A LIST INDEX THAT DOES NOT EXIST. BY ACCESSING A VALID INDEX OR CHECKING THE LIST LENGTH, THE PROGRAM RUNS WITHOUT CRASHING.

CONCLUSION :

THIS LAB SHOWS HOW AI TOOLS CAN HELP IDENTIFY AND FIX SYNTAX, LOGIC, AND RUNTIME ERRORS IN PYTHON PROGRAMS IN A CLEAR AND SYSTEMATIC WAY.

