Solution Review: Implement the Derived Class

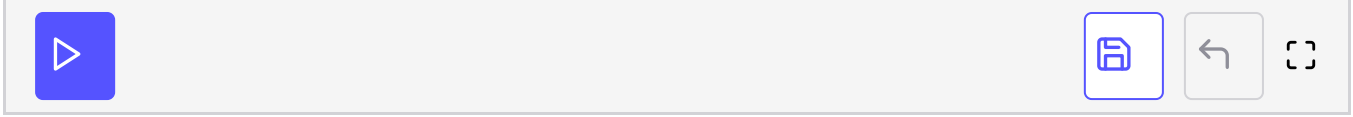
This review provides a detailed analysis to solve the 'Implement the Derived Class' challenge.

WE'LL COVER THE FOLLOWING

Solution

```
// Base Class
class Product {

  // Private Data Members
  private string name;
  private double price;

  // Getter method for name
  public string GetName(int liters) {
      if (liters == 1) {
          this.name = "Cola";
          return this.name;
      }
      else if (liters == 2) {
          this.name = "Fanta";
          return this.name;
      }
      else if (liters == 3) {
          this.name = "Dew";
          return this.name;
      }
      else return "";
  }

  // Getter method for price
  public double GetPrice(int liters) {
      if (liters == 1) {
          this.price = 2;
          return this.price;
      }

      else if (liters == 2) {
          this.price = 3.5;
          return this.price;
      }
```

```
      else if (liters == 3) {
          this.price = 4;

          return this.price;
      }
      else return 0;
  }

}

// Derived Class
class Beverage : Product {

    public int Liters { get; set; } //  Liters of a Beverage

    public string GetDetails() {
        string details = GetName(this.Liters) + ", " + GetPrice(this.Liters) + ", " + Liters;
        return details;
    }

}


class Demo {

    public static void Main(string[] args) {
        Beverage berverage = new Beverage();
        berverage.Liters = 2;
        Console.WriteLine(berverage.GetDetails());
    }

}
```



Explanation

- **Line 47:** `Beverage` class is derived from the `Product` class.
- **Line 51-54:** In the `GetDetails()` method, we call the base class' getter methods by passing them the `Liters` as an argument to get the respective `name` and `price` details of the beverage and prepend them to the `Liters` to output the complete details.