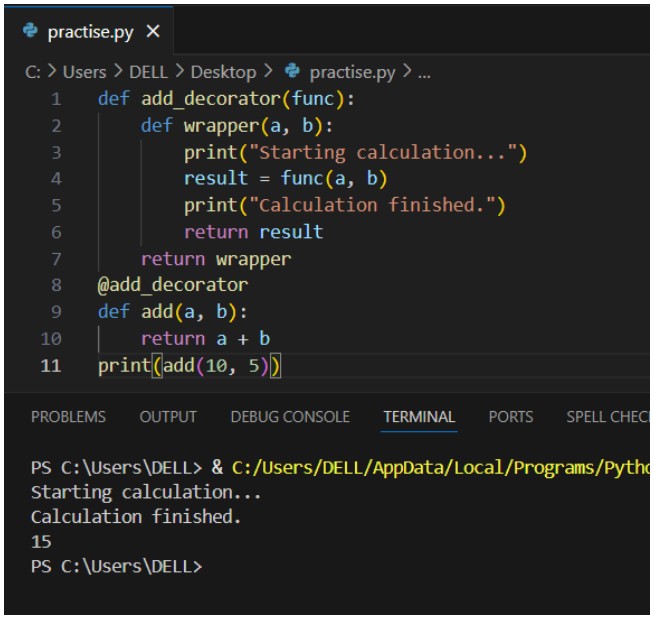
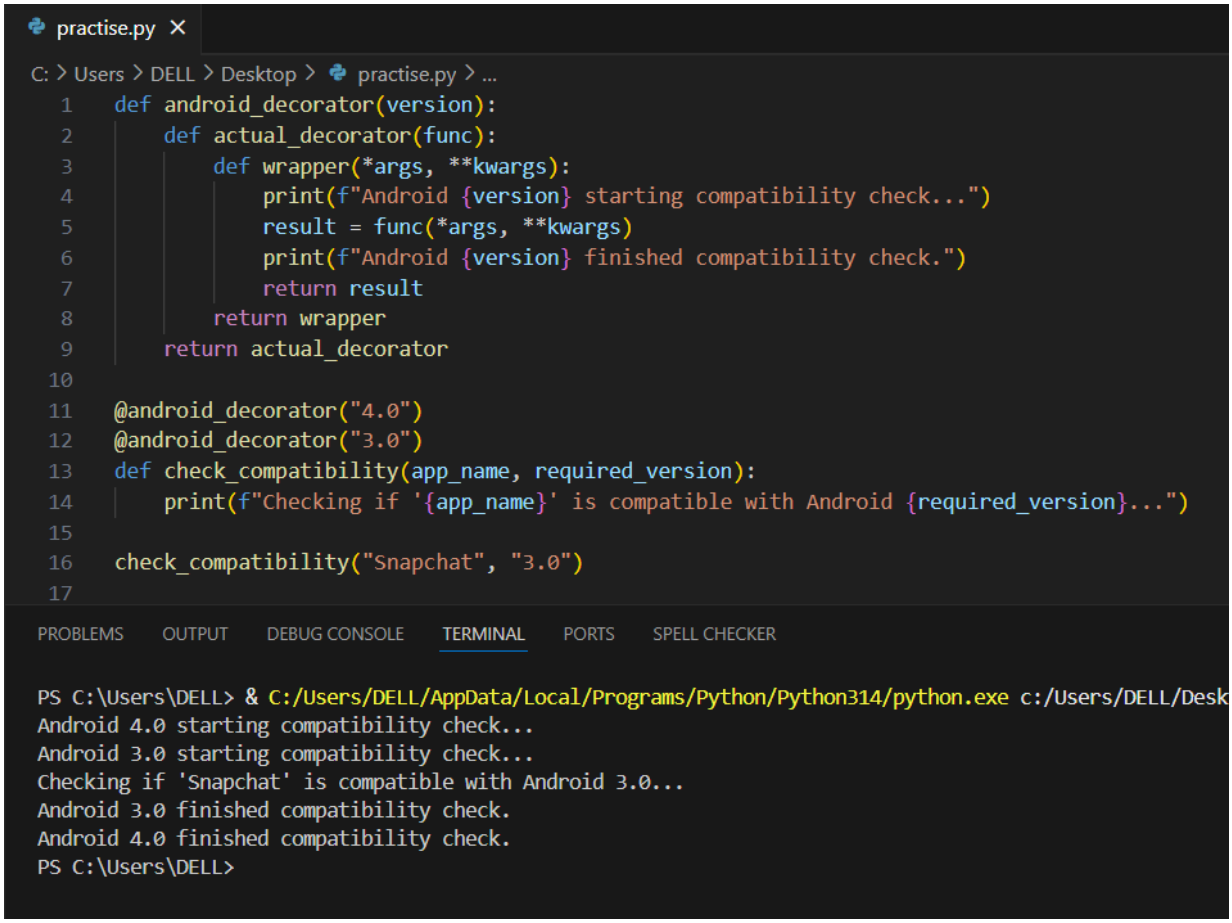
Assessment on Python Decorators:

In Python, a **decorator** is a special function that allows us to modify or extend the behavior of another function — without changing its actual code.
For example :



=> When add(10, 5) is called, it doesn't directly jump into add. Instead, it first runs the decorator's function which is @add_decorator.



=> When you call → check_compatibility("Snapchat", "3.0")

=>Python reads it from the bottom up. That means,The function check_compatibility is first passed into android_decorator("3.0").
=>Then it takes that result and **wraps** it again with android_decorator("4.0").
=>Android 3.0's wrapper starts and prints, Android 3.0 starting compatibility check...
=>Inside Android 3.0's wrapper, the real function check_compatibility finally runs and prints, Checking if 'Snapchat' is compatible with Android 3.0...
=>Once the inner function finishes, Android 3.0's wrapper continues. It prints,Android 3.0 finished compatibility check. Then, it gives back its result to Android 4.0.
=>Now, control comes back to Android 4.0's wrapper. After receiving the inner result, it prints: Android 4.0 finished compatibility check.