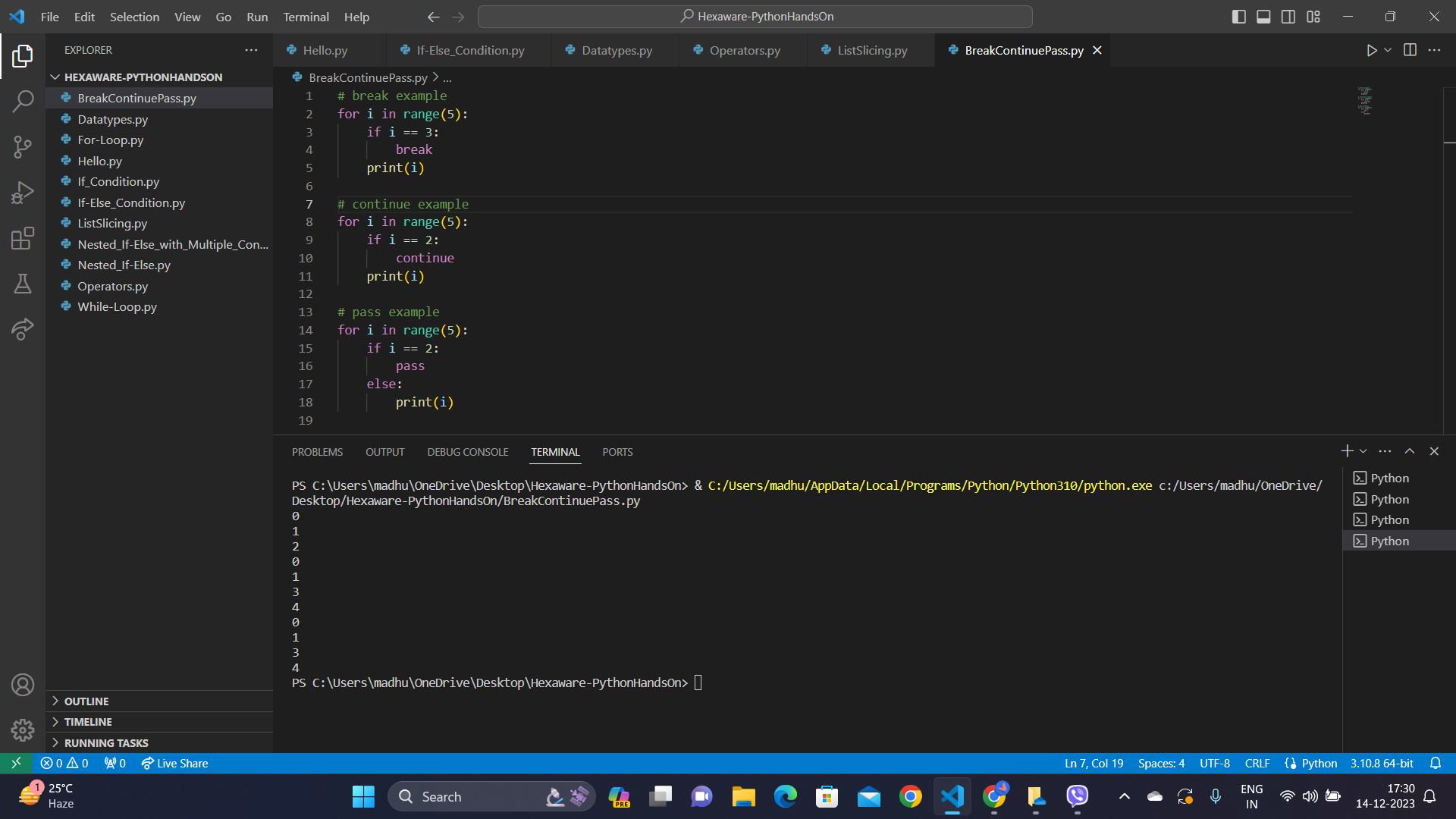
**Day- 10  
Madhu Kalyani Gadi (14-12-2023)**

