

**Name: Hamza Ahmad**

**Intern ID: TN/IN02/PY/008**

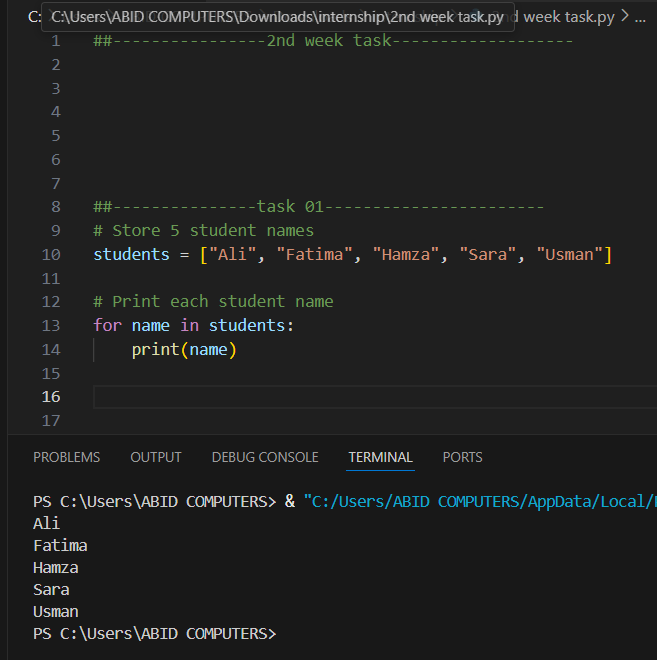
**Email ID: hamzaahmad3632@gmail.com**

**Task no: Week 02.**

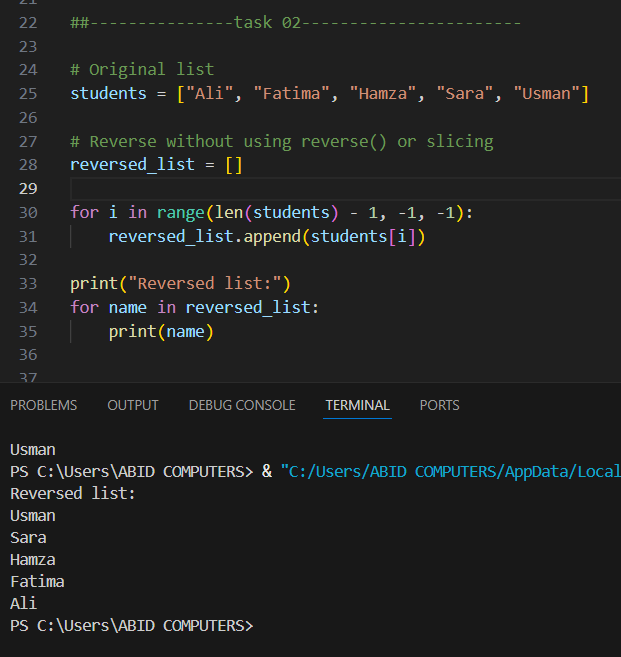
**Internship Domain: Python**

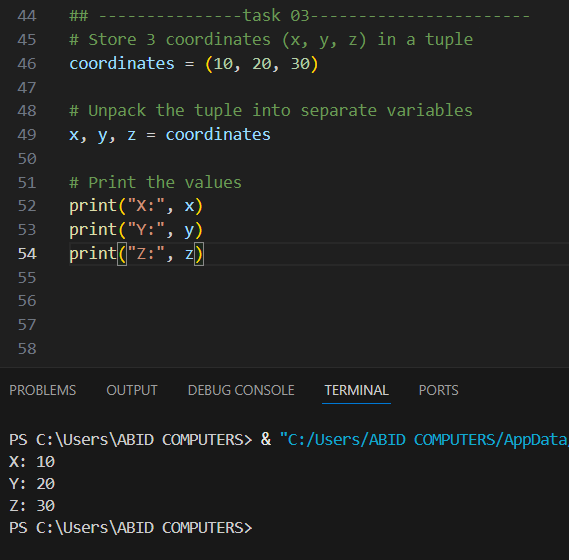
**Instructor Name: Mr. Hassan Ali**

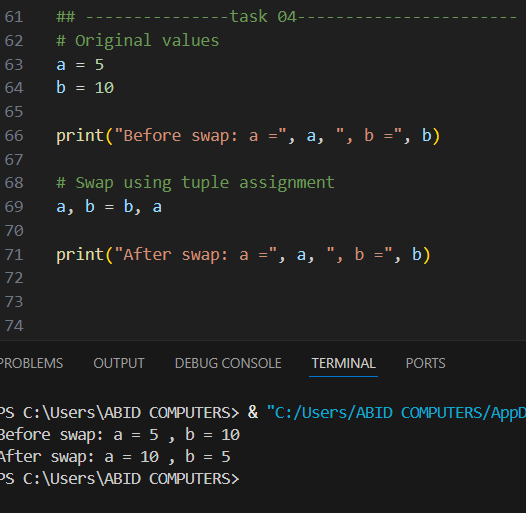
**1. Store 5 student names and print them**  
In this task, I created a simple list to store five student names and used a loop to print each name.  
🔹 What I learned: How to use lists and loops in Python.  
🔹 Challenge: It was easy, but I had to remember the proper syntax of a for loop.

****

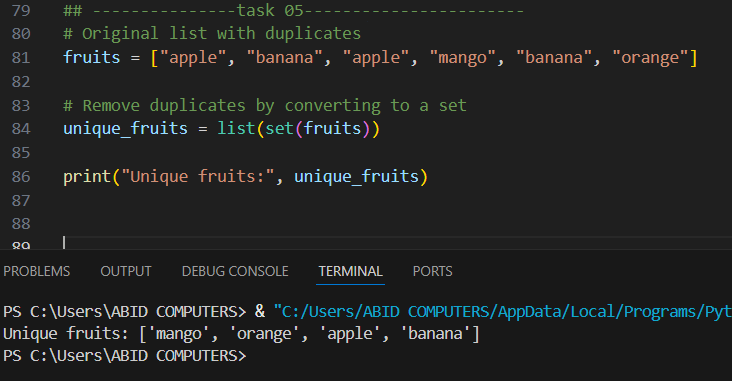
**2. Reverse list without using .reverse()**  
Here, I learned how to reverse a list using slicing like list[::-1] instead of using built-in functions.  
🔹 What I learned: Python slicing and how it can manipulate data.  
