

# Challenge 1: Override a Method in the Derived Class

Can you override a method in a derived class? A solution is placed in the solution section to help you, but we suggest you try to solve it on your own first.

## WE'LL COVER THE FOLLOWING ^

- Problem Statement
  - Input
  - Output
  - Sample Input
  - Sample Output
- Coding Exercise

## Problem Statement #

When a method in a derived class overrides a `virtual` method in a base class, it is still possible to call the overridden method using the `base` keyword.

If you write `base.Method()` , it will call the method that was defined in the superclass.

The `GetClassName()` method is already implemented in the base class.

You are given a partially completed code in the editor. Modify and override the method `GetClassName()` in the derived class code so that the code returns the following:

## Input #

Derived Class Name

## Output #

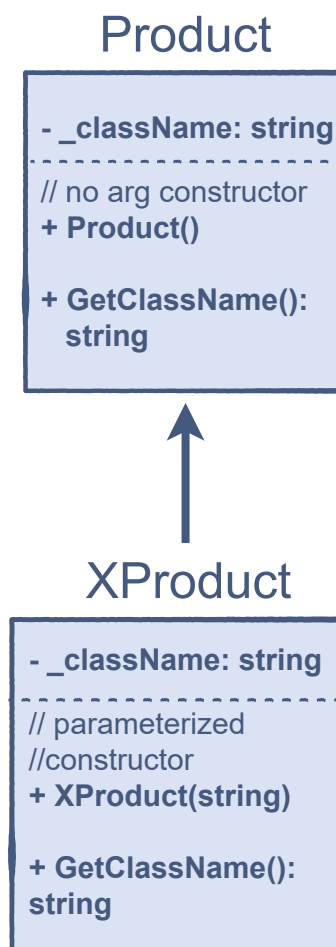
Base Class Name, Derived Class Name

## Sample Input #

```
Product beverage = new XProduct("Beverage");
```

## Sample Output #

```
"Product, Beverage"
```



Based and Derived Classes Structure

## Coding Exercise #

First, take a close look and design a step-by-step algorithm before jumping to the implementation. This problem is designed for practice, so try to solve it on your own. If you get stuck, you can always refer to the solution provided in the solution review.

Good luck!

Uncomment the **lines 4 and 7** before writing your solution.

```
// Derived Class
class XProduct : Product {

    //private string _className; // Uncomment this line

    public XProduct(string className) { // Parameterized Constructor
        //this._className = className;    // Uncomment this line
    }

    //Override and implement the GetClassName method here
}
```



The solution will be explained in the next lesson.