**Secton2:**

**Problem 1: Robust Temperature Converter**

**1. Understand the problem**

Create a temperature converter that handles user input errors and converts between Celsius and Fahrenheit.

**Inputs**:

* Temperature value (as string → must validate numeric)
* Original unit ('C' or 'F')
* Target unit ('C' or 'F')

**Outputs**:

* Converted temperature with units, or meaningful error messages.

**2. Work by Hand**

**Manual Example**:

* Input: 100, C, F → Output: 212°F
* Input: 32, F, C → Output: 0°C

**3. Pseudocode**:

1. Ask user for temperature input

- Try to convert to float

- If fail: show error and exit

2. Ask for original unit (C/F)

- Convert to uppercase, check if valid

- If invalid: show error and exit

3. Ask for target unit (C/F)

- Convert to uppercase, check if valid

- If same as original: show message and exit

4. Perform conversion:

- If C → F: use F = C \* 9/5 + 32

- If F → C: use C = (F - 32) \* 5/9

5. Print result formatted with 2 decimal places

**4. Convert into python code**

# Temperature converter with error handling

def convert\_temperature():

try:

temp = float(input("Enter temperature value: "))

except ValueError:

print("Invalid temperature! Please enter a numeric value.")

return

original\_unit = input("Enter original unit (C/F): ").strip().upper()

if original\_unit not in ["C", "F"]:

print("Invalid original unit. Please enter 'C' or 'F'.")

return

target\_unit = input("Enter target unit (C/F): ").strip().upper()

if target\_unit not in ["C", "F"]:

print("Invalid target unit. Please enter 'C' or 'F'.")

return

if original\_unit == target\_unit:

print("Original and target units are the same. No conversion needed.")

return

if original\_unit == "C" and target\_unit == "F":

converted = temp \* 9/5 + 32

print(f"{temp:.2f}°C is {converted:.2f}°F")

elif original\_unit == "F" and target\_unit == "C":

converted = (temp - 32) \* 5/9

print(f"{temp:.2f}°F is {converted:.2f}°C")

convert\_temperature()

**5.Tests**:

* Valid: 100 Fto C
* Invalid: 100c

**Problem 2: Simple Quiz System**

**1. Understand the problem**

**Ask multiple-choice questions, collect answers, validate inputs, and calculate score.**

**Quiz Structure:  
Use a list of dictionaries to store:**

* **question, options, correct\_answer**

2. Implement & Test

quiz = [

{

"question": "What data type is '42' in Python?",

"options": {"A": "int", "B": "str", "C": "float", "D": "bool"},

"correct": "B"

},

{

"question": "Which keyword defines a function in Python?",

"options": {"A": "func", "B": "define", "C": "def", "D": "function"},

"correct": "C"

},

{

"question": "What symbol is used for comments in Python?",

"options": {"A": "//", "B": "#", "C": "/\* \*/", "D": "--"},

"correct": "B"

}

]

score = 0

for q in quiz:

print("\n" + q["question"])

for key, value in q["options"].items():

print(f"{key}: {value}")

while True:

answer = input("Your answer (A/B/C/D): ").strip().upper()

if answer in q["options"]:

break

print("Invalid choice. Please choose A, B, C, or D.")

if answer == q["correct"]:

print("✅ Correct!")

score += 1

else:

print(f"❌ Incorrect. The correct answer was {q['correct']}.")

percentage = (score / len(quiz)) \* 100

print(f"\nFinal Score: {score}/{len(quiz)} ({percentage:.1f}%)")

**Section 3: Reflection**

**Q1: What was your most valuable AI conversation and why?**  
The most helpful AI chat was about the temperature converter. It showed me how to check if the user enters a number and how to give a nice error message if they don't. I learned how to make my code more user-friendly.

**Q2: Describe one time you disagreed with or changed the AI’s suggestion.**  
When I made the quiz, the AI told me to use a list of tuples, but it was hard to read. So I used a list of dictionaries instead. It made my code easier to understand and change later.