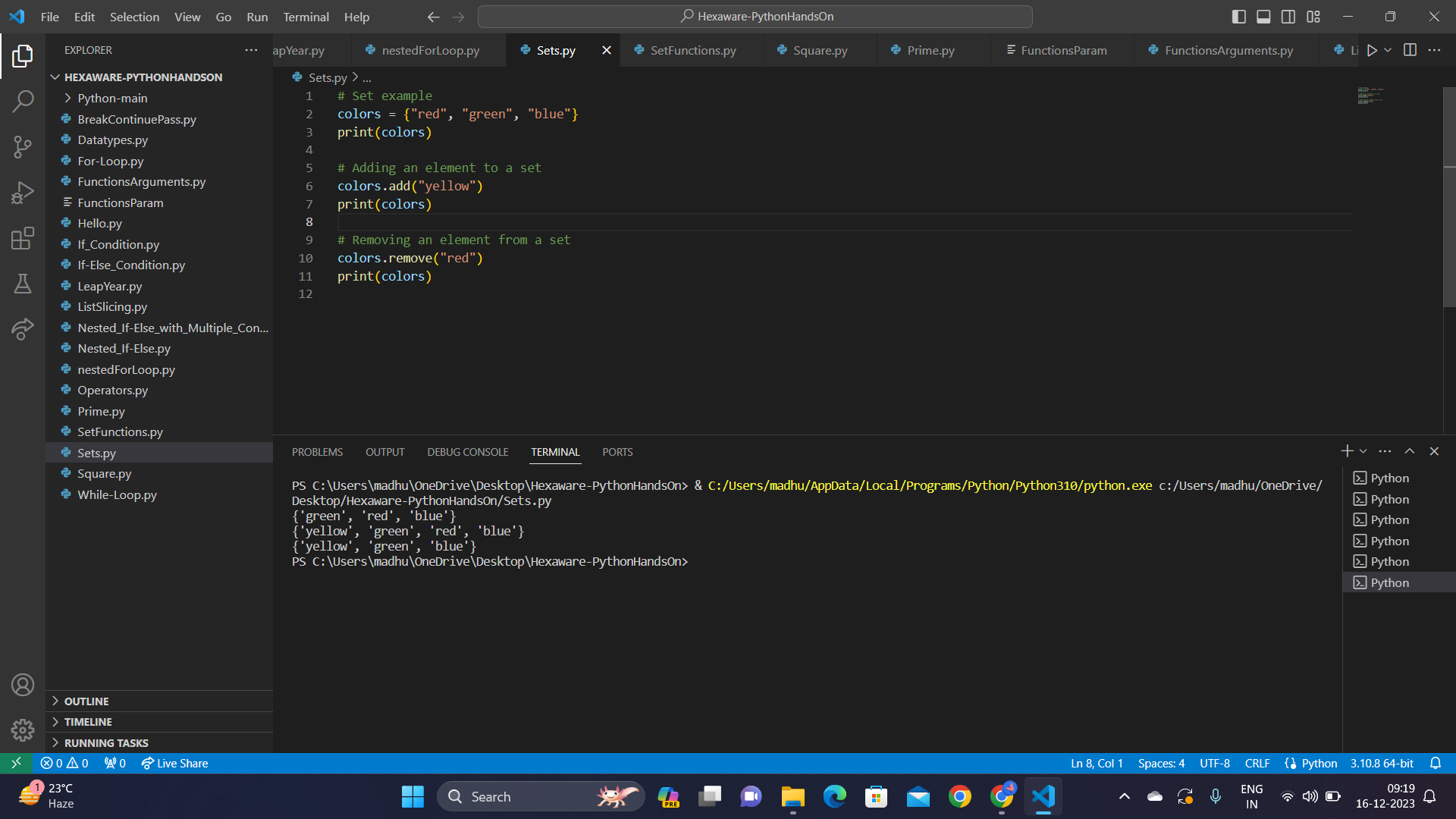
**Day-11  
Madhu Kalyani Gadi (15-12-2023)**

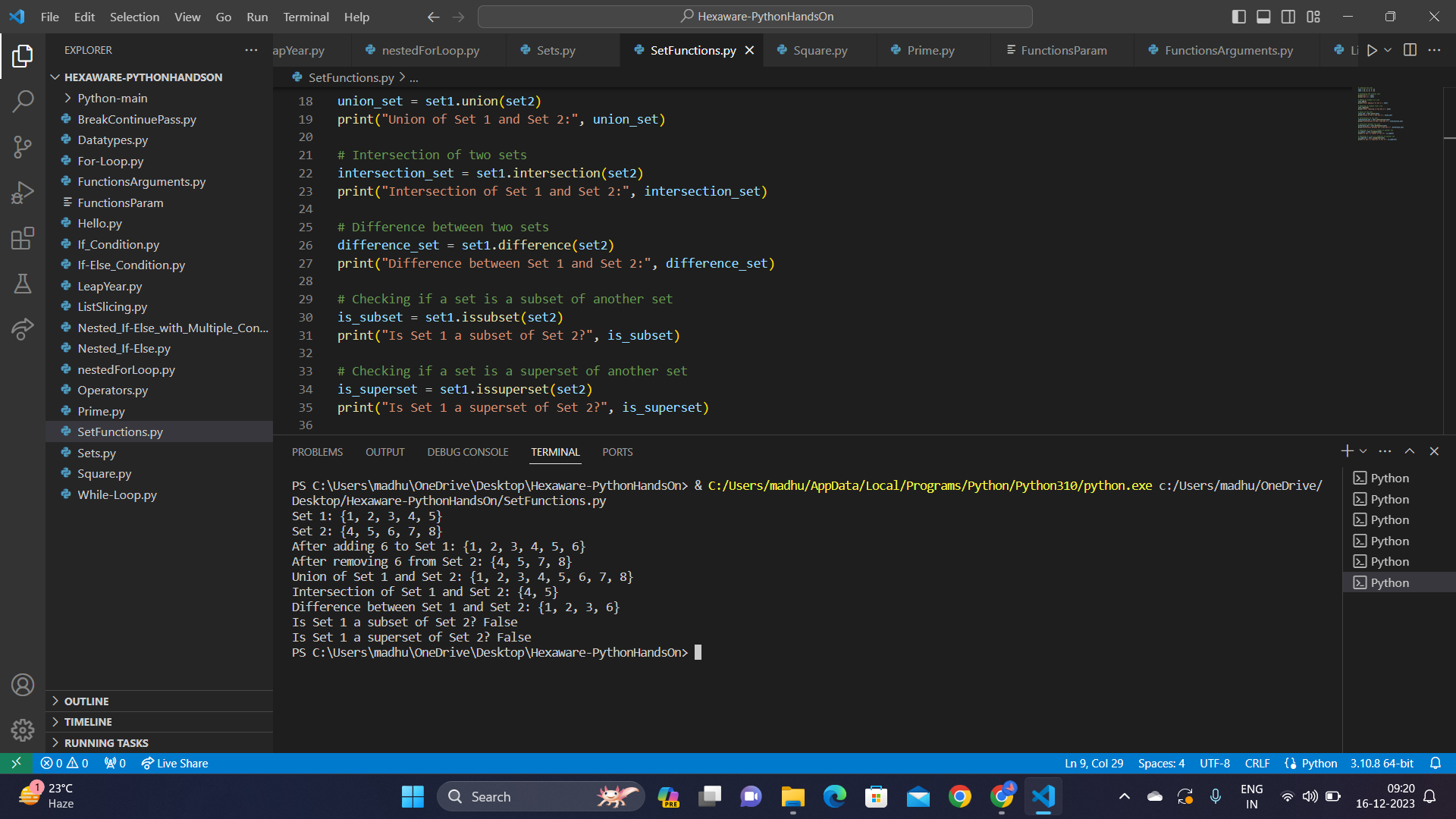
**Set Data Type:**

A set is an unordered collection of unique elements. It is defined using curly braces {} or the set() constructor.  


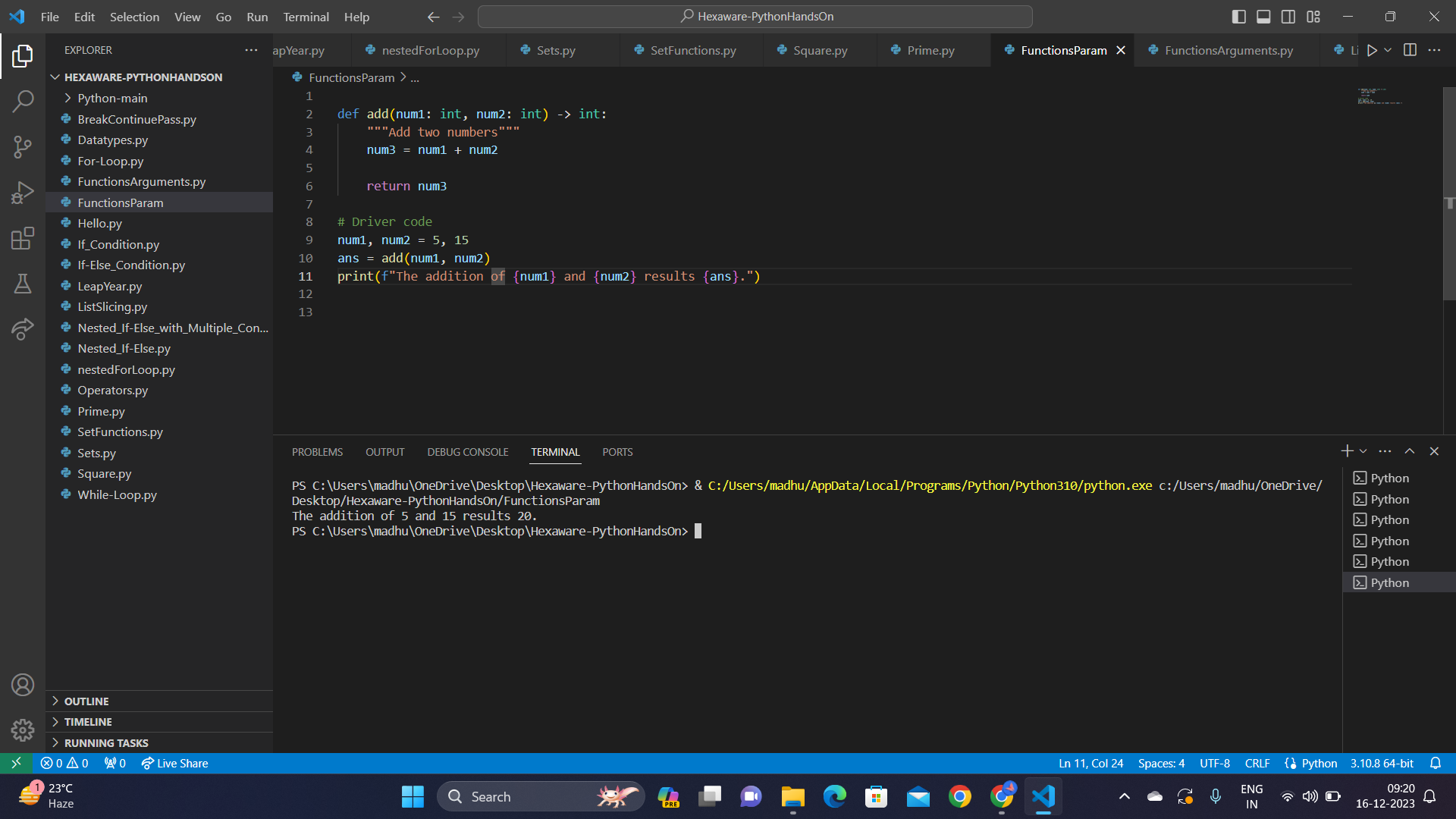
**Set Functions:**

Set functions provide operations on unordered collections of unique elements.

1. Adding an element to a set: The add(6) operation adds the element 6 to set1.
2. Removing an element from a set: The remove(6) operation removes the element 6 from set2.
3. Union of two sets: The union\_set = set1.union(set2) creates a new set containing all unique elements from both set1 and set2.
4. Intersection of two sets: The intersection\_set = set1.intersection(set2) creates a new set containing common elements between set1 and set2.
5. Difference between two sets: The difference\_set = set1.difference(set2) creates a new set with elements present in set1 but not in set2.
6. Checking if a set is a subset of another set: The is\_subset = set1.issubset(set2) evaluates whether set1 is a subset of set2 and returns a boolean.
7. Checking if a set is a superset of another set: The is\_superset = set1.issuperset(set2) evaluates whether set1 is a superset of set2 and returns a boolean.

  
  
**Python Functions with Parameters**  
Functions with parameters allow to define and execute reusable blocks of code; parameters act as input values.  
Syntax:   
def my\_function(param1, param2):

#body

my\_function(value1, value2)  
  
  
  
**Python functions with Arguments:**  
Functions with arguments allow passing values during function calls, influencing the function's nature.

Types of Function Arguments:

1. Default Arguments: Some function parameters have default values, so we don't always need to provide values for them. For example, in myFun(x, y=50), y defaults to 50 unless you specify a different value.
2. Keyword Arguments: You can explicitly name the values you're passing to a function using keywords. In student(firstname='Hexa', lastname='Practice'), it's clear that 'Hexa' corresponds to firstname and 'Practice' to lastname.
3. Positional Arguments: When calling a function, the values you provide are matched to parameters in the order they're defined. So, nameAge("Madhu", 27) works, but changing the order like nameAge(27, "Madhu") can lead to unexpected results.
4. Arbitrary Arguments (\*args): With \*args, a function can accept any number of arguments. In myFun1('Hello', 'Welcome', 'to', 'HexaforHexa'), all the values are captured and printed, allowing flexibility in the number of arguments passed.

