

ASSIGNMENT-7.2

NAME: HABEEBA KHANAM

HT.NO:2303A51474

BATCH:29

LAB 7: Error Debugging with AI: Systematic approaches to finding and fixing bugs

TASK DESCRIPTION

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.

CODE:

The screenshot shows the Visual Studio Code interface. The code editor has the following content:

```
1 num = str(input("Enter a number: "))
2 result = num + 10
```

An inline suggestion menu is open at the bottom of the code editor, listing options like 'Go To / Accept' and 'Reject'. The terminal window below shows the following output:

```
num1 = float(input("Enter first number: "))
ValueError: could not convert string to float: '& "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1 721111.py"'
PS C:/Users/Sameera Khan & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/Untitled-1 721111.py"
Enter first number: 5
Enter second number: 2
Enter operation (+, -, *, /): +
Result: 7.0
PS C:/Users/Sameera Khan:
```

The status bar at the bottom indicates the file is saved in Python (Python 3.13.5), the code is 3.13.5 lines long, and it was last modified at 02:52 on 13-02-2026.

The screenshot shows the Visual Studio Code interface. The code editor has the following content:

```
1 #Task 1 - Runtime Error Due to Invalid Input Type
2 num: str = input("Enter a number: ")
3 try:
4     result = int(num) + 10
5     print(result)
6 except ValueError:
7     print("Error: Please enter a valid integer.")
```

The terminal window below shows the following output:

```
15
PS C:/Users/Sameera Khan & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
Enter a number: 5
15
PS C:/Users/Sameera Khan & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
Enter a number: 5
15
PS C:/Users/Sameera Khan:
```

The status bar at the bottom indicates the file is saved in Python (Python 3.13.5), the code is 3.13.5 lines long, and it was last modified at 03:13 on 13-02-2026.

JUSTIFICATION:

- `int(input())` converts the string input into an integer.
- Arithmetic operations can now be performed safely.
- The runtime error is eliminated.

TASK DESCRIPTION

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

CODE:

```

15
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
Enter a number: 5
15
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
Enter a number: 5
15
PS C:\Users\Sameera Khan>

```

```

1
2 #Task 2 - Incorrect Function Return Value
3
4 def square(n):
5     result = n * n
6     return result
7 print(square(5)) # Output: 25

```

```

PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
25
PS C:\Users\Sameera Khan>

```

JUSTIFICATION:

- $n * n$ correctly computes the square.
- return result ensures the value is passed back.

- The function now behaves as expected.

TASK DESCRIPTION

TASK 3 – INDEX ERROR 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.

CODE:

The screenshot shows the Visual Studio Code interface. A code editor window displays the following Python code:

```

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

```

An inline suggestion box is open at the second line, showing the current selection as "for i in range(0, len(numbers)):" with a green highlight. Below the suggestion box is a context menu with options: "✓ Go To / Accept", "Tab", "✗ Reject", "Escape", ">Show Collapsed", "Snooze", "Settings", "Feedback", and "Learn More...".

The bottom of the screen shows the terminal output:

```

PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
25
PS C:\Users\Sameera Khan>

```

The status bar at the bottom indicates the file is saved ("master" icon), the Python extension is active, and the date/time is 13-02-2026.

The screenshot shows the Visual Studio Code interface. A code editor window displays the following Python code:

```

1 # Task 3 - IndexError in List Traversal
2
3 numbers = [10, 20, 30]
4 for i in range(0, len(numbers)):
5     print(numbers[i])

```

The terminal output shows an error:

```

c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
25
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
10
20
30
PS C:\Users\Sameera Khan>

```

The status bar at the bottom indicates the file is saved ("master" icon), the Python extension is active, and the date/time is 13-02-2026.

JUSTIFICATION:

- `range(0, len(numbers))` prevents out-of-range access.

- The enhanced loop avoids index handling entirely.
- The Index Error is fully eliminated.

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

CODE:

The screenshot shows a code editor interface with several tabs at the top: AI - 6.5.py, Untitled-1.py, Untitled-1 hh.py, Untitled-1 656.py, Untitled-1 777.py, Untitled-1 721111.py, and BB AI.py 2. The main editor area contains Python code. A context menu is open over the line 'if True:'. The menu has a title 'Inline Suggestion' and includes options: '✓ Go To / Accept' (selected), 'Tab', '✗ Reject' (disabled), 'Escape', '>Show Collapsed', 'Snooze', 'Settings', 'Feedback', and 'Learn More...'. Below the editor is a 'PROBLEMS' panel showing a single error: 'c:/Users/Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/Desktop/BB AI.py"'. To the right of the editor is a 'Python' panel containing the command 'PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/Desktop/BB AI.py"'. At the bottom of the screen is a taskbar with various icons.

```
c:/Users/Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/Desktop/BB AI.py"
25
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/Desktop/BB AI.py"
10
20
30
PS C:\Users\Sameera Khan>
```

The screenshot shows the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, and a search bar. The left sidebar has icons for file operations like Open, Save, Find, and others. The main editor window displays a Python script with the following code:

```
C: > Users > Sameera Khan > OneDrive > Desktop > BB AI.py > ...
1 # Task 4 - Uninitialized Variable Usage
2
3 if True:
4     total = 100
5     print(total)
```

The terminal below the editor shows the command line history:

```
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
10
20
30
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
100
PS C:\Users\Sameera Khan>
```

The bottom status bar shows the file path as master*, the line and column numbers as Ln 4, Col 16, and the encoding as UTF-8.

JUSTIFICATION:

- Initializing total ensures it exists in memory.
- The program can now safely access and print the variable.
- The runtime error is eliminated.

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.

CODE:

```

File Edit Selection View Go ... ← → Search
C: > Users > Sameera Khan > OneDrive > Desktop > BB AI.py > ...
1 # Task 5 - Logical Error in Student Grading System
2
3
4 marks = 85
5 if marks >= 90:
6     grade = "A"
7 elif marks >= 80:
8     Click to collapse the range
9 else:
10     grade = "B"
11 print(grade)
12

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
25
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
10
20
30
PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"
100
PS C:\Users\Sameera Khan>

```

The screenshot shows the Visual Studio Code interface. The code editor displays the following Python script:

```

1 grade = "B"
2 print(grade)

```

A runtime error message is visible in the status bar at the bottom:

PS C:\Users\Sameera Khan> & "C:/Users/Sameera Khan/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/Sameera Khan/OneDrive/Desktop/BB AI.py"

The terminal tab shows the command run and the output of the script.

The screenshot shows the Visual Studio Code interface with the same Python script as before, but now it runs successfully without errors. The code editor displays:

```

1 # Task 5 - Logical Error in Student Grading System
2 marks = 85
3 if marks >= 90:
4     grade = "A"
5 elif marks >= 80:
6     grade = "B"
7 else:
8     grade = "C"
9 print(grade)

```

The terminal tab shows the command run and the output of the script, which correctly prints "B".

JUSTIFICATION:

- Initializing total ensures it exists in memory.
- The program can now safely access and print the variable.
- The runtime error is eliminated.

