Creating Class Properties

- Set a new property of a form class
  - Need a private class-level variable to store the value
    - Available only to methods and property blocks within the class
  - Need public property methods to allow other classes to view and/or set the property

- When program creates objects from a class
  - Need to assign values to the properties
  - Class controls access to its properties through property methods

Creating Properties in any Class

- Set a new property of a form class
  - Need a private class-level variable to store the value
    - Available only to methods and property blocks within the class
  - Need public property methods to allow other classes to view and/or set the property

- When program creates objects from a class
  - Need to assign values to the properties
  - Class controls access to its properties through property methods

Property Blocks

- A class allows its properties to be retrieved or set with accessor methods in a property block
  - Get accessor method retrieves a property value
    - Similar to a method declared with a return value
    - Before the closing brace a return value must be assigned
  - Set accessor method assigns a value to the property
    - Uses the value keyword to refer to the incoming value for the property

- Property block declared as public to allow external objects to access it
- Data type of incoming value for a set must match type of the return value of the corresponding get
The Property Block – General Form

```csharp
// Class-level variable to hold the value internally.
private DataType MemberVariable;

public DataType PropertyName
{
    get
    {
        return MemberVariable;
    }
    set
    {
        // Statements, such as validation.
        MemberVariable = value;
    }
}
```

Holds value of property internally
Property name to "outside world". Use a "friendly" name
Keyword that refers to the incoming value assigned to the property

The Property Block – Example

```csharp
private string lastNameString;
public string LastName
{
    get
    {
        return lastNameString;
    }
    set
    {
        lastNameString = value;
    }
}
```

Name of property
Retrieve the value of the property
Assign a value to the property

Read-Only Properties

- In some instances, a property can be retrieved by an object but not changed
- Write a property block that contains only a `get` (creates a read-only property)

```csharp
// Private class-level variable to hold the property value.
private decimal payDecimal;

// Property block for read-only property.
public decimal Pay
{
    get
    {
        return payDecimal;
    }
}
```
Write-only Properties

- A property that can be assigned by an object but not retrieved
- Write a property block that contains only a `set` (creates a write-only property)

```csharp
private decimal hoursDecimal;

// Property block for write-only property.
public decimal Hours
{
    set
    {
        hoursDecimal = value;
    }
}
```

To maintain the value of a property for multiple instances of a class
- Declare the property as `static`
- Static variables retain their value for every object instantiated using the `new` keyword

- `Static` keyword is used on both the private variable and property method
  - A `static` keyword in a property method creates a public property that can be accessed from anywhere in the project
  - A specific object of the class does not have to be instantiated