🔹 Challenge: Understanding what the -1 does in slicing was a bit tricky at first.

****

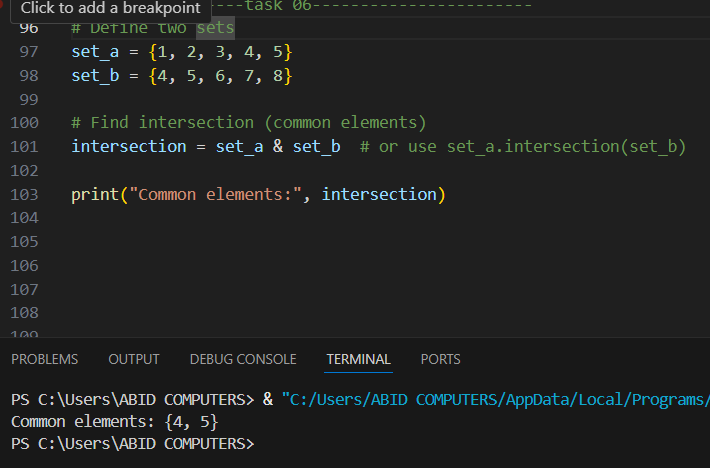
**3. Store 3 coordinates and unpack**  
I stored a tuple of 3 coordinates like (x, y, z) and unpacked it into separate variables.  
🔹 What I learned: Tuple unpacking and multiple variable assignment.  
🔹 Challenge: I initially confused lists with tuples and had to learn the difference.****

**4. Swap variables using tuple assignment**  
Instead of using a temporary variable, I used a, b = b, a to swap values.  
🔹 What I learned: A clean and Pythonic way to swap variables.  
🔹 Challenge: No major challenge—this was a very straightforward concept.****

**5. Remove duplicate items from a list**  
I removed duplicates by converting the list to a set and back to a list using list(set(my\_list)).  
🔹 What I learned: How sets automatically remove duplicate values.

