Using the [Range] Attribute with Numeric and Date Data Labs

Perform these labs on your own computer using Visual Studio 2022 or later to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Limit Numeric Values Using the [Range] Attribute

In the Product class you have the **StandardCost** and the **ListPrice** properties that you should apply a [Range] attribute to. You do not want a cost or a price value to be less than zero dollars. Open the **Product.cs** file and locate the **StandardCost** and **ListPrice** properties and add the [Range] attribute as shown below. Be sure to include the **ErrorMessage** property so you can format the cost as currency.

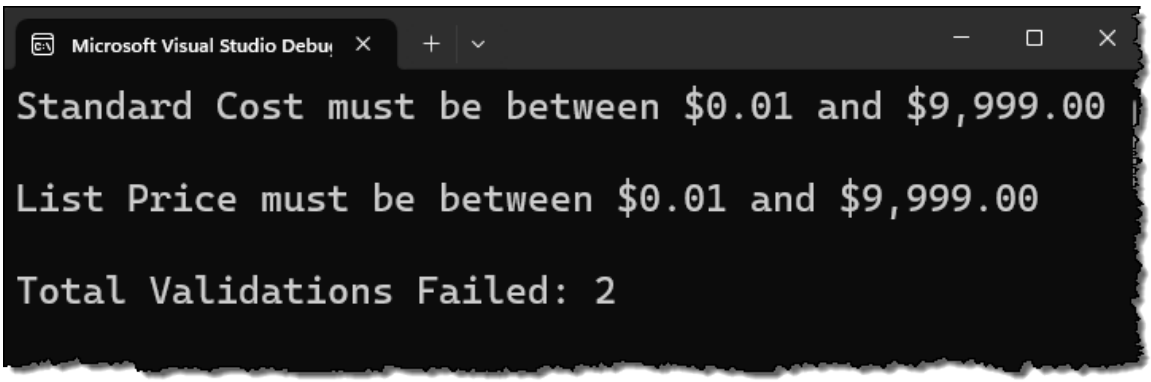
```
[Display(Name = "Standard Cost")]
[Range(0.01, 9999,  ErrorMessage = "{0} must be between
{1:c} and {2:c}")]
public decimal StandardCost { get; set; }
[Display(Name = "List Price")]
[Range(0.01, 9999, ErrorMessage = "{0} must be between
{1:c} and {2:c}")]
public decimal ListPrice { get; set; }
```

Try It Out

Open the **Program.cs** file and change the Name property in the initialization of the entity as shown below.

```
Product entity = new() {
  ProductID = 1,
  Name = "A New Product",
  ProductNumber = "NEW-001",
  Color = "Red",
  StandardCost = 0,
  ListPrice = 0,
  SellStartDate = Convert.ToDateTime("1/1/1999"),
  SellEndDate = Convert.ToDateTime("1/1/2031"),
  DiscontinuedDate = DateTime.Now
};
```

Run the application and view the results as shown below. Notice that the currency values are formatted due to the colon and 'c' in the **ErrorMessage** property "{n:**c**}".



Lab 2: Limit Date Ranges Using the [Range] Attribute

When using the [Range] attribute with numbers, you specify the minimum and maximum values as the first and the second parameters. When you wish to check a property for a date range, you must pass to the first parameter a typeof(DateTime) so the [Range] attribute class knows to check for a DateTime range. Open the **Product.cs** file and add a [Range] attribute to the **SellStartDate** and the **SellEndDate** properties.

```
[Display(Name = "Selling Start Date")]
[Range(typeof(DateTime), "1/1/2000", "12/31/2030",
ErrorMessage = "{0} must be between {1:d} and {2:d}")]
public DateTime SellStartDate { get; set; }

[Display(Name = "Selling End Date")]
[Range(typeof(DateTime), "1/1/2000", "12/31/2030",
ErrorMessage = "{0} must be between {1:d} and {2:d}")]
public DateTime? SellEndDate { get; set; }
```

Try it Out

Open the **Program.cs** file and modify the initialization of the **Entity** object to look like the following code.

```
Product entity = new() {
  ProductID = 1,
  Name = "A New Product",
  ProductNumber = "NEW-001",
  Color = "Red",
  StandardCost = 1,
  ListPrice = 10,
  SellStartDate = Convert.ToDateTime("1/1/1999"),
  SellEndDate = Convert.ToDateTime("1/1/2031"),
  DiscontinuedDate = DateTime.Now
};
```

Run the application and you should see error messages that look like the following.