

School of Computer Science and Artificial Intelligence

Lab Assignment # 7.2

Program	: B. Tech (CSE)
Specialization	: -
Course Title	: AI Assisted Coding
Course Code	: 23CS002PC304
Semester	II
Academic Session	: 2025-2026
Name of Student	::R.AKASH REDDY
Enrollment No.	: 2403A51L30
Batch No.	51
Date	: 30/01/26

Submission Starts here

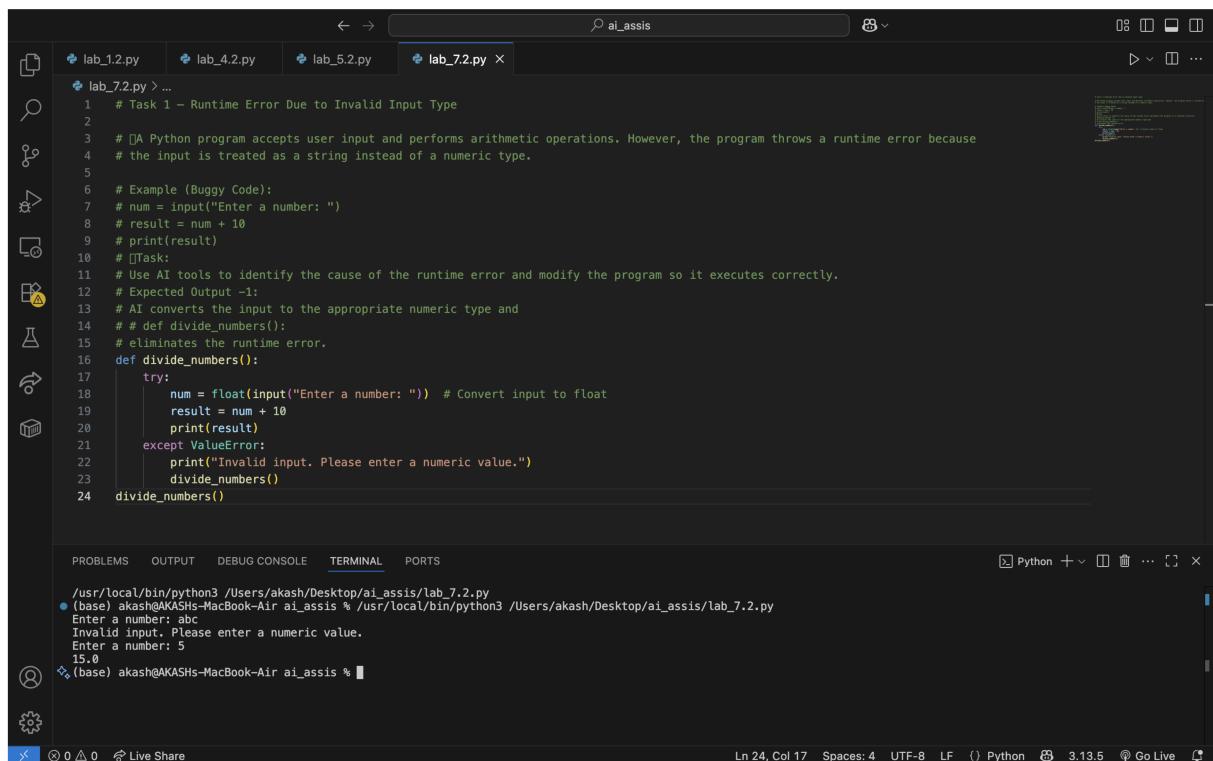
Screenshots:

Task 1 – Runtime Error Due to Invalid Input Type

(Buggy Code):