🔹 Challenge: The set doesn't preserve the original order, which I had to consider.****

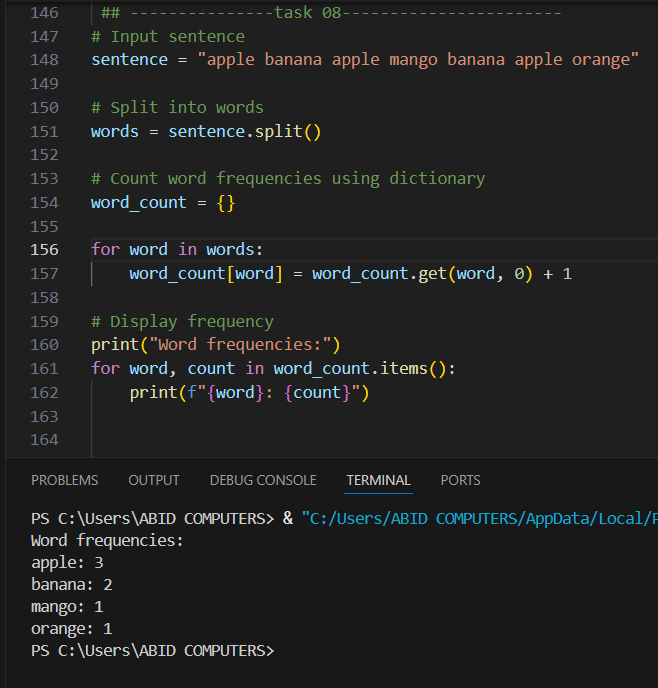
**6. Find intersection of two sets**  
I created two sets and found common elements using set1 & set2.  
🔹 What I learned: Basic set operations like intersection.

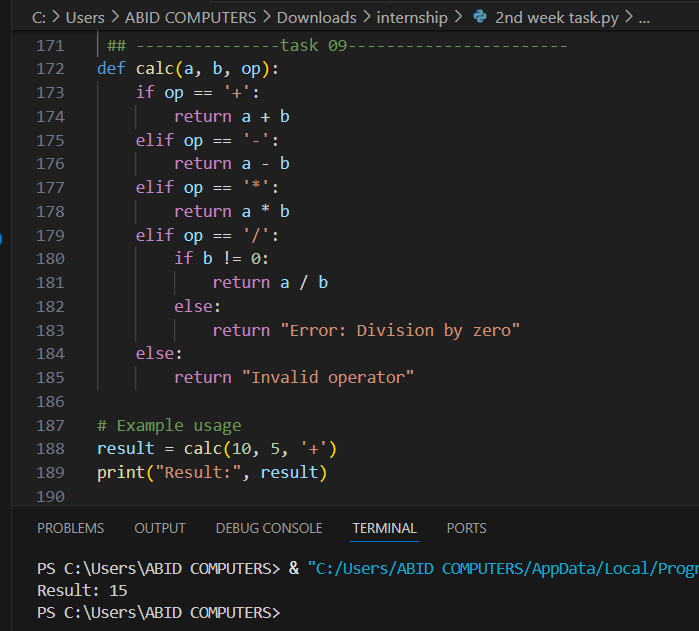
🔹 Challenge: Remembering the correct operator for intersection.****

**7. Student record CRUD using dictionary**  
This was a small console-based app to Create, Read, Update, and Delete student records using