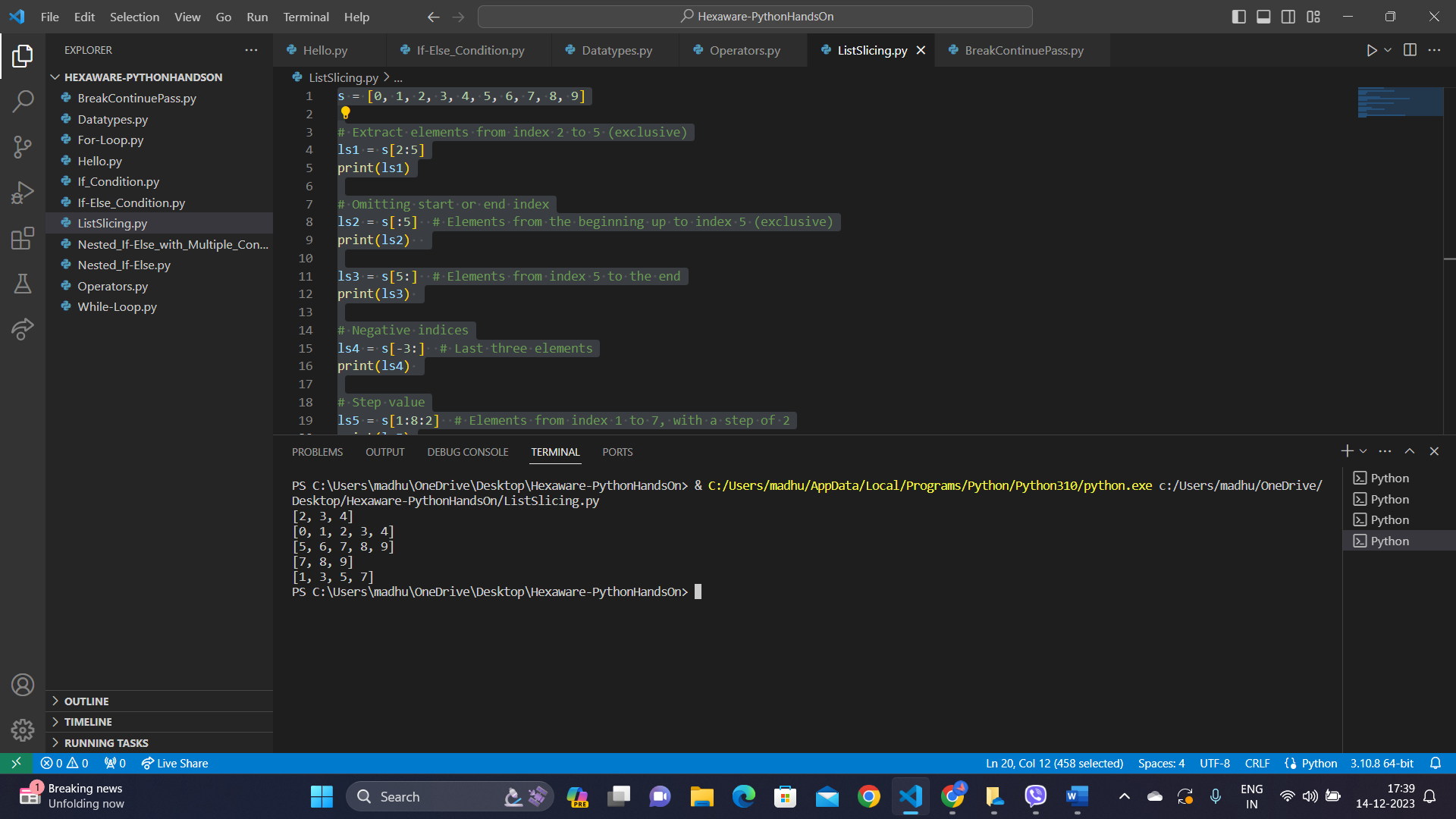
**Loop Control Statements**  
These are used to alter the flow of execution within loops. There are three main loop control statements.  
  
**Break:**  
It is used to exit the loop. Below, when i reaches 3, the loop is terminated, and the program moves on to the next statement after the loop.  
  
**Continue:**  
It is used to skip the rest of the code inside a loop for the current iteration and proceed to the next iteration.  
Below, when i reaches 3, the print(i) statement is skipped, and the loop proceeds to the next iteration.  
  
**Pass:**

It is a null operation, meaning nothing happens when it is encountered. Below, when i is equal to 3, the pass statement is encountered, and nothing happens. The program continues with the else block.  
  


**List Slicing:**

List slicing is a technique in Python that allows you to create a new list by extracting a portion of an existing list.   
  
Syntax: list[start:stop:step]

1. ls1 = s[2:5] extracts elements from index 2 to 4 (exclusive) in the list s, resulting in the sublist [2, 3, 4].
2. ls2 = s[:5] creates a sublist from the beginning of the list s up to index 4 (exclusive), producing [0, 1, 2, 3, 4].
3. ls3 = s[5:] forms a sublist of elements from index 5 to the end of the list s, yielding [5, 6, 7, 8, 9].
4. ls4 = s[-3:] utilizes negative indices to extract the last three elements of the list s, resulting in [7, 8, 9].
5. ls5 = s[1:8:2] extracts elements from index 1 to 7 with a step of 2, creating the sublist [1, 3, 5, 7].

  
  
**Operators**  
  
Operators in Python are symbols or special keywords that perform operations on operands.

1. Arithmetic Operators: Add, subtract, multiply, divide, find the remainder, raise to a power, and get the whole number from division.
2. Comparison Operators: To check if something is equal to, not equal to, greater than, less than, or equal to something else.
3. Logical Operators: To check if two conditions are both true (and), if at least one condition is true (or), or if a condition is not true (not).
4. Identity Operators: To check if two things are the same—see if two variables refer to the exact same object or if they are different.
5. Membership Operators: To check if something is in a group—see if a value is present in a list, tuple, or other iterable (in) or if it's not present (not in).

