

School of Computer Science and Artificial Intelligence

Lab Assignment -7.2

Program :B. Tech (CSE)
Specialization :AIML
Course Title : AI Assisted Coding
Course Code : 23CS002PC304
Semester : VI
Academic Session : 2025-2026
Name of Student : K. Shailaja
Enrollment No. : 2303A52155
Batch No. : 34

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.

```

1 num = input("Enter a number: ")
2
3 if num.isdigit():
4     num = int(num)
5     result = num + 10
6     print(result)
7 else:
8     print("Invalid input! Please enter a numeric value.")
9

```

I have a Python program that gives a runtime error.

Buggy Code:

```

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

```

Find the error, explain why it happens, and correct the code so it works properly.

Local Shift Tab to plan

Problems Output Debug Console Terminal Ports

PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/3-2/AIAC/bug_fixing.py"
Enter a number: 15
25
PS D:\3-2\AIAC>

+ ... ^ x

powerShell Python

Ctrl+K to generate command

Cursor Tab Ln 9, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.9 64-bit (Microsoft Store) Q

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.

```
Review Next File
❶ bug square of a number.py > ...
1 def square(n):
2     result = n * n
3     return result
4
5 num = input("Enter a number: ")
6 if num.isdigit():
7     num = int(num)
8     result = square(num)
9     print(result)
10 else:
11     print("Invalid input! Please enter a numeric value.")

I have a Python function that is supposed to return the square of a number.

Buggy Code:
def square(n):
    result = n * n

Find the mistake in this function and correct it so that it properly returns the square value.

Auto ▾ Auto ▾
Local ▾ Shift Tab to plan
Problems Output Debug Console Terminal Ports
PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/3-2/AIAC
/bug square of a number.py"
Enter a number: 8
64
PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/3-2/AIAC
/bug square of a number.py"
Enter a number: 10
100
PS D:\3-2\AIAC>
+
... ^ x
powershell
Python
```

Ctrl+K to generate command

Cursor Tab In 7, Col 19 Spaces: 4 UTF-8 CRLF Python

Task Description

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 a code editor interface with a dark theme. On the left, a code editor window displays a Python script named 'list.py' with the following code:

```
element from list.py > ...
1 numbers = [10, 20, 30]
2 for i in range(len(numbers)):
3     print(numbers[i])
4
```

A status bar at the top right indicates 'Review Next File'. To the right of the code editor is a panel containing a task description and some configuration options. The task description reads:

I have a Python program that prints elements from a list using a loop.

Buggy Code:

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

The program throws an IndexError. Find the mistake and correct the loop so it works properly.

Below the task description are several small icons and buttons, including a 'Local' dropdown, a 'Shift' button, and a 'Tab' button labeled 'to plan'.

At the bottom of the screen, a terminal window is open. The tab bar at the top of the terminal says 'Terminal'. The terminal output shows the execution of the script:

```
PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/3-2/AIAC/list.py"
10
20
30
PS D:\3-2\AIAC>
```

On the far right of the terminal window, there are status indicators: 'powershell' (unchecked), 'Python' (checked), 'Cursor Tab', 'Ln 4, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Python'.

Task Description

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

```
variable.py > ...
1 total = 0
2
3 for i in range(1, 6):
4     total = total + i
5
6 print(total)
7
```

I have a Python program where a variable is used before it is initialized.

Buggy Code:

```
if True:
    pass
print(total)
```

Find the error and modify the code so that the variable is properly initialized before use.

Terminal:

```
PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/3-2/AIAC/
variable.py
● 15
○ PS D:\3-2\AIAC>
```

Cursor Tab | In 7, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python

Task Description

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:

The screenshot shows a Visual Studio Code (VS Code) interface with a code editor, terminal, and a floating code review panel.

Code Editor:

```
grades.py > ...
1  marks = 85
2
3  if marks >= 90:
4      grade = "A"
5  elif marks >= 80:
6      grade = "B"
7  else:
8      grade = "C"
9
10 print(grade)
11
```

Code Review Panel:

Review Next File

I have a python grading program, but it assigns wrong grades due to incorrect conditional logic.

Buggy Code:

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

Analyze the logic, find the mistake, and correct the conditions so that proper grades are assigned.

∞ Auto ⌂ Local Shift Tab to plan ↑

Terminal:

Problems Output Debug Console Terminal Ports

```
PS D:\3-2\AIAC> & C:/Users/pashi/AppData/Local/Microsoft/WindowsApps/python3.11.exe d:/3-2/AIAC/
grades.py
B
PS D:\3-2\AIAC>
```

Ctrl+K to generate command

Cursor Tab Ln 11, Col 1 Spaces: 4 UTF-8 CRLF Python