Object Language Specifics

SOME NOTES ON LANGUAGE SPECIFICS

Properties – what happened to getters and setters

The C# language has special methods called Properties

Properties allow access to private members
- Read
- Write
- Compute

They provide encapsulation from direct member access.

They should be used in place of *getters and setters*

DO NOT USE GETTERS AND SETTERS IN THIS COURSE! Use properties instead. Using a getter and setter like below will result in point loss.
- *public string GetName()*

Pros of Properties

Usable as lambda expressions as opposed to values

- *In C# useful with LINQ or extensions*
- *During debugging breakpoints can be triggered when a property changes*
- *Many libraries use Properties in place of getters/setters*
  - *Serialization*
  - *WPF*
  - *Mocking*

DO NOT USE GETTERS AND SETTERS IN THIS COURSE! Use properties instead. Using a getter and setter like below will result in point loss.

- *public string GetName()*

Property Elements

```
public type Name
{
        get
        {
                return <something>
        }
        set
        {
                <something> = value;
        }
}
```

type is the object type of the getter and setter

get Properties must *return* something.

value is a keyword of the element

Auto-Properties

```
public string Name
{
        get;
        set;
}
```

An auto-property is a property with an implied private member variable.

In the case to the left, the private member would be

private string m_name

Other Valid Properties

// Read only

```
public string Name
{
        get
        {
                return <something>;
        }
}
```

// Read only but accessible (settable) from within the class

```
public string Name
{
        get;
        private set;
}
```