Inheritance Lab

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

Lab 1: Create Person Class and Refactor Customer

Right mouse-click on the OOPLab project and add a new class named **Person**.

Make the file look like the following:

```
namespace OOPLab;

public class Person {

}
```

Open the **Customer.cs** file and cut the **FirstName** and **LastName** properties from this class and move them to the **Person** class.

Add another property to the **Person** class named **Age**

```
public int Age { get; set; }
```

Move the FullName() method from the **Customer** class to the **Person** class too.

Inherit from the **Person** class within the **Customer** class

```
public class Customer : Person {
```

Try it Out

Run the application and ensure the output is the same as in the last lab.

```
2 - John Smith
Smith, Inc.
John.Smith@smithinc.com
$75,000.00
Smith, John
```

Lab 2: Add Init() to Person Class

Open the **Person.cs** file and add a virtual Init() method.

```
public virtual void Init() {
  FirstName = string.Empty;
  LastName = string.Empty;
}
```

Open the **Customer.cs** file and modify the Init() method to look like the following.

```
public override void Init() {
  base.Init();

  CustomerId = 1;
  CompanyName = string.Empty;
  EmailAddress = string.Empty;
  _CreditLimit = 50000;
}
```

You call the Init() method in the base class to initialize the **FirstName** and **LastName** properties.

Try it Out

Run the application and ensure the output still looks like the output from the last lab.

```
2 - John Smith
Smith, Inc.
John.Smith@smithinc.com
$75,000.00
Smith, John
```

Copyright © 2002-24 by Paul D. Sheriff
All rights reserved worldwide. Reproduction is strictly prohibited.

Lab 3: Make FullName Virtual

Open the **Person.cs** file and add the **virtual** keyword to the FullName() method.

```
public virtual string FullName() {
  return $"{LastName}, {FirstName}";
}
```

Open the **Customer.cs** file and replace the FullName() method you previously wrote with an override that returns the company name with the last name and first name.

```
public override string FullName() {
  return $"{CompanyName}: {LastName}, {FirstName}";
}
```

Try it Out

Run the application and see the new FullName() override.

```
2 - John Smith
Smith, Inc.
John.Smith@smithinc.com
$75,000.00
Smith, Inc.: Smith, John
```

Lab 4: Override the ToString() method

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

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

Open the **Customer.cs** file and override the ToString() method

Copyright © 2002-24 by Paul D. Sheriff
All rights reserved. Reproduction is strictly prohibited.

```
public override string ToString() {
  return $"Customer ID: {CustomerId} - {FullName()}";
}
```

Try it Out

Open the **Program.cs** file and at the bottom of the file add a new line

```
Console.WriteLine(entity.ToString());
```

Below this code create an instance of a **Person** class and display the ToString() of that object.

```
Person person = new() {
  FirstName = "Sally",
  LastName = "Seashell"
};
Console.WriteLine();
Console.WriteLine(person.ToString());
```

Run the application and ensure the output is the same as show below.

```
2 - John Smith
Smith, Inc.
John.Smith@smithinc.com
$75,000.00
Smith, Inc.: Smith, John
Customer ID: 2 - Smith, Inc.: Smith, John

Seashell, Sally
```