**Sergio Bravo**

**Lesson 13 Prove Assignment: Student Chosen Program**

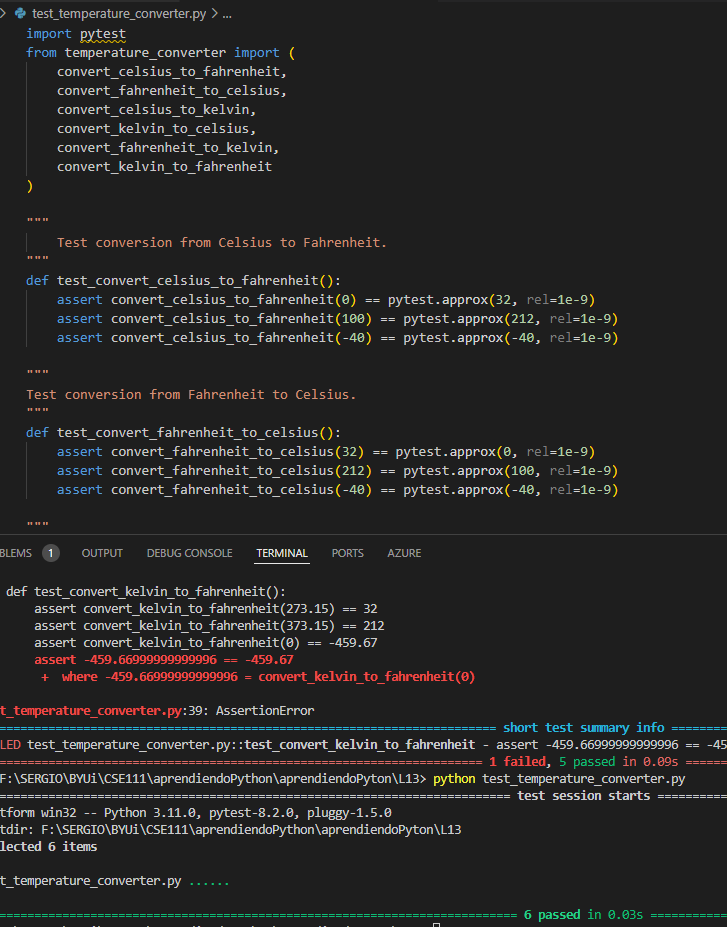
**Purpose**: Prove that you can write a significant Python project that solves a real-world problem and is well organized with functions. My program is called Temperature Converter, and it involves entering a number and then telling it what temperature it represents, for example, if it is celcius, it will assign a letter C to the second request that the program will ask for, then enter the temperature of the conversion, for example, if es farengeit will give you an F

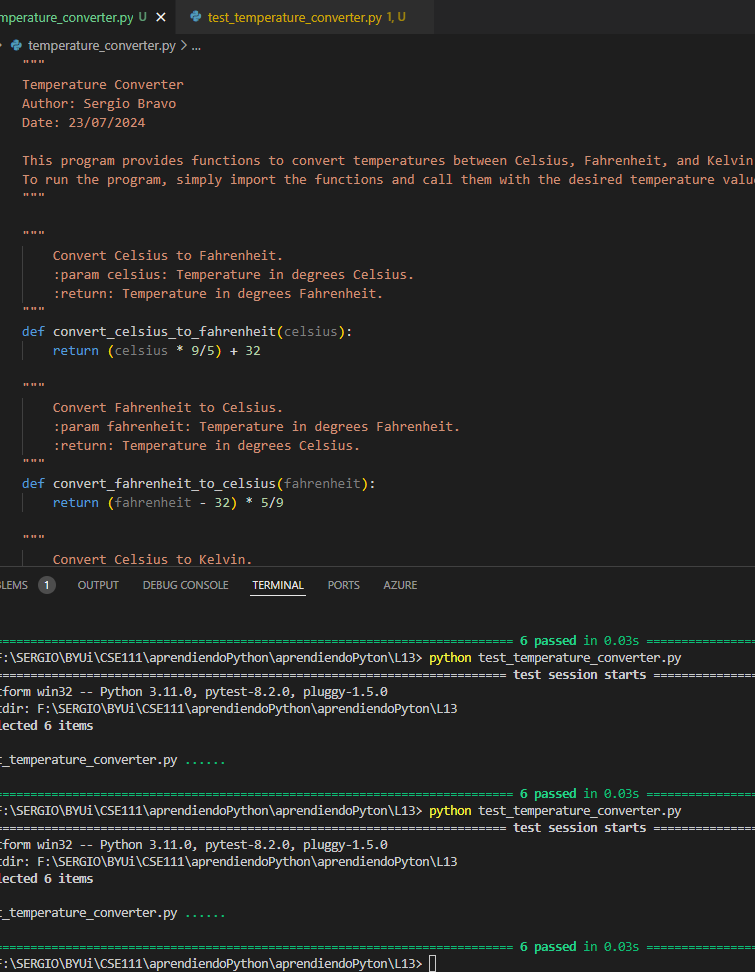
**Time Spent**:

