**Oriette Ramos**

**Lesson Plan 18**

**ISTA-220**

1. What is the difference between a property and a field?  
   **A property is a cross between a field and a method—it looks like a field but acts like a method. You access a property by using exactly the same syntax that you use to access a field. However, the compiler automatically translates this field-like syntax into calls to accessor methods (sometimes referred to as property getters and property setters). One key difference is that a field can be assigned a value without the object being initialized whereas a property cannot be assigned a value until the object is initialized. The most important difference is that a property is protected against direct access.**
2. What is the difference between a property and a method?  
   **See the answer above; basically a property is designed to store data like a field, while implementing accessor methods to keep the field data private while allowing public access to the methods that can modify it.**
3. What is your understanding of encapsulation?  
   **The way I understand encapsulation is that the goal is to keep the data and implementation hidden/private. For example, I can call the ToString method for an integer understanding that it converts an integer to a string version without needing to know what is actually being done in that method to accomplish the result.**
4. Some languages are case insensitive, that is, an "a" and an "A" are considered to be the same letter. C# is case sensitive. What implications does this have regarding the naming of variables, methods, and other identifiers? Do you think that the difference in case in the initial character of two different identifiers is sufficient to distinguish them?  
   **While the case sensitive variation is allowed, it is usually bad practice for readability and maintenance reasons. I avoid using case sensitivity as the only differentiating feature of a variable or method name.**
5. Give an example that is not in the book of an instance where you might want to use a read-only property. Give an example not in the book of an instance where you might want to use s write-only property.  
   **An example of a good use for a read-only property is a customer's current balance. You want them to be able to see how much they owe, but not allow them to change that value. Write-only properties have many applicable uses, especially with encapsulation. One specific example is providing a seed to the random number generator for the purpose of encryption.**
6. Can you think of a reason why you might ever want to make getters and setters private? Give an example. Also, make a case why getters and setters should never be private.  
   \*\*Making getters or setters private are a way to make a property read-only or write-only, especially when generating automatic properties. For example, if you had an interface for username and passwords, the interface might contain a property like string Password{ private get; set; }. I can't think of a reasonable use for both get and set to be private, because at that point you could just use a private field and use a private method in the class to do whatever manipulation needed to be done with it. I cannot make a case that getters and setters should never be private. Based on my knowledge of application security, if a method of altering data exists that was not meant to me accessed outside the class, it can potentially be exploited. \*\*
7. What are restrictions on the use of properties?  
   **The restrictions on the use of properties are:**
   * **You can assign a value through a property of a structure or class only after the structure or class has been initialized.**
   * **You can’t use a property as a ref or an out argument to a method (although you can use a writable field as a ref or an out argument). This makes sense because the property doesn’t really point to a memory location; rather, it points to an accessor method.**
   * **A property can contain at most one get accessor and one set accessor. A property cannot contain other methods, fields, or properties.**
   * **The get and set accessors cannot take any parameters. The data being assigned is passed to the set accessor automatically by using the value variable.**
   * **You can’t declare properties by using const.**
8. What is an object initializer? What is the syntax for an object initializer?  
   **The keyword new, combined with a class constructor is an object initializer. The syntax for object initialization is new MyObject() and is usually used in conjunction with an assignment operator like MyObject thing = new MyObