Solution Review: Override a Method in the Derived Class

This review provides a detailed analysis to solve the 'Override a Method in the Derived Class' challenge.

WE'LL COVER THE FOLLOWING

Solution

```
// Base Class
class Product {

  // Private Data Members
  private string _className;

  public Product() { // Default Constructor
        this._className = "Product";
  }

  // Getter Function
  public virtual string GetName() {
    return this._className;
  }

}

// Derived Class
class XProduct : Product {

  private string _className;

  public XProduct(string className) { // Default Constructor
    this._className = className;
  }

  // Overriden Method
  public override string GetName() {
    return base.GetName() + ", " +this._className;
  }

}
```

```
class Demo {

 public static void Main(string[] args) {

   Product beverage = new XProduct("Beverage");
   Console.WriteLine(beverage.GetName());

 }
}
```

Explanation

- **Line 29:** The `base` keyword is used to call the **Base Class** method `GetName()`. With the help of the `base` keyword, the method returns the base class name. Then we append the derived class name to it and return the string.