

AI Assisted Coding Assignment- 7.2

RUDROJU RUPA SRI

2303A51918

BATCH-30

Task 1 – Runtime Error Due to Invalid Input Type

- A Python program accepts user input and performs arithmetic operations. However, the program throws a runtime error because the input is treated as a string instead of a numeric type.

Example (Buggy Code):

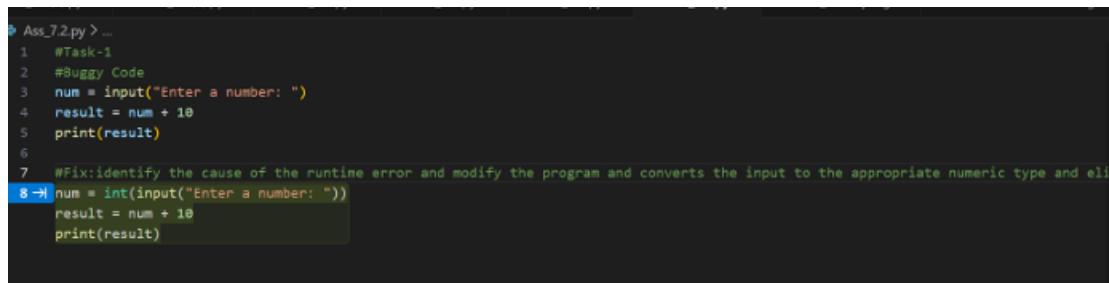
```
num = input("Enter a number: ")
result = num + 10
print(result)
```

- Task:

Use AI tools to identify the cause of the runtime error and modify the program so it executes correctly.

Expected Output -1:

- AI converts the input to the appropriate numeric type and eliminates the runtime error.



```
Ass_7.2.py > ...
1  #Task-1
2  #Buggy Code
3  num = input("Enter a number: ")
4  result = num + 10
5  print(result)
6
7  #Fix:identify the cause of the runtime error and modify the program and converts the input to the appropriate numeric type and eli
8 → num = int(input("Enter a number: "))
    result = num + 10
    print(result)
```

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. In the top left, there are tabs for 'ASS-6.5.py', 'ASS-7.5.py', 'ASS-7.2.py' (which is currently active), and 'Untitled-1'. The main area displays Python code for a task involving string concatenation and infinite recursion. The terminal below shows the execution of the script and its output. On the right side, there's a 'CHAT' sidebar with a list of sessions related to fixing bugs, and a 'SESSIONS' sidebar showing recent interactions with AI. A message in the AI sidebar indicates a quota has been reached. The bottom of the screen shows various status icons and settings.

```
1 #Task-1
2 #Buggy Code
3 #num = input("Enter a number: ")
4 #result = num + 10
5 #print(result)
6 #Fix: identify the cause of the runtime error and modify the program
7 num = int(input("Enter a number: "))
8 result = num + 10
9 print(result)
```

Task 2 – Incorrect Function Return Value

A function is designed to calculate the square of a number, but it does not return the computed result properly.

Example (Buggy Code):

```
def square(n):
    result = n * n
```

Task:

Use AI assistance to analyze the function and ensure the correct value is returned.

Expected Output -2:

AI fixes the missing return statement and the function returns the correct output.

The screenshot shows the Visual Studio Code (VS Code) interface with a dark theme. At the top, there are tabs for 'ASS-6.5.py', 'ASS-7.5.py', 'ASS-7.2.py' (which is currently active), and 'Untitled-1'. Below the tabs is a code editor with the following Python code:

```
10
11  #Task-2
12  #Buggy Code
13  def square(n):
14  | result = n * n
15  #Fix: Fixes the missing return statement and the function returns
16  def square(n):
17  | result = n * n
18  | return result
19  print(square(5))
```

On the right side of the interface, there is a 'SESSIONS' panel showing three recent sessions:

- Fixing string and list concatenation bug - Completed in 24s. (1 day ago)
- Fixing infinite recursion in countdown func... - Completed in 41s. (1 day ago)
- Fixing infinite recursion in countdown func... - Completed in 34s. (1 day ago)

Below the sessions is a 'MORE (3)' link.

In the center, there is a 'TERMINAL' tab open, showing the command line prompt 'PS C:\Users\HP\Desktop\AI>'. The terminal also displays the message 'powershell'.

At the bottom of the interface, there is a status bar with the following information: 'Ln 19, Col 17', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', 'Chat quota reached', 'Python 3.13 (64-bit)', '(i) Go Live', and a bell icon.

