Alejandra Reyes

3/25/2021

Homework 1 Question 11

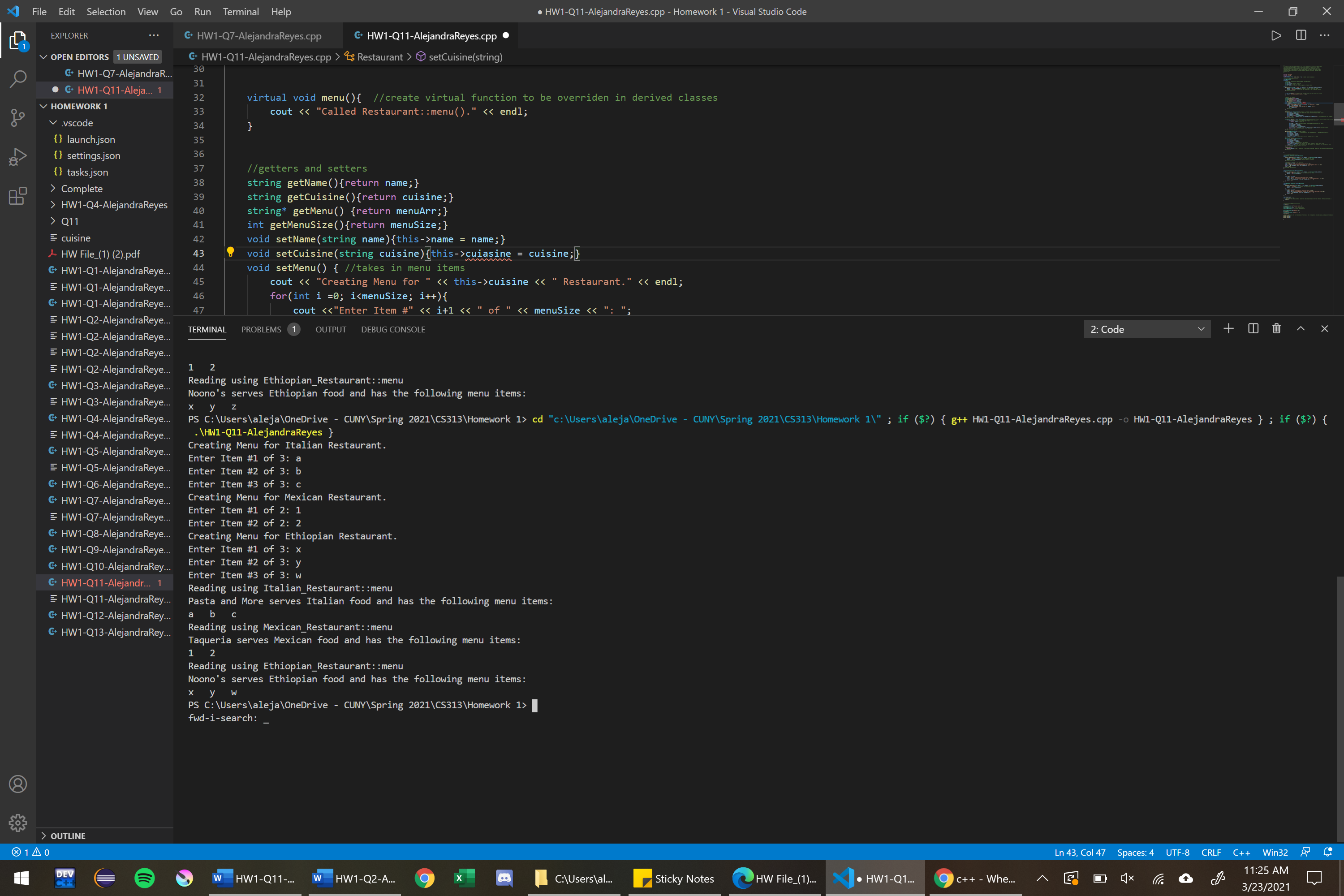
Create a class called Restaurant. Give it the necessary structure, variables, and functions, and create a pure virtual function called menu. Create a class called Italian\_Restaurant, Greek\_Restaurant, and Chinese\_Restaurant as derived classes. Use the parent classes constructor as well as any other variables that need to be created. Create a templated class called Reader\_Robot. It should take in any kind of restaurant and use polymorphism to read the menu (that you overrode). Show how this works in main.

**Program Description:**

The program includes the class Restaurant as a parent class, and child classes Italian\_Restaurant, Mexican\_Restaurant, and Ethiopian\_Restaurant. The parent class includes getters and setters for member variables of the class, a destructor, as well as copy and move assignment operators and constructors. The constructors for the derived classes use the parent class constructor to create a new restaurant with a specified name and menu size. The parent class has a virtual function called menu that is overridden in each of the child classes.

The program also includes a templated function called Reader\_Robot that takes in any type of restaurant, and reads the menu using the overridden menu function. Reader\_Robot uses polymorphism to decide which menu function to call. If the restaurant menu being read belongs to a Mexican\_Restaurant object, the menu function of the Mexican\_Restaurant class is invoked, rather than the menu function of Ethiopian\_Restaurant or Italian\_Restaurant.

Here is a sample output from the program:



**Conclusion:**

Polymorphism is achievable in C++ through the use of virtual functions. A virtual function, once defined in a base class, can be redefined by derived classes. Derived classes can change the implementation of virtual functions present in the base class. Polymorphism allows us to declare a single function that will be implemented differently depending on the object invoking the function.