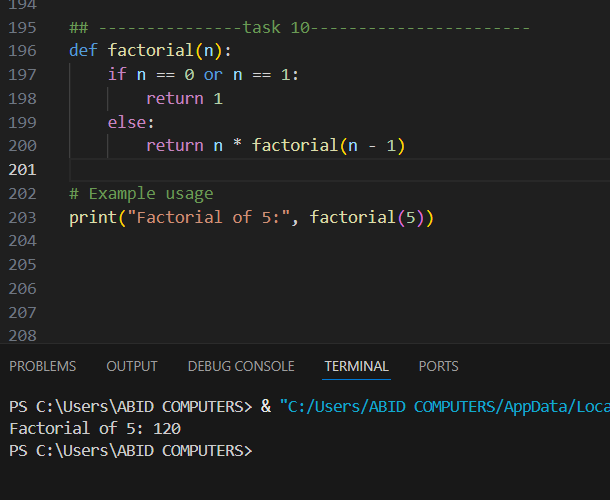
a dictionary.  
🔹 What I learned: How to manage dynamic data using dictionaries and basic CRUD

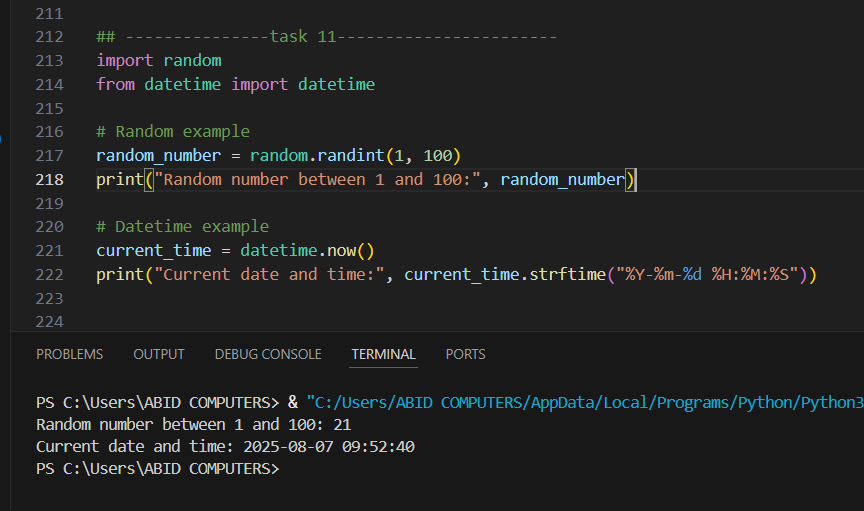
operations.  
🔹 Challenge: Writing clean code to handle user choices and prevent key errors.****

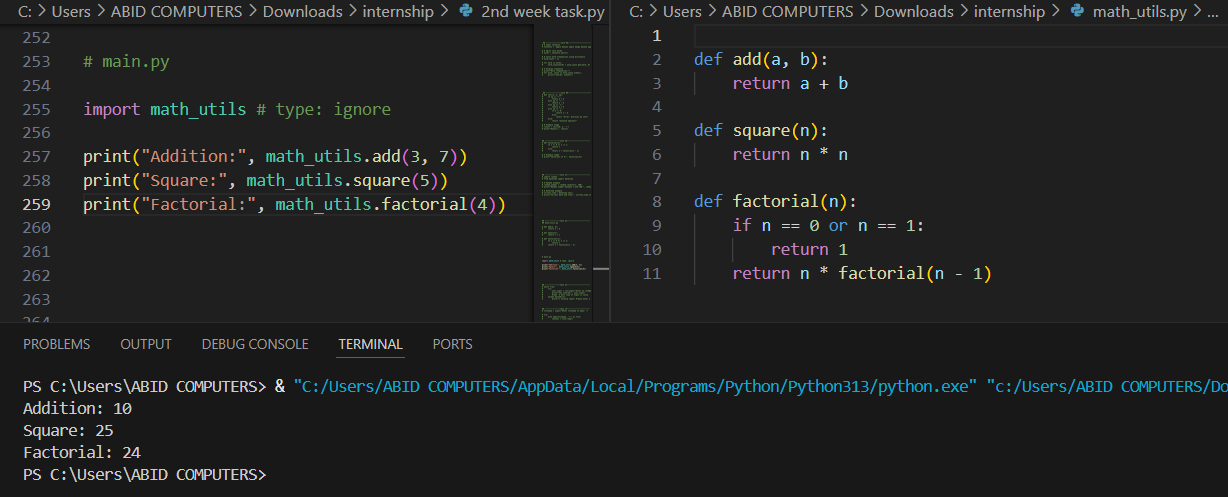
**8. Count word frequency in a sentence**  
I took a sentence, split it into words, and used a dictionary to count how often each word appeared.  
🔹 What I learned: How to loop through words and use dictionaries to count items.  
🔹 Challenge: Handling case-sensitivity and removing punctuation for accurate results.****

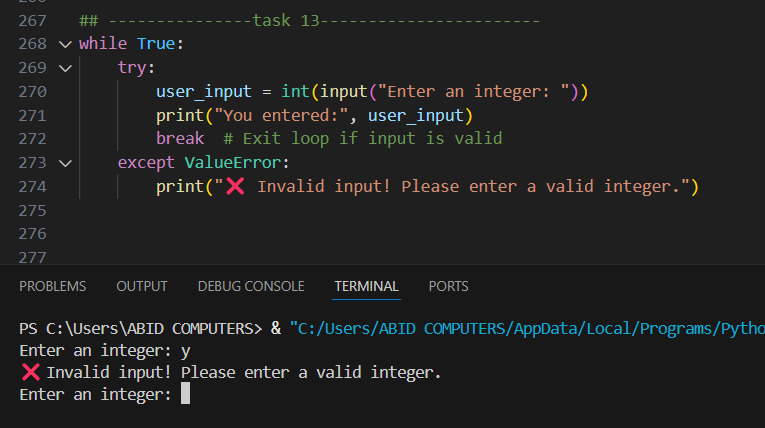
**9. Write a calc(a, b, op) function**  
I made a calculator function that performs operations based on user input like "+", "-", "\*", or "/".  
🔹 What I learned: Function creation, conditional logic, and operator handling.  
🔹 Challenge: Handling divide-by-zero errors and invalid operator input.****

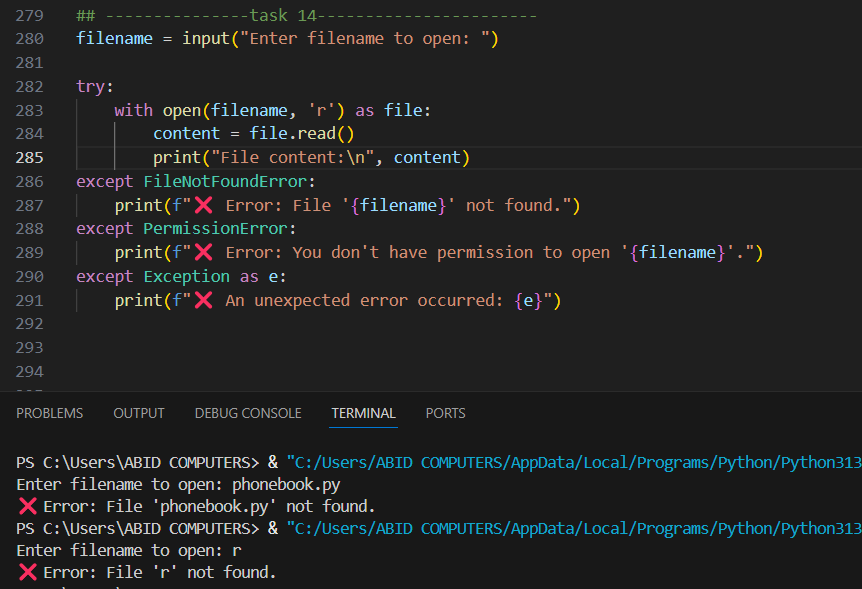
**10. Write recursive factorial(n)**  
I wrote a recursive function to calculate factorial, with a base case of n == 0.  
🔹 What I learned: How recursion works and how to define base and recursive cases.  
🔹 Challenge: Understanding the recursive flow was hard at first but became clear after practice.