Task 3 – IndexError in List Traversal

A Python program iterates over a list using incorrect index limits, causing an IndexError.

Example (Buggy Code):

```
numbers = [10, 20, 30]
for i in range(0, len(numbers)+1):
    print(numbers[i])
```

Task:

Use AI to identify the incorrect loop boundary and correct the iteration logic.

Expected Output -3:

AI fixes the loop condition and prevents out-of-range list access.

The screenshot shows the Visual Studio Code (VS Code) interface. The top bar has tabs for ASS-6.5.py, ASS-7.5.py, ASS-7.2.py, and Untitled-1. The main area displays Python code:

```
20
21 #Task-3
22 #Buggy Code
23 numbers = [10, 20, 30]
24 #for i in range(0, len(numbers)+1):
25 # print(numbers[i])
26 #Fix: Fixes the loop condition and prevents out-of-range list access
27 numbers = [10, 20, 30]
28 for i in range(0, len(numbers)):
29     print(numbers[i])
```

The terminal below shows the output:

```
10
20
30
PS C:\Users\HP\Desktop\AI>
```

The right sidebar includes a Chat section with sessions for fixing bugs, a Sessions list, and an AI-powered code exploration feature.

Task 4 – Uninitialized Variable Usage

A program uses a variable in a calculation before assigning it any value.

Example (Buggy Code):

if True:

pass

print(total)

Task:

Use AI tools to detect the uninitialized variable and correct the program.

Expected Output -4:

AI initializes the variable correctly before it is used

The screenshot shows the Visual Studio Code interface. In the top left, there are tabs for 'ASS-6.5.py', 'ASS-7.5.py', 'ASS-7.2.py' (which is currently active), and 'Untitled-1'. The main code editor window displays the following Python code:

```
30
31 #Task-4
32 #Buggy Code
33 #if True:
34 # pass
35 #print(total)
36 #Fix:Initializes the variable correctly before it is used
37 total = 0
38 if True:
39 | total = 100
40 print(total)
```

Below the code editor are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected), and 'PORTS'. The terminal window shows the command 'PS C:\Users\HP\Desktop\AI>' followed by the output '100'. At the bottom of the terminal, there are status indicators: 'Ln 31, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', 'Chat quota reached', 'Python 3.13 (64-bit)', '(i) Go Live', and a 'Help' icon.

In the top right corner, there is a 'SESSIONS' panel with three items listed under 'SESSIONS':

- Fixing string and list concatenation bug
Completed in 24s. 1 day ago
- Fixing infinite recursion in countdown func...
Completed in 41s. 1 day ago
- Fixing infinite recursion in countdown func...
Completed in 34s. 1 day ago

Below the sessions, there is a 'MORE (3)' link. To the right of the sessions, there is a 'CHAT' section with a message: 'You've reached the limit for chat mes...' and a 'Upgrade' button. Below the message, there is a 'ASS-7.2.py' entry with a 'Ask' button and an 'Auto' dropdown. At the very bottom right, there is a small 'Help' icon.

Task 5 – Logical Error in Student Grading System

A grading program assigns incorrect grades due to improper conditional logic.

Example (Buggy Code):

```
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "C"
else:
    grade = "B"
print(grade)
```

Task:

Use AI to analyze the grading conditions and correct the logical flow.

Expected Output -5:

AI corrects the conditional logic so grades are assigned accurately.

The screenshot shows the Visual Studio Code interface. The code editor displays a Python script named `ASS-7.2.py`. The terminal below it shows a PowerShell prompt at `C:\Users\HP\Desktop\AI>`. The top right corner shows a list of recent sessions.

```
41
42 #Task-5
43 #Buggy Code
44 #marks = 85
45 #if marks >= 90:
46 # grade = "A"
47 # elif marks >= 80:
48 #grade = "C"
49 #else:
50 #grade = "B"
51 #print(grade)
52 #fix:Corrects the conditional logic so grades are assigned accurately
53 marks = 85
54 if marks >= 90:
55     grade = "A"
56 elif marks >= 80:
57     grade = "B"
58 else: grade = "C"
59 print(grade)
```

SESSIONS

- Fixing string and list concatenation bug
Completed in 24s. 1 day ago
- Fixing infinite recursion in countdown func...
Completed in 41s. 1 day ago
- Fixing infinite recursion in countdown func...
Completed in 34s. 1 day ago

MORE (3)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

B
PS C:\Users\HP\Desktop\AI>

Ln 59, Col 13 Spaces: 4 UTF-8 CRLF { } Python Chat quota reached Python 3.13 (64-bit) (•) Go Live