* **Day 1 (2 hours)**:
  + Reviewed the project requirements and planned the structure.
  + Set up the development environment and created initial function stubs.
  + Started implementing basic temperature conversion functions.
* **Day 2 (1.5 hours)**:
  + Continued implementing the remaining temperature conversion functions.
  + Added initial test cases for the conversion functions.
  + Conducted research on best practices for writing test functions and handling floating-point precision issues.
* **Day 3 (1 hour)**:
  + Refined the test cases using pytest.approx for precision handling.
  + Ran initial tests and debugged any issues that arose.
  + Conducted more research on pytest to ensure comprehensive test coverage.
* **Day 4 (1.5 hours)**:
  + Completed the remaining test functions.
  + Ensured that all functions were covered by corresponding test functions.
  + Finalized the program code, added comments, and improved readability.

**Description of Work**:

1. **Overall Design**:
   * The program includes functions for converting temperatures between Celsius, Fahrenheit, and Kelvin.
   * Each conversion function performs a specific task, ensuring modularity and reusability.
   * Test functions are provided to validate the correctness of each conversion function.
2. **Function Interactions**:
   * The conversion functions are independent of each other and do not share state.
   * Each function takes a single input (a temperature value) and returns the converted temperature.
   * The test functions call the conversion functions with various inputs to verify their correctness.
3. **Test Functions**:
   * **Conversion Functions**:
     + convert\_celsius\_to\_fahrenheit
     + convert\_fahrenheit\_to\_celsius
     + convert\_celsius\_to\_kelvin
     + convert\_kelvin\_to\_celsius
     + convert\_fahrenheit\_to\_kelvin
     + convert\_kelvin\_to\_fahrenheit
   * **Test Functions**:
     + test\_convert\_celsius\_to\_fahrenheit
     + test\_convert\_fahrenheit\_to\_celsius
     + test\_convert\_celsius\_to\_kelvin
     + test\_convert\_kelvin\_to\_celsius
     + test\_convert\_fahrenheit\_to\_kelvin
     + test\_convert\_kelvin\_to\_fahrenheit
4. **Compiled Research**:
   * **Documentation**:
     + [Official Python documentation on math module](https://docs.python.org/3/library/math.html)
     + [pytest documentation for writing test functions](https://docs.pytest.org/en/stable/contents.html)
   * **Videos**:
     + "[Python Programming Tutorial: Functions](https://www.youtube.com/watch?v=u-OmVr_fT4s&ab_channel=ProgrammingwithMosh)" on YouTube
     + "[Unit Testing with Pytest](https://www.youtube.com/watch?v=YbpKMIUjvK8&ab_channel=pixegami)" on YouTube
   * **Coding Experiments**:
     + Experimented with different ways to implement temperature conversions
     + Tested various edge cases and unusual values for temperature conversion functions
5. **Conclusion**:
   * **Completed Work**:
     + Implemented all required temperature conversion functions.
     + Created and ran comprehensive test cases for each function.
   * **Lessons Learned**:
     + Importance of handling floating-point precision in numerical computations.
     + Value of modular and reusable code design.
   * **Ah-ha Moments**:
     + Using pytest.approx to handle precision issues was a significant improvement.
   * **Screenshots**:





**Submission**:

1. **Python Program File**: temperature\_converter.py
2. **Python Test File**: test\_temperature\_converter.py