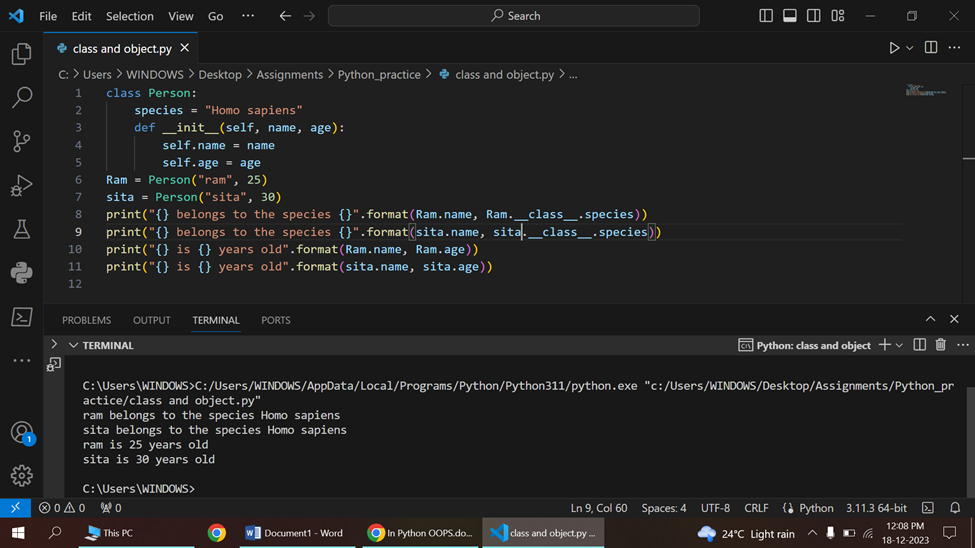
OOPS

S.R.TAANUSRI

18.12.2023

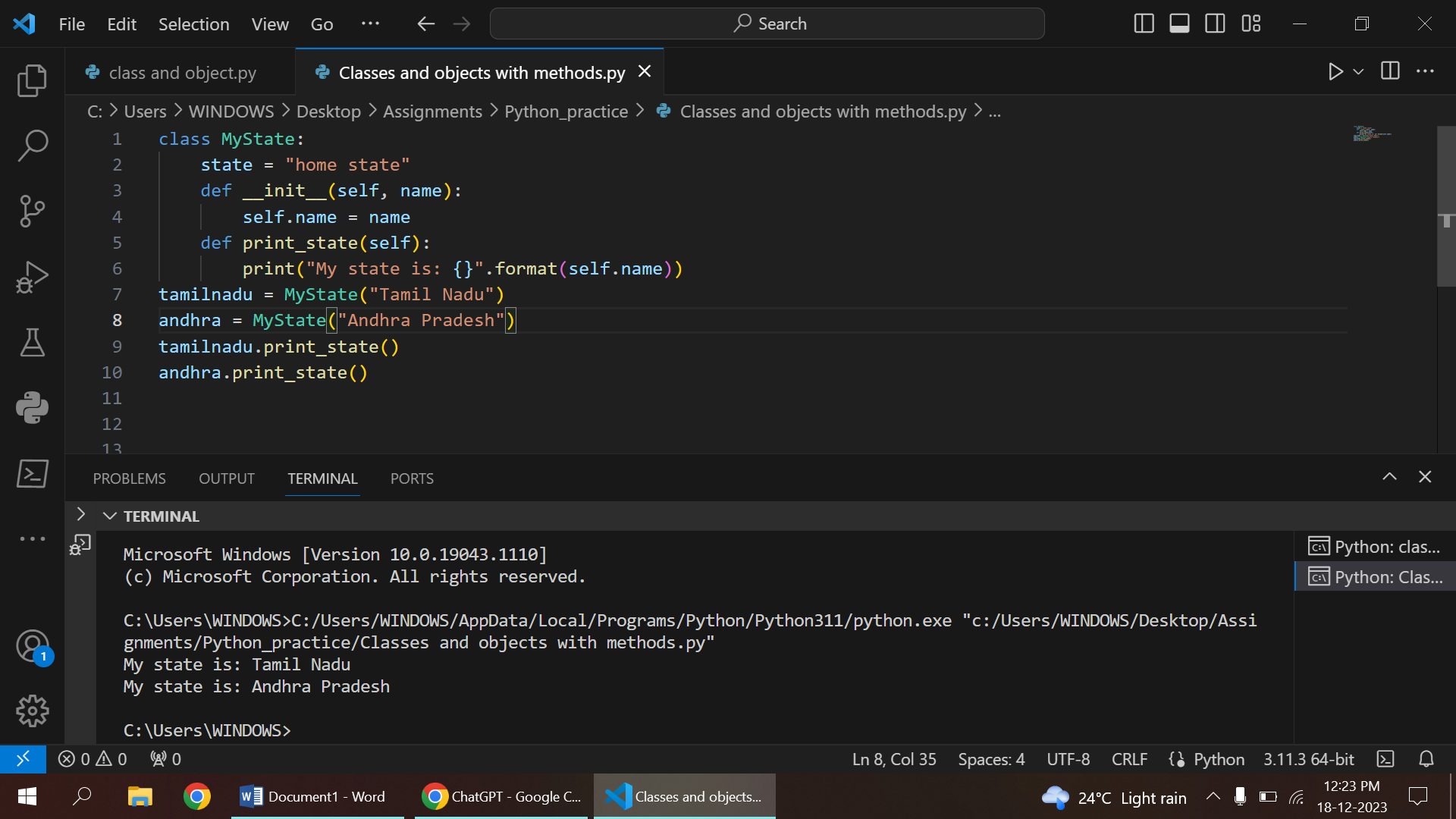
**1.Creating class and object:**

* a class is a user-defined blueprint or template for creating objects.
* It defines a data structure that encapsulates attributes (variables) and methods (functions) that operate on those attributes.
* An object is an instance of a class created using the class as a blueprint.
* It represents a real-world entity and has its own unique identity, attributes, and behavior.
* Multiple objects can be created from the same class.



**2.Class and object with methods**

* A method is defined within a class using the def keyword, similar to a function. The first parameter of a method is always self, which refers to the instance of the class.



class MyClass:

    def my\_method(self, param1, param2):

        # Method implementation goes here

* The self parameter is a convention in Python that represents the instance of the class. It is the first parameter of every method and is used to refer to the instance's attributes and other methods.

class Person:

    def \_\_init\_\_(self, name, age):

        self.name = name

    def display\_info(self):

        print(f"Name: {self.name}")

3.INHERITANCE

* Inheritance is a fundamental concept in object-oriented programming (OOP).
* Allows a class (subclass or derived class) to inherit attributes and methods from another class (base class or parent class). I
* n Python, inheritance supports the creation of a hierarchy of classes, promoting code reuse and modularity.
* There are three types of inheritance in Python:

1.Single Inheritance:

a class inherits from only one base class.

2.Multiple Inheritance:

a class inherits from more than one base class.

3.Multilevel Inheritance:

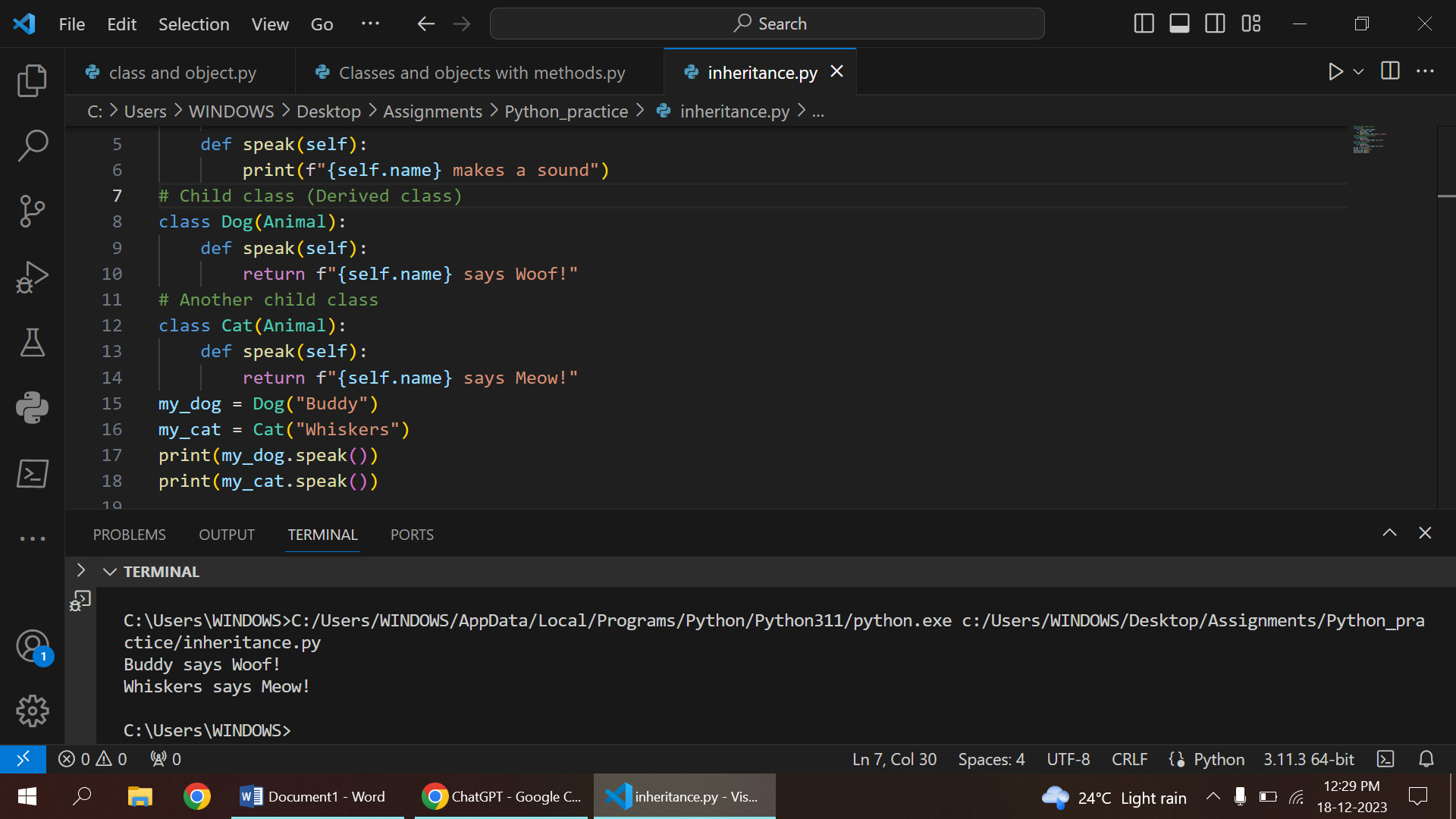
a class inherits from another class, and then a new class is derived from the first derived class.

4.Hierarchial inheritance:

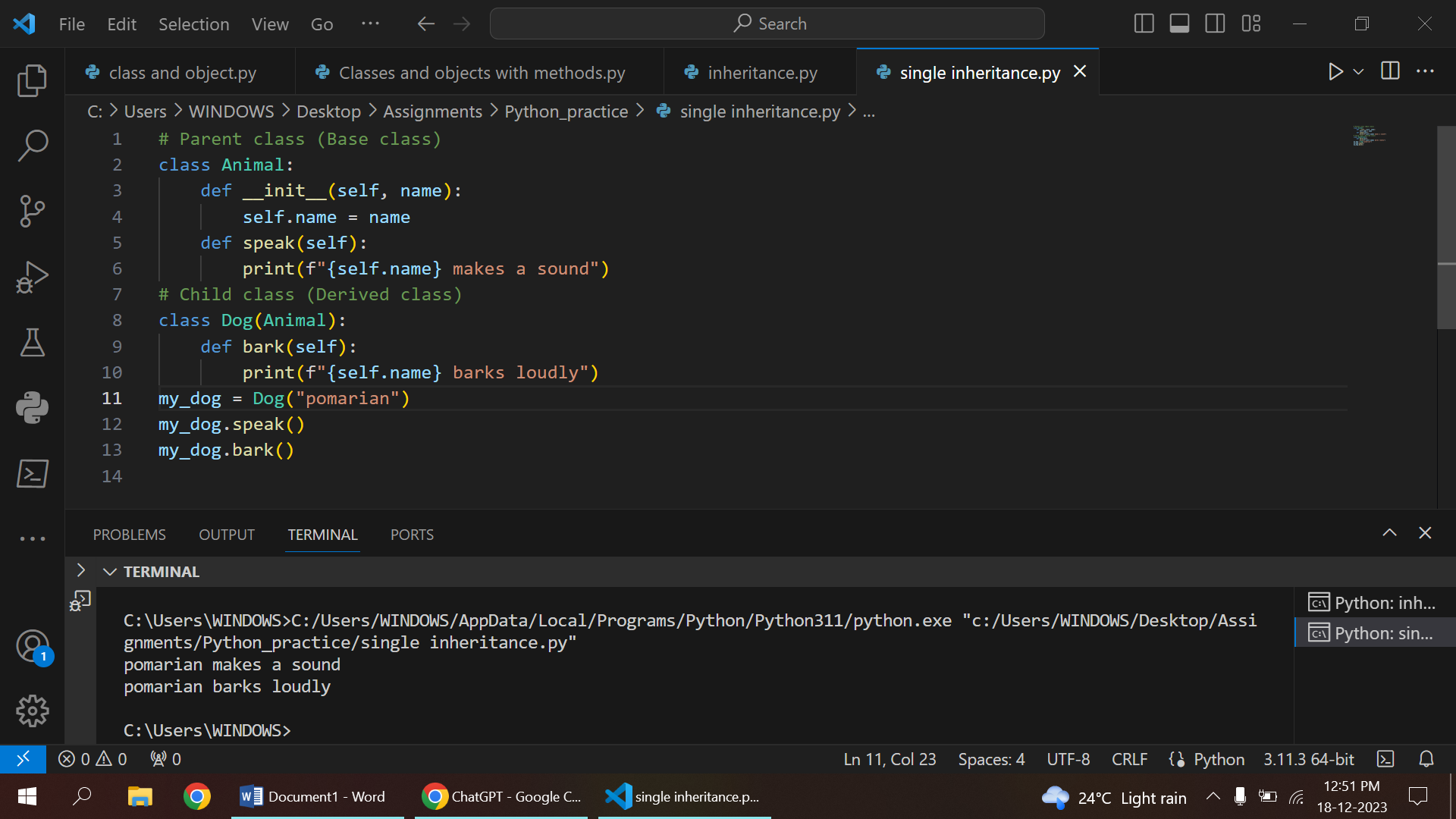
a type of inheritance in which multiple derived classes inherit from a single base or parent class.

5.Hybrid Inheritance:

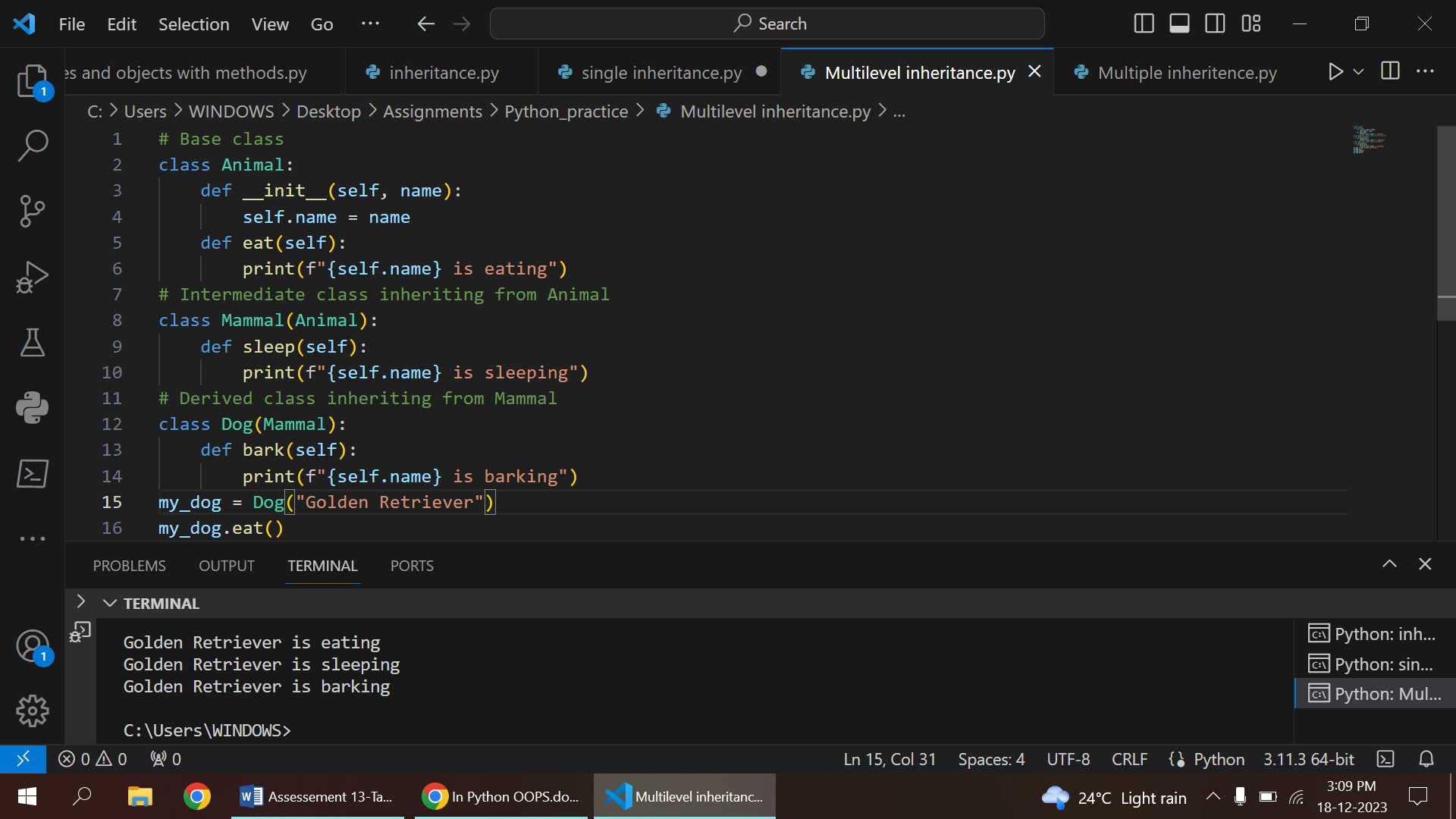
a combination of different types of inheritance, such as single, multiple, and hierarchical inheritance, within a single program or system.



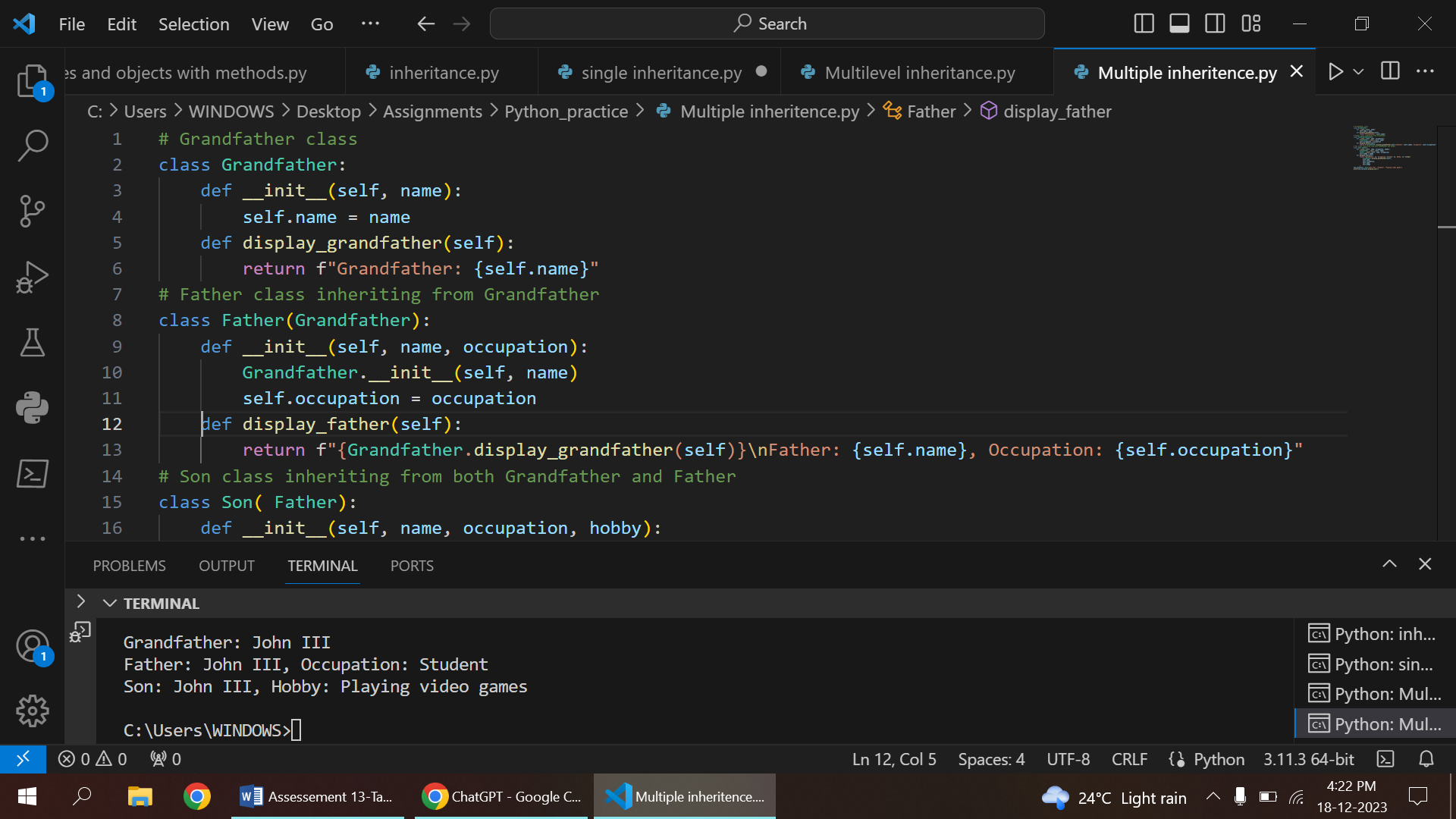
SINGLE INHERITANCE



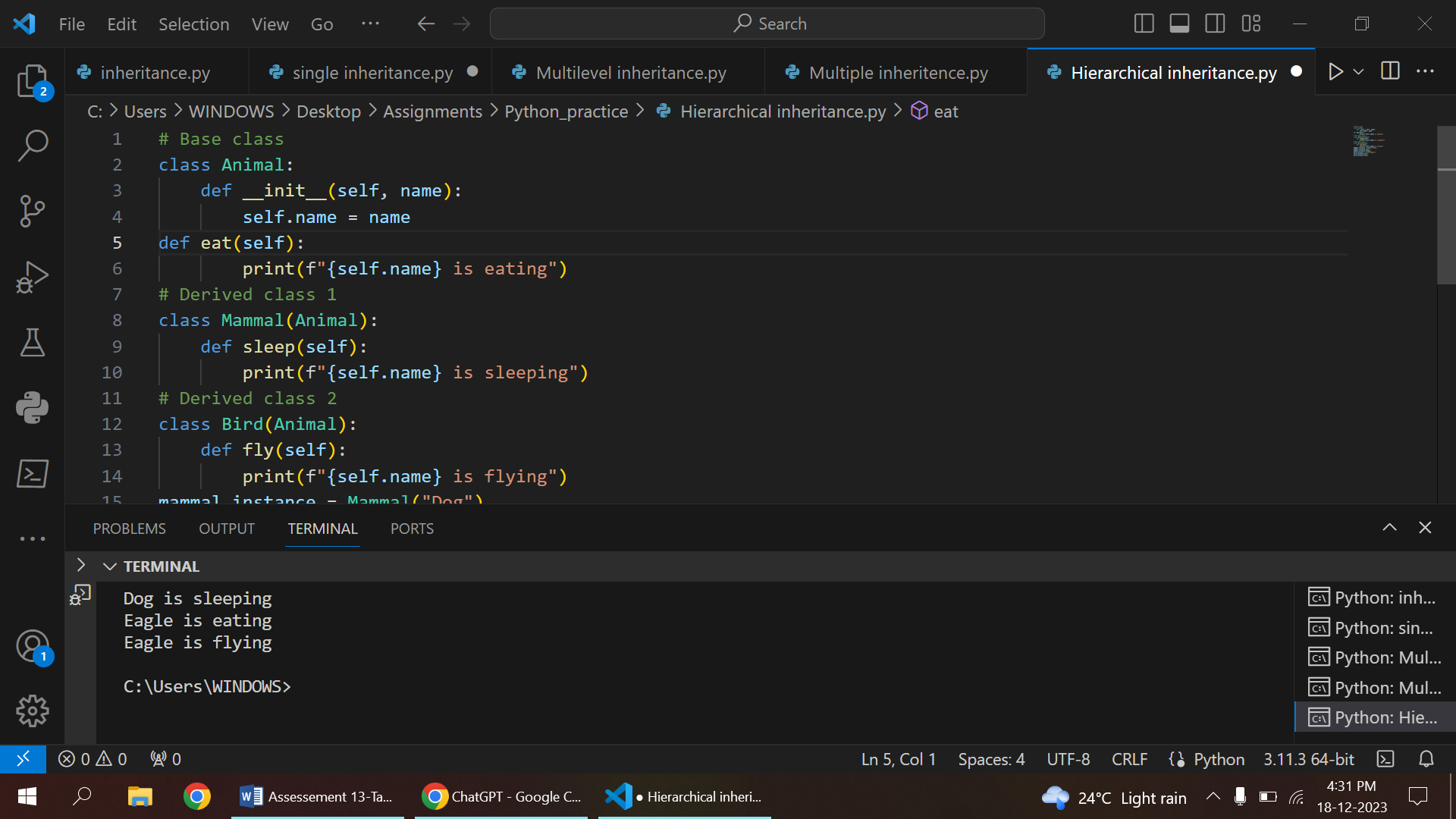
MULTILEVEL INHERITANCE



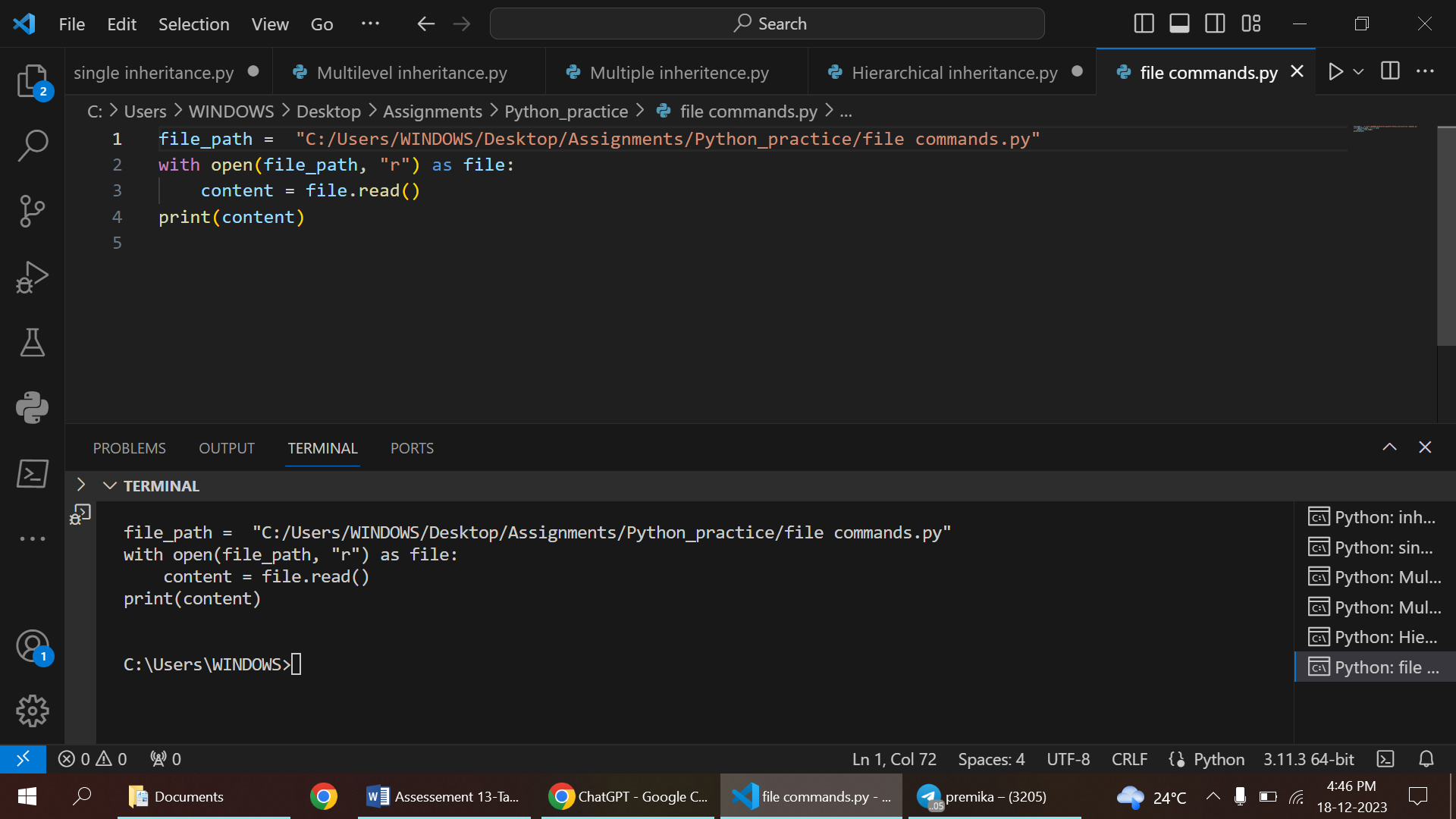
MULTIPLE INHERITANCE



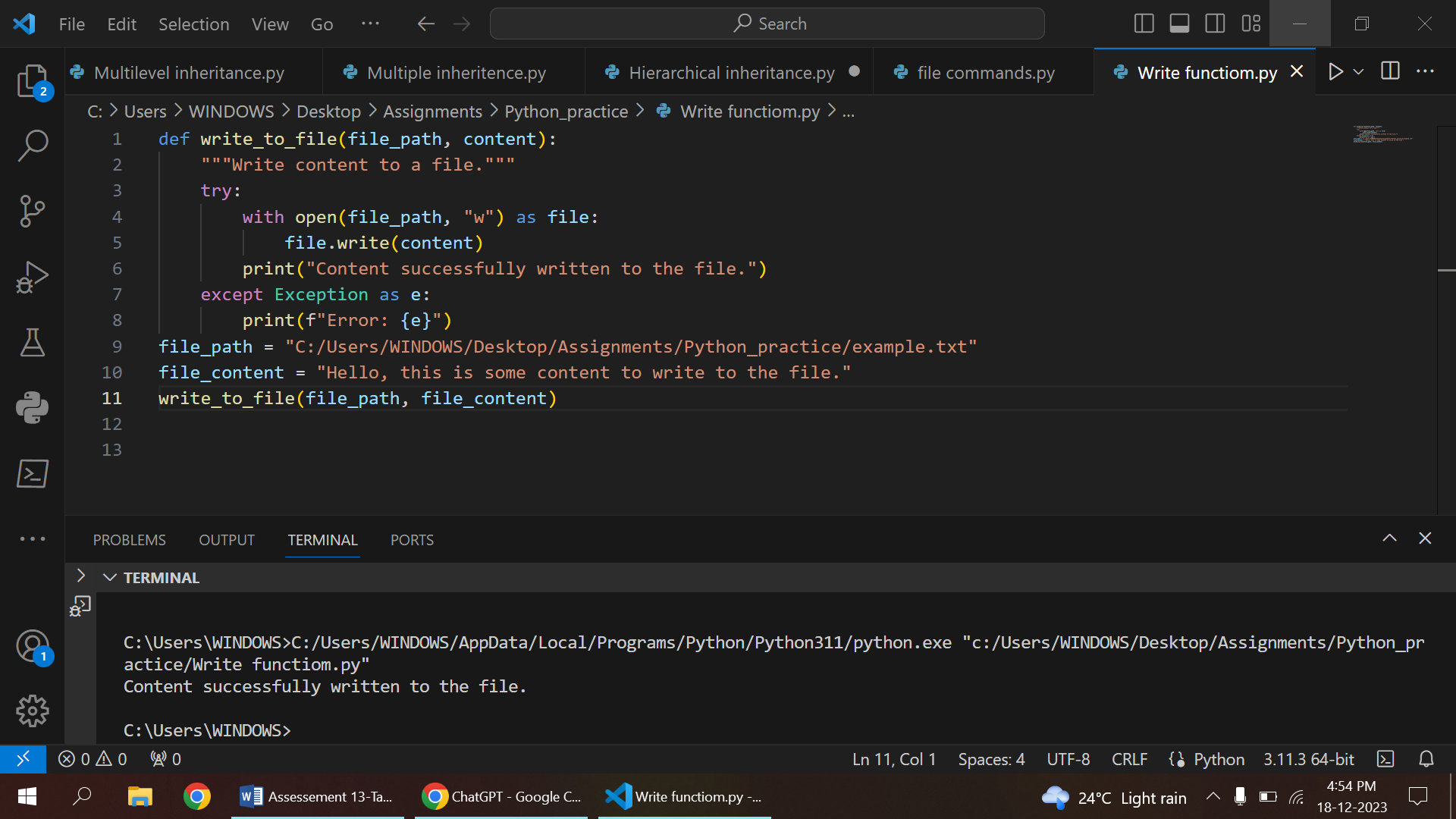
HIERARCHIAL INHERITANCE



READ FUNCTION()

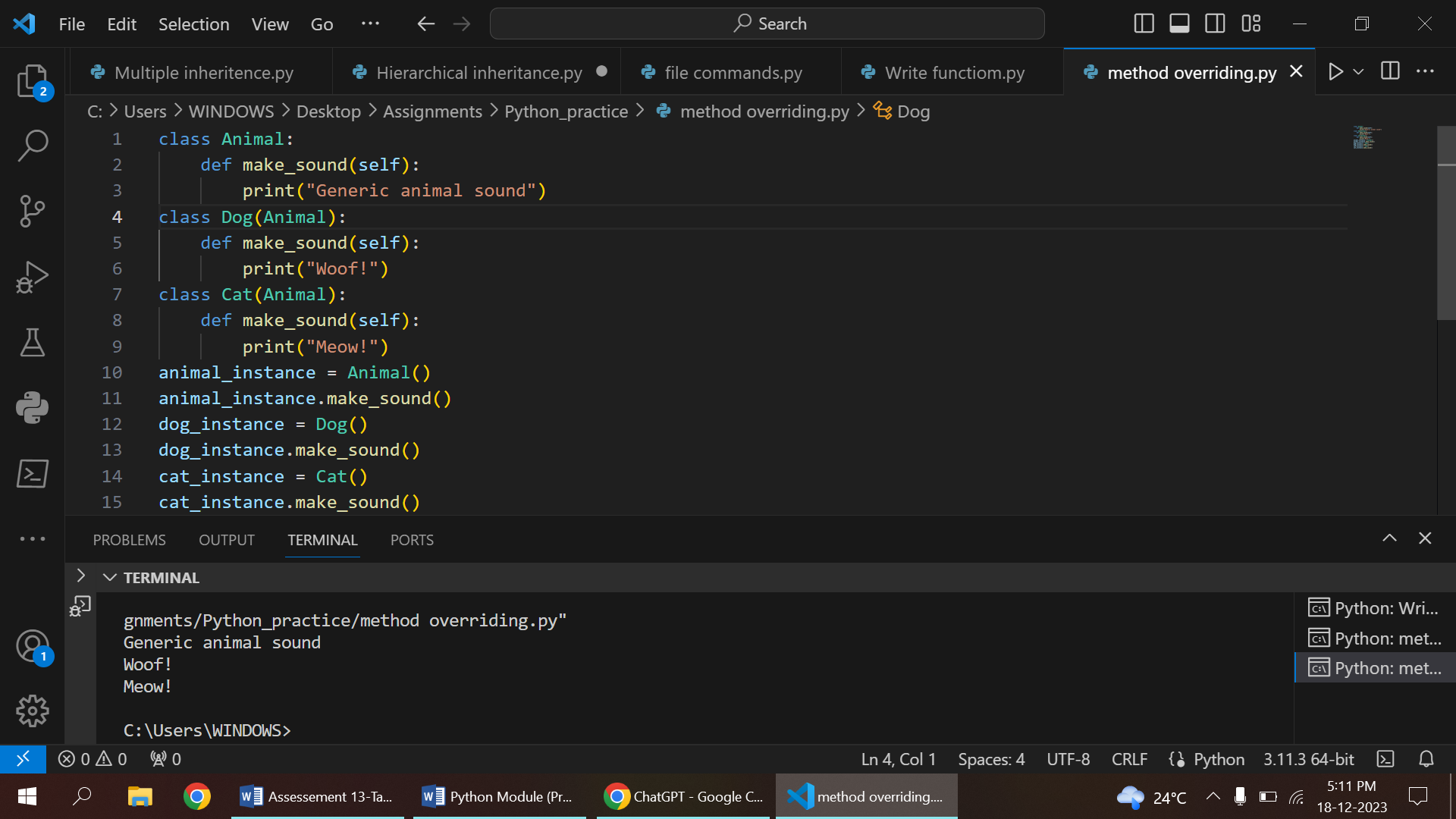


WRITE() FUNCTION



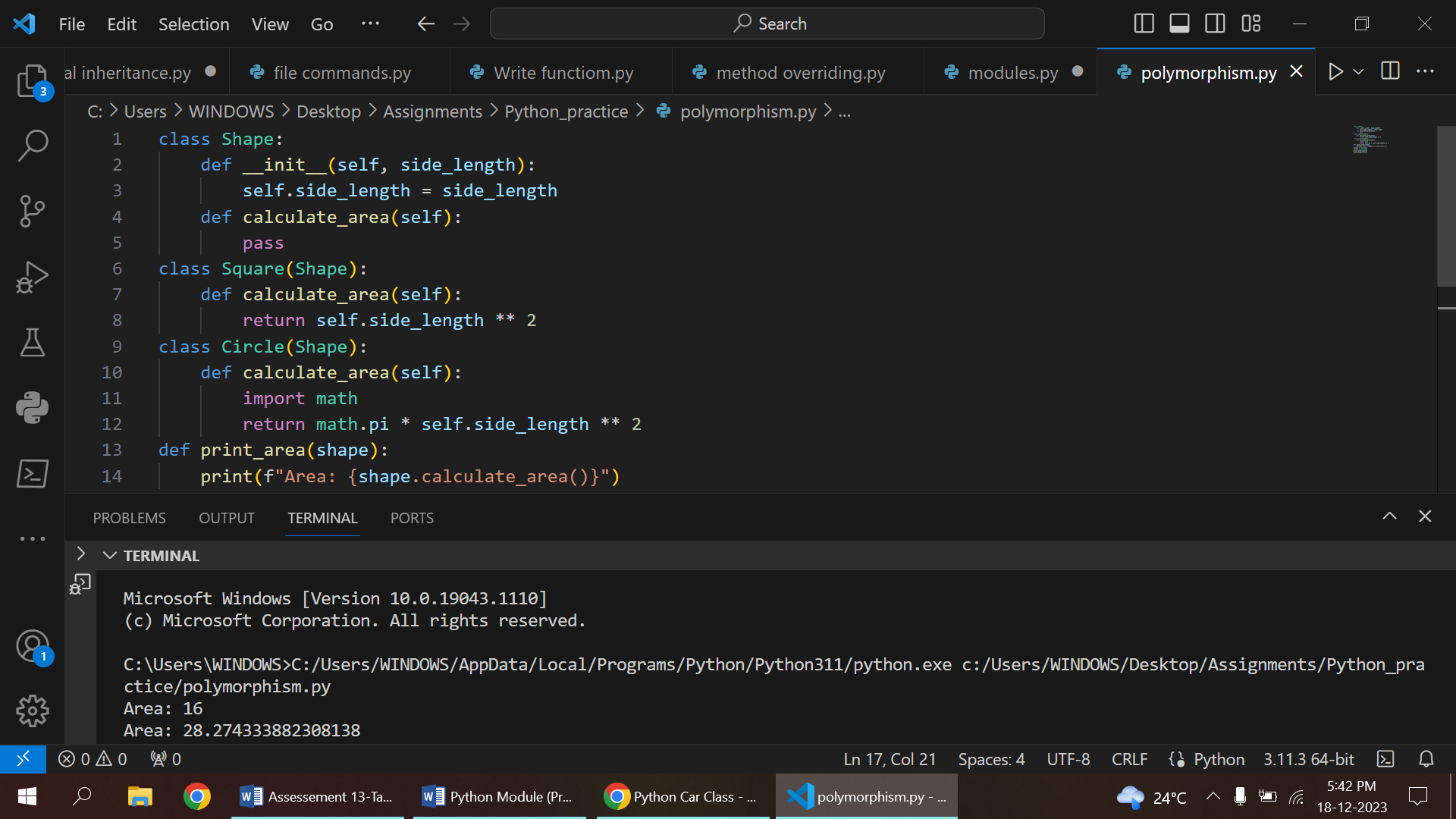
METHOD OVERRIDING

* Method overriding in Python occurs when a derived class provides a specific implementation for a method that is already defined in its base class.
* This allows the derived class to tailor the behavior of the inherited method to better suit its own needs.



POLYMORPHISM

* The ability of a single function or method to work with different types of objects, or the ability of different classes to provide a common interface for their instances.
* There are two main types of polymorphism in Python:
  1. compile-time (or static) polymorphism
  2. run-time (or dynamic) polymorphism



ENCAPSULATION

* Encapsulation in Python is the concept of bundling the data (attributes) and the methods (functions) that operate on the data into a single unit, known as a class.
* It restricts access to some of the object's components, preventing the accidental modification of data from outside the class.
* encapsulation is achieved through the use of private and protected attributes and methods.

