Override ToString() in Your C# Classes Labs

Perform these labs on your own computer using Visual Studio 2022 or later, or VS Code 1.8x or later, to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: Create a Product Class and Display Default ToString()

Open **Visual Studio** or **VS code** and create a new **console** application named **ToStringSamples**.

Add a new class named **Product**. Make the class look like the following.

```
namespace ToStringSamples;

public class Product
{
  public string ProductName { get; set; } =
string.Empty;
  public int ProductId { get; set; }
}
```

Modify the **Program.cs** file to look like the following.

```
using ToStringSamples;

Product entity = new() {
  ProductId = 1,
  ProductName = "Acme Bazooka"
};

Console.WriteLine(entity);
```

Try It Out

Run the application and you should see the following displayed in the console window.

```
ToStringSamples.Product
```

Lab 2: Override ToString()

Open the **Product.cs** file and add an override of the ToString() method.

```
public override string ToString()
{
  return ProductName;
}
```

Try It Out

Run the application to you should now see the actual product name appear in the console window.

Lab 3: Using an Interpolated String in the ToString() Override

The problem with the ToString() override you just wrote is it does not uniquely identify the product. Open the **Product.cs** file and modify the ToString() method to look like following.

```
public override string ToString()
{
  return $"{ProductName} ({ProductId})";
}
```

Try It Out

Run the application to you should now see the product name and the product id appear in the console window.

```
Acme Bazooka (1)
```