Turn your attention to the `Main()` method in the default `Program` class and create an instance of the `Dog` class

```csharp
Dog fido = new Dog();
```

Next, call the new instance object’s setter method to initialize all its variable members

```csharp
fido.setValues("Fido", 3, "Brown");
```

Now, retrieve all properties of the new object

```csharp
string tagF = String.Format("{0} is a {1} year old {2} dog", fido.getName(), fido.getAge(), fido.getColor());
```

Display all properties and call the miscellaneous method

```csharp
Console.WriteLine( tagF + fido.bark() );
```

Now, create another instance of the `Dog` class

```csharp
Dog lucy = new Dog();
lucy.setValues("Lucy", 2, "Gray");
```

Next, retrieve all properties of this new object

```csharp
string tagL = String.Format("{0} is a {1} year old {2} dog", lucy.getName(), lucy.getAge(), lucy.getColor());
```

Display all properties and call the miscellaneous method

```csharp
Console.WriteLine( tagL + lucy.bark());
Console.ReadKey();
```

Press `Start` or `F5` to see each object’s properties

```csharp
Fido is a 3 year old Brown dog
Woof, woof!

Lucy is a 2 year old Gray dog
Woof, woof!
```

Notice here how the `String.Format()` method is coded using newlines to clearly build a `string` value for output.