****

**11. Use random and datetime modules**  
In this task, I imported two Python modules—random to generate random numbers and datetime to get the current date and time.  
🔹 What I learned: How to use built-in Python libraries.  
🔹 Challenge: Remembering to import modules and using correct function names.****

**12. Create math\_utils module & import**  
In this task, I created a separate Python file called math\_utils.py and added basic math functions like add(), subtract(), etc. Then I imported it into another script using import math\_utils and used its functions.  
🔹 What I learned: How to build and reuse my own Python module for better code organization.  
🔹 Challenge: Faced minor confusion about file paths and making sure both files were in the same folder so that the import worked properly.****.

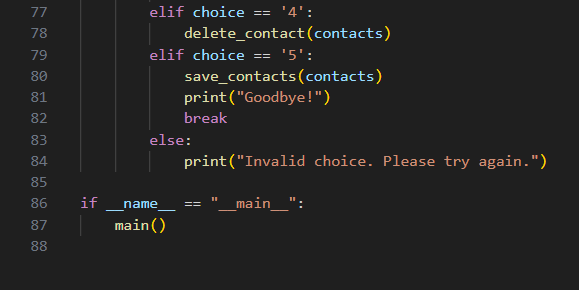
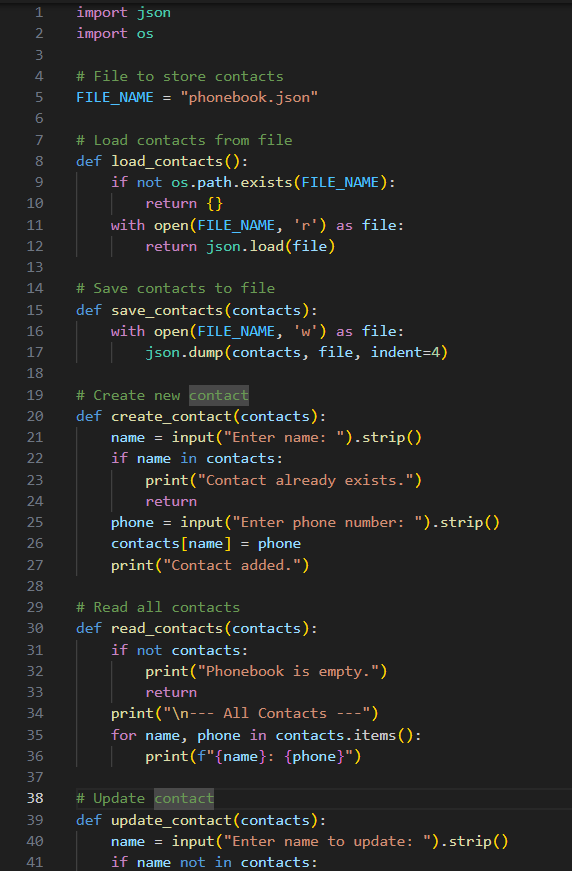
**13. Safe integer input using loop**  
I created a loop that keeps asking for a number until the user enters a valid integer using try-except.  
🔹 What I learned: How to handle user input errors and prevent program crashes.  
🔹 Challenge: Understanding exception handling and choosing the right exception (ValueError). ****

**14. File open with error message**  
This task taught me to open a file safely using try-except and show a friendly error message if the file doesn't exist.  
🔹 What I learned: Basics of file handling and managing file-related exceptions.  
🔹 Challenge: Knowing when to use FileNotFoundError specifically **15. Phonebook App: CRUD contact dict ↔ JSON file**  
In this task, I built a small phonebook application using a Python dictionary to store contact information (like name, phone number, etc.). I implemented basic CRUD operations:

* **Create**: Add a new contact
* **Read**: View existing contacts
* **Update**: Modify an existing contact
* **Delete**: Remove a contact

To make the data **persistent**, I used the json module to **save the dictionary to a JSON file** and **load it back** when the program runs again.

🔹 *What I learned:* How to build a practical mini-application, use file I/O, and work with the json module to store structured data permanently.  
🔹 *Challenge:* The biggest challenge was converting between Python dictionaries and JSON format properly, and making sure to read/write the file without errors.

****

