Create Custom Validations by Inheriting from the ValidationAttribute Class 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: Is Date Greater Than Minimum Date

Right mouse-click on the **ValidationClasses** folder and create a new class named **DateMinimumAttribute**. Replace the entire contents of this new file to the code shown below.

```
using System.ComponentModel.DataAnnotations;

namespace Common.Library;

public class DateMinimumAttribute : ValidationAttribute
{
  public DateMinimumAttribute(string minDate)
  {
    _minDate = Convert.ToDateTime(minDate);
  }

  private readonly DateTime _minDate;

  protected override ValidationResult? IsValid(object?
value, ValidationContext vc)
  {
    // Ensure the object passed in is valid
    if (value != null) {
      // Get the value entered
      DateTime dateEntered = (DateTime)value;

      // Get display name for validation message
      string displayName = vc.DisplayName;

      // If the date entered is less than
      // or equal to the minimum date set
      // return an error
      if (dateEntered <= _minDate) {
        // Check if ErrorMessage is filled in
        if (string.IsNullOrEmpty(ErrorMessage)) {
          ErrorMessage = $"{displayName} must be greater
than or equal to '{_minDate:MM/dd/yyyy}'.";
        }

        return new ValidationResult(ErrorMessage, new[]
{ vc.MemberName ?? "UnknownProperty" });
      }
    }

    return ValidationResult.Success;
  }
}
```

Try it Out

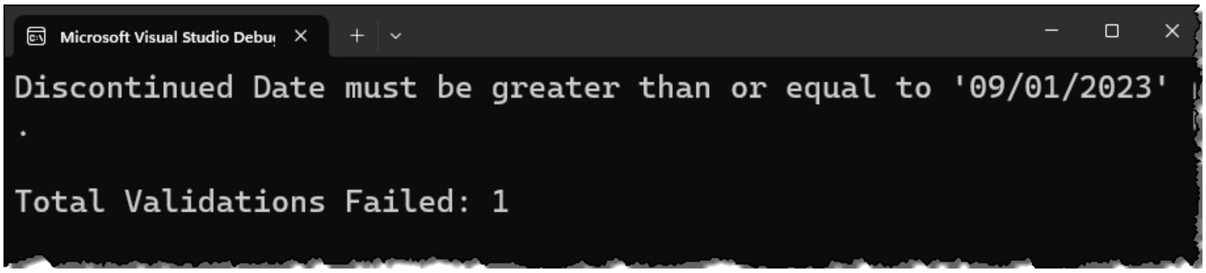
Open the **Product.cs** file and add the **[DataMinimum]** attribute to the **DiscontinuedDate** property as shown below.

```
[DateMinimum("9/1/2023")]
```

Open the **Program.cs** file and create an instance of the Product class and initialize the entity object to the following code. Notice the **DiscontinuedDate** property is set to a date less than the minimum date specified in the [DateMinimum] attribute.

```
Product entity = new() {
  ProductID = 1,
  Name = "A New Product",
  ProductNumber = "PROD001",
  ProductUrl = "https://www.advworks.com",
  Color = "Red",
  StandardCost = 5,
  ListPrice = 12,
  SellStartDate = DateTime.Today,
  SellEndDate = DateTime.Today.AddYears(+5),
  DiscontinuedDate = Convert.ToDateTime("1/1/2020")
};
```

Run the application and view the error message you get back from the **DateMinimumAttribute** class.



```
Discontinued Date must be greater than or equal to '09/01/2023'
.

Total Validations Failed: 1
```

Lab 2: Is Date Less Than Maximum Date

Right mouse-click on the **ValidationClasses** folder and create a new class named **DateMaximumAttribute**. Replace the entire contents of this new file to the code shown below.

```
using System.ComponentModel.DataAnnotations;

namespace Common.Library;

public class DateMaximumAttribute : ValidationAttribute
{
  public DateMaximumAttribute(string maxDate)
  {
    _maxDate = Convert.ToDateTime(maxDate);
  }

  private readonly DateTime _maxDate;

  protected override ValidationResult? IsValid(object?
value, ValidationContext vc)
  {
    if (value != null) {
      // Get the value entered
      DateTime dateEntered = (DateTime)value;

      // Get display name for validation message
      string displayName = vc.DisplayName;

      // See if the date entered is greater than
      // or equal to the maximum date set
      if (dateEntered >= _maxDate) {
        // Check if ErrorMessage is filled in
        if (string.IsNullOrEmpty(ErrorMessage)) {
          ErrorMessage = $"{displayName} must be less
than or equal to '{_maxDate:MM/dd/yyyy}'.";
        }

        return new ValidationResult(ErrorMessage, new[]
{ vc.MemberName ?? "UnknownProperty" });
      }
    }

    return ValidationResult.Success;
  }
}
```

Try it Out

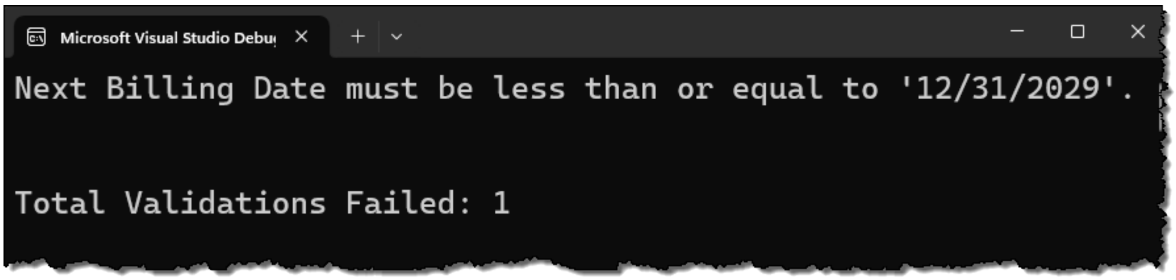
Open the **Customer.cs** file and add to the **NextBillingDate** property the **[DateMaximum]** attribute as shown below.

```
[DateMaximum("12/31/2029")]
```

Open the **Program.cs** file and initialize the **NextBillingDate** property within the entity object to the following code.

```
Customer entity = new() {
  NextBillingDate = Convert.ToDateTime("1/1/2030")
};
```

Run the application and view the error message you get back from the **DateMaximumAttribute** class.



```
Next Billing Date must be less than or equal to '12/31/2029'.


Total Validations Failed: 1
```

Lab 3: Check for a Minimum and Maximum Date

Right mouse-click on the **ValidationClasses** folder and add a new class named **DateYearRangeAttribute**. Replace the entire contents of this new file with the code shown below.

```
using System.ComponentModel.DataAnnotations;

namespace DataAnnotationsSamples;

public class DateYearRangeAttribute :
ValidationAttribute
{
  public DateYearRangeAttribute(int yearsPrior, int
yearsAfter)
  {
    _minDate = DateTime.Now.AddYears(yearsPrior);
    _maxDate = DateTime.Now.AddYears(yearsAfter);
  }

  private readonly DateTime _minDate;
  private readonly DateTime _maxDate;

  protected override ValidationResult? IsValid(object?
value, ValidationContext vc)
  {
    if (value != null) {
      // Get the value entered
      var dateEntered = (DateTime)value;

      // Get display name for validation message
      string displayName = vc.DisplayName;

      // Is date entered within the date range
      if (dateEntered < _minDate ||
          dateEntered > _maxDate) {
        // Check if ErrorMessage is filled in
        if (string.IsNullOrEmpty(ErrorMessage)) {
          ErrorMessage = $"{displayName} must be between
'{_minDate:MM/dd/yyyy}' and '{_maxDate:MM/dd/yyyy}'.";
        }

        return new ValidationResult(ErrorMessage, new[]
{ vc.MemberName ?? "UnknownProperty" });
      }
    }

    return ValidationResult.Success;
  }
}
```

Try it Out

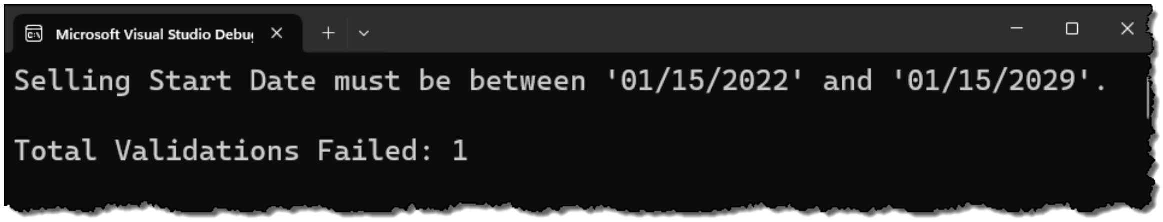
Open the **Product.cs** file and remove the **[Range]** attribute from the **SellStartDate** property. Add to the **SellStartDate** property the [DateYearRange] attribute as shown below.

```
[DateYearRange(-2, 5)]
```

Open the **Program.cs** file and initialize the entity object to the following code. Notice the **SellStartDate** property is set to six years prior to today's date. This will cause the [DateYearRange] attribute to fail the validation.

```
Product entity = new() {
  ProductID = 1,
  Name = "A New Product",
  ProductNumber = "PROD001",
  ProductUrl = "http://www.advworks.com",
  Color = "Red",
  StandardCost = 5,
  ListPrice = 12,
  SellStartDate = DateTime.Today.AddYears(-6),
  SellEndDate = DateTime.Today,
  DiscontinuedDate = DateTime.Today
};
```

Run the application and view the error message you get back from the **DateYearRange** class.



```
Selling Start Date must be between '01/15/2022' and '01/15/2029'.

Total Validations Failed: 1
```