

**Course Name:AI Assistant For Coding**

**Name:Swetha Dixit**

**Batch:33**

**Hall-Ticket:2303A52415**

**Assignment:7.2**

**Task 1:**

**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

The screenshot shows a software interface for AI coding. At the top, there's a menu bar with File, Edit, Selection, View, Go, Run, Terminal, Help, and tabs for task\_2\_7.2.py, Task\_1\_7.2.py, Task\_3\_7.2.py, task\_4\_7.2.py, and task\_5\_7.2.py. Below the tabs, the main area displays Python code:

```

Assignment 4.2 > Task_1_7.2.py > ...
num = input("Enter a number: ")
result = num + 10
print(result)
try:
    num = float(input("Enter a number: "))
    result = num + 10
    print(result)
except ValueError:
    print("Please enter a valid number.")

```

On the right side, there's an AI assistance panel titled "Loop IndexError explanation". It shows a GPT-5 Mini prompt asking, "What is wrong with the logic in this grading system?", followed by the provided corrected code:

```

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

```

Below this, a list of problems is shown:

- The grade branches are swapped: scores >=80 assign "C" and the else assigns "B" — so 80-89 will get "C" instead of "B".
- Also make sure the assignment lines are indented under the if/elif/else (otherwise Python raises IndentationError).

Under "Fixed version:" is the corrected code:

```

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

```

At the bottom, there's a terminal window showing command history:

```

(.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/ai coding/Assignment 4.2/task_3_7.2.py"
10
20
30
(.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/ai coding/Assignment 4.2/task_4_7.2.py"
0
(.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/ai coding/Assignment 4.2/task_5_7.2.py"
B
(.venv) PS C:\Users\intec\AI coding>

```

Below the terminal is a status bar with various icons and text.

## 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**

The screenshot shows a software interface for AI coding. At the top, there's a menu bar with File, Edit, Selection, View, Go, Run, Terminal, Help, and a tab labeled 'Ai coding'. Below the menu, several tabs are open: 'task\_2\_7.2.py', 'Task\_1\_7.2.py', 'Task\_3\_7.2.py', 'task\_4\_7.2.py', and 'task\_5\_7.2.py'. A tooltip 'Review Next File' is visible above the tabs.

In the center, there's a code editor window with the following Python code:

```

1 def square(n):
2     result = n * n
3     return result
4 print(square(5))

```

A tooltip 'Fix the missing return statement.' is shown above the code, with options 'Reject' and 'Accept Ctrl+D'. Below the code editor, there are buttons for Undo Ctrl+N and Keep Ctrl+Shift+Y.

To the right of the code editor, there's a panel titled 'Loop IndexError explanation' with the following text:

12m · GPT-5 Mini

**task\_5\_7.2.py (1-8)** "What is wrong with the logic in this grading system?"

Providing corrected code 9s

```

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

```

Problems:

- The grade branches are swapped: scores >=80 assign "C" and the else assigns "B" — so 80-89 will get "C" instead of "B".
- Also make sure the assignment lines are indented under the if/elif/else (otherwise Python raises IndentationError).

Fixed version:

```

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

```

At the bottom of the interface, there's a terminal window showing command-line history:

```

● (.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/AI coding/Assignment 4.2/task_3_7.2.py"
10
20
30
● (.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/AI coding/Assignment 4.2/task_4_7.2.py"
0
● (.venv) PS C:\Users\intec\AI coding> & "C:\Users\intec\AI coding\.venv\scripts\python.exe" "c:/Users/intec/AI coding/Assignment 4.2/task_5_7.2.py"
B
○ (.venv) PS C:\Users\intec\AI coding>

```

Below the terminal, there are status indicators: Cursor Tab, OVR, Lr 4, Col 1, Spaces: 4, UTF-8, CR/LF, Python, venv (3.14.0), Go Live, and a refresh 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 a code editor interface with multiple tabs open. The active tab is 'Task\_3\_7.2.py' containing the following code:

```

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

```

A tooltip above the code says: "Fix the loop boundary to prevent out-of-range access...". A button labeled "Accept Ctrl+d" is visible. To the right, there's a panel titled "Loop IndexError explanation" with an AI-generated fix:

```

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

```

The interface includes a terminal at the bottom showing command-line history and a status bar at the bottom right.

## 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 a Python code editor interface with several tabs open. The active tab is 'task\_4\_7.2.py' containing the following code:

```

1 total = 0
2 if True:
3     pass
4 print(total)

```

A tooltip from AI coding suggests: "Initialize the variable total before printing it." Below the code, there are buttons for 'Reject' and 'Accept Ctrl+d'. A sidebar on the right titled 'Loop IndexError explanation' discusses a bug in the grading system and provides a corrected version of the code:

```

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

```

The terminal below shows command-line interactions related to the assignment.

## 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 a Python code editor interface with several tabs at the top: task\_2\_7.2.py, Task\_1\_7.2.py, Task\_3\_7.2.py, task\_4\_7.2.py, task\_5\_7.2.py, and task\_5.7.2.py (the active tab). A modal window titled "Loop IndexError explanation" is open, showing a fix for grading logic. The code in the editor is:

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

The modal suggests fixing the grading logic so 80-89 is B and anything low... with options to "Accept Ctrl+d" or "Reject". Below the code, a terminal window shows command history:

```
(.venv) PS C:\Users\intec\Ai coding> & "C:/Users/intec/Ai coding\venv\scripts\python.exe" "c:/users/intec/Ai coding/Assignment 4.2/Task_3_7.2.py"
10
20
30
● (.venv) PS C:\Users\intec\Ai coding> & "C:/Users/intec/Ai coding\venv\scripts\python.exe" "c:/users/intec/Ai coding/Assignment 4.2/task_4_7.2.py"
0
● (.venv) PS C:\Users\intec\Ai coding> & "C:/Users/intec/Ai coding\venv\scripts\python.exe" "c:/users/intec/Ai coding/Assignment 4.2/task_5_7.2.py"
B
○ (.venv) PS C:\Users\intec\Ai coding>
```

The terminal also shows a prompt: "Ctrl+K to generate command".