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

The error occurs because `input()` returns a string and adding an integer to it causes a `TypeError`.



```
lab_7.2.py > ...
1  # Task 1 – Runtime Error Due to Invalid Input Type
2
3  # 🚫 A Python program accepts user input and performs arithmetic operations. However, the program throws a runtime error because
4  # the input is treated as a string instead of a numeric type.
5
6  # Example (Buggy Code):
7  # num = input("Enter a number: ")
8  # result = num + 10
9  # print(result)
10 # [Task:
11 # Use AI tools to identify the cause of the runtime error and modify the program so it executes correctly.
12 # Expected Output -1:
13 # AI converts the input to the appropriate numeric type and
14 # # def divide_numbers():
15 # # eliminates the runtime error.
16 def divide_numbers():
17     try:
18         num = float(input("Enter a number: ")) # Convert input to float
19         result = num + 10
20         print(result)
21     except ValueError:
22         print("Invalid input. Please enter a numeric value.")
23     divide_numbers()
24 divide_numbers()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
/usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
(base) akash@AKASHs-MacBook-Air ai_assis % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
Enter a number: abc
Invalid input. Please enter a numeric value.
Enter a number: 5
15.0
(base) akash@AKASHs-MacBook-Air ai_assis %
```

The issue is fixed by converting the input to a numeric type using `float()` and handling invalid input with `try-except`

Task 2 – Incorrect Function Return Value

(Buggy Code):

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

The function calculates the square but does not return the result, so it returns None. Adding the return statement fixes the problem and gives the correct output.

A screenshot of the Visual Studio Code (VS Code) interface. The top navigation bar shows tabs for 'lab_1.2.py', 'lab_4.2.py', 'lab_5.2.py', and 'lab_7.2.py'. The 'lab_7.2.py' tab is active, displaying the following Python code:

```
24     # divide_numbers()
25
26     # Task 2 – Incorrect Function Return Value
27
28     # A function is designed to calculate the square of a number, but it does not return the computed result properly.
29     # Example (Buggy Code):
30     #     def square(n):
31     #         result = n * n
32     # Task:
33     # Use AI assistance to analyze the function and ensure the correct value is returned.
34     # Expected Output -2:
35     # AI fixes the missing return statement and the function returns the correct output.
36     def square(n):
37         result = n * n
38         return result # Added return statement
39     # Example usage
40     number = 5
41     print(f"The square of {number} is {square(number)}")
```

The bottom right corner of the code editor shows a small AI icon. Below the code editor is the 'TERMINAL' tab, which displays the command-line output of running the script:

```
/usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
(base) akash@AKASHs-MacBook-Air ~ % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
The square of 5 is 25
(base) akash@AKASHs-MacBook-Air ~ %
```

The status bar at the bottom shows 'Ln 41, Col 53' and other terminal details.

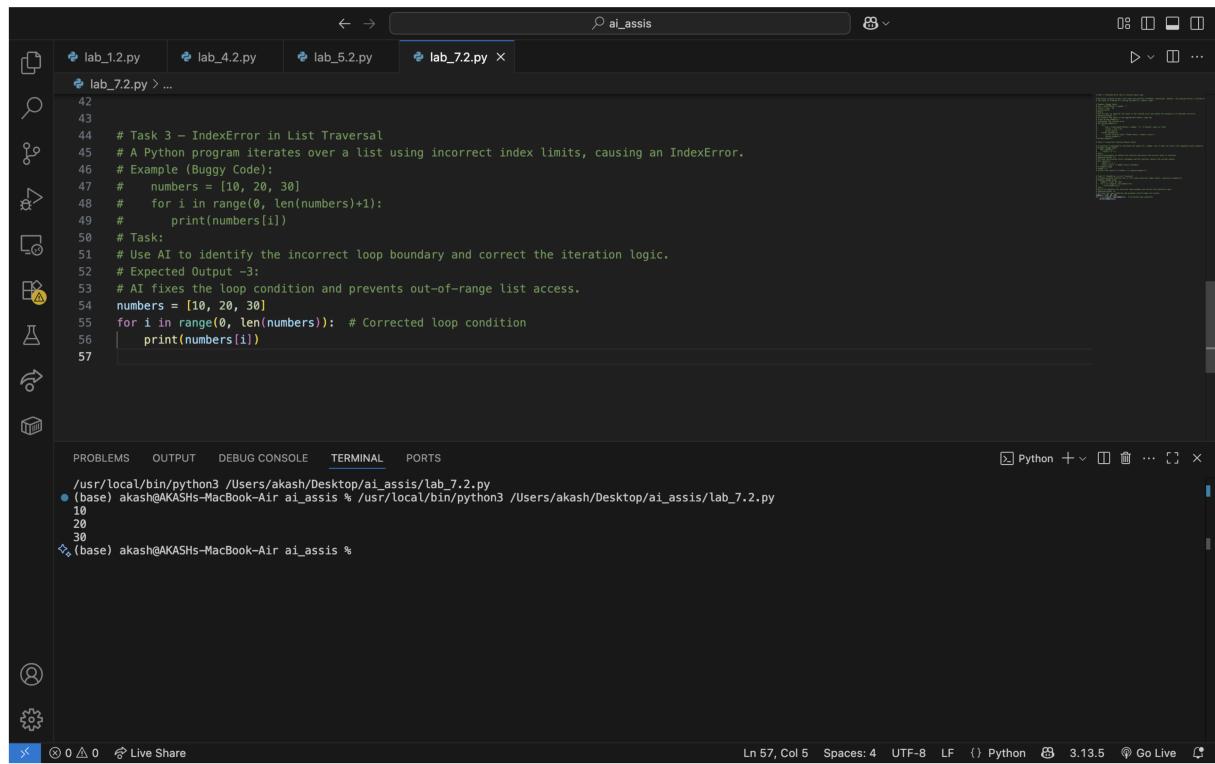
Task 3 – IndexError in List Traversal

(Buggy Code):

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

The loop used `len(numbers)+1`, which tries to access an index outside the list range and causes an `IndexError`.

Correcting the loop to `range(len(numbers))` prevents out-of-range access.



The screenshot shows a dark-themed instance of Visual Studio Code. The left sidebar has icons for file operations like Open, Save, Find, and Copy/Paste. The top bar shows tabs for 'lab_1.2.py', 'lab_4.2.py', 'lab_5.2.py', and 'lab_7.2.py' (which is currently active). The main editor area contains Python code for Task 3:

```

42
43
44 # Task 3 - IndexError in List Traversal
45 # A Python program iterates over a list using incorrect index limits, causing an IndexError.
46 # Example (Buggy Code):
47 #     numbers = [10, 20, 30]
48 #     for i in range(0, len(numbers)+1):
49 #         print(numbers[i])
50 # Task:
51 # Use AI to identify the incorrect loop boundary and correct the iteration logic.
52 # Expected Output -3:
53 # AI fixes the loop condition and prevents out-of-range list access.
54 numbers = [10, 20, 30]
55 for i in range(0, len(numbers)): # Corrected loop condition
56     print(numbers[i])
57

```

The terminal below shows the command `python3 lab_7.2.py` being run, with the output `10\n20\n30`.

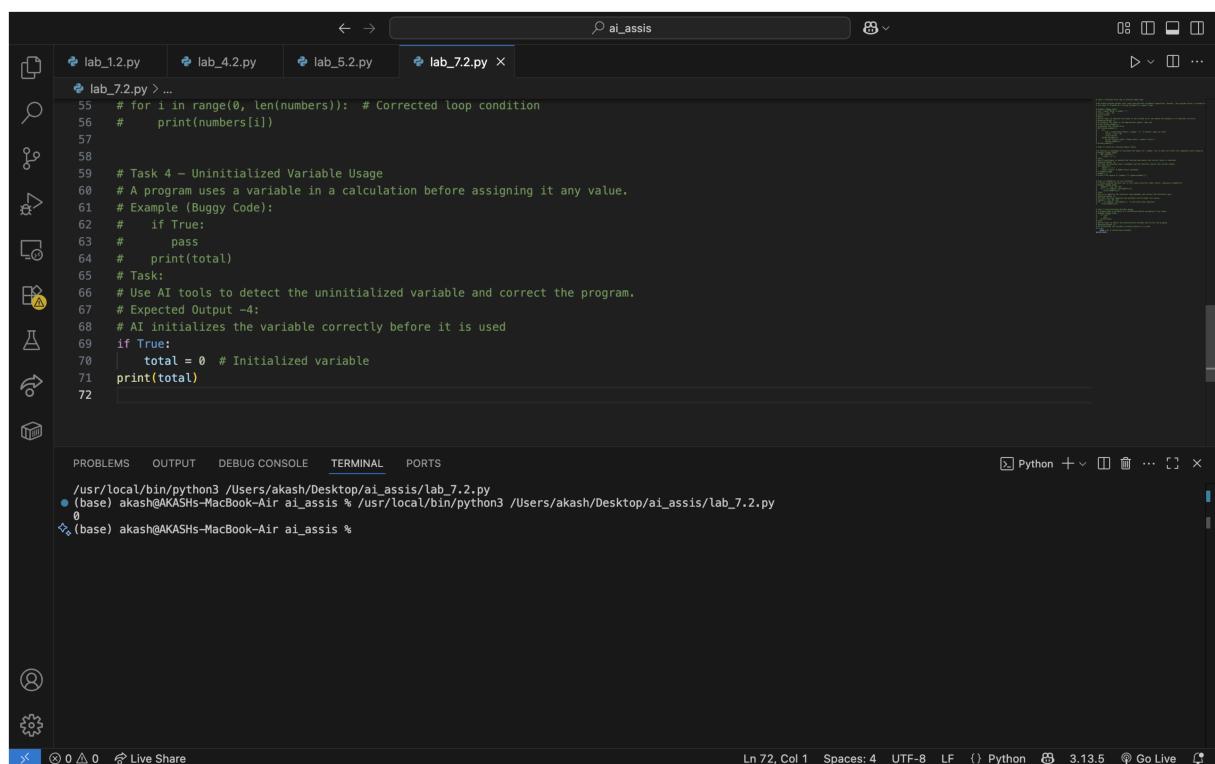
Task 4 – Uninitialized Variable Usage

(Buggy Code):

if True:

 pass print(total)

The variable total was used before assigning any value, causing a NameError. Initializing total = 0 before using it resolves the issue.



The screenshot shows a dark-themed instance of Visual Studio Code. The left sidebar has icons for file operations like Open, Save, Find, and Copy/Paste. The top bar shows tabs for 'lab_1.2.py', 'lab_4.2.py', 'lab_5.2.py', and 'lab_7.2.py' (which is currently active). The main editor area contains Python code for Task 4:

```

55 # for i in range(0, len(numbers)): # Corrected loop condition
56 #     print(numbers[i])
57
58
59 # Task 4 - Uninitialized Variable Usage
60 # A program uses a variable in a calculation before assigning it any value.
61 # Example (Buggy Code):
62 #     if True:
63 #         pass
64 #     print(total)
65 # Task:
66 # Use AI tools to detect the uninitialized variable and correct the program.
67 # Expected Output -4:
68 # AI initializes the variable correctly before it is used
69 if True:
70     total = 0 # Initialized variable
71 print(total)
72

```

The terminal below shows the command `python3 lab_7.2.py` being run, with the output `0`.

Task 5 – Logical Error in Student Grading System

(Buggy Code):

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

The grading conditions were incorrect, assigning the wrong grade for certain marks. Reordering and correcting the conditions ensures grades are assigned accurately.

The screenshot shows a code editor interface with several tabs at the top: lab_1.2.py, lab_4.2.py, lab_5.2.py, and lab_7.2.py (which is the active tab). The code in lab_7.2.py is as follows:

```
1 # print(total)
2
3 # Task 5 – Logical Error in Student Grading System
4 # A grading program assigns incorrect grades due to improper conditional logic.
5 # Example (Buggy Code):
6 # marks = 85
7 # if marks >= 90:
8 #     grade = "A"
9 # elif marks >= 80:
10 #     grade = "C"
11 # else:
12 #     grade = "B"
13 # print(grade)
14
15 # Task:
16 # Use AI to analyze the grading conditions and correct the logical flow.
17 # Expected Output -5:
18 # AI corrects the conditional logic so grades are assigned accurately.
19 marks = 85
20 if marks >= 90:
21     grade = "A"
22 elif marks >= 80:
23     grade = "B"
24 else:
25     grade = "C"
26 print(grade)
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
```

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

```
/usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
(base) akash@AKASHs-MacBook-Air ~ % /usr/local/bin/python3 /Users/akash/Desktop/ai_assis/lab_7.2.py
B
(base) akash@AKASHs-MacBook-Air ~ %
```

The bottom status bar indicates the file is Python, version 3.13.5, and the current line and column are Ln 89, Col 16.