Methods 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 Void Method

Since you added a read-only property in the last lab, you need to create some way to set the **CreditLimit**. Create a new method called SetCreditLimit to perform this operation.

Open the **Customer.cs** file and add the following method.

```
public void SetCreditLimit(decimal limit) {
  _CreditLimit = limit;
}
```

Try it Out

Open the **Program.cs** file and add the following line of code after the initialization of the Customer object.

```
entity.SetCreditLimit(100000);
```

Add a new Console.WriteLine() at the end of the file.

```
Console.WriteLine(entity.CreditLimit.ToString("c"));
```

Run the application and the Credit Limit should display as **$100,000.00**.

Lab 2: Create Method to Return a Value

Create a method named FullName() to return the last name and first name of the customer.

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

Try it Out

Open the **Program.cs** file and add the following line of code at the end of the file.

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

Run the application and the full name should be displayed as **Smith, John**.

Lab 3: Create Constructor

Open the **Customer.cs** file and immediately after the class definition, add the following code:

```
public Customer() {
  CustomerId = 1;
  FirstName= string.Empty;
  LastName= string.Empty;
  CompanyName= string.Empty;
  EmailAddress= string.Empty;
  _CreditLimit = 50000;
}
```

Try it Out

Open the **Program.cs** file and comment out the line of code that calls the SetCreditLimit() method.

```
//entity.SetCreditLimit(100000);
```

Run the application and you should now see the credit limit property displayed as **$50,000.00**.

Lab 4: Create an Overloaded Constructor

Open the **Customer.cs** file and add a new constructor that looks like the following:

```
public Customer(int id, decimal limit) {
  CustomerId = id;
  _CreditLimit = limit;
}
```

Try it Out

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

```
Customer entity = new(2, 75000) {
  FirstName = "John",
  LastName = "Smith",
  CompanyName = "Smith, Inc.",
  EmailAddress = "John.Smith@smithinc.com"
};
```

Run the application and view customer id and the credit limit as shown below:

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

Lab 5: Create an Init() Method

Open the **Customer.cs** file and create an Init() method just below the constructors.

```
public void Init() {
  CustomerId = 1;
  FirstName = string.Empty;
  LastName = string.Empty;
  CompanyName = string.Empty;
  EmailAddress = string.Empty;
  _CreditLimit = 50000;
}
```

Modify the two constructors to look like the following:

```
public Customer() {
  Init();
}

public Customer(int id, decimal limit) {
  Init();
  CustomerId = id;
  _CreditLimit = limit;
}
```

Try it Out

Run the application and view the output to ensure you get the same values as in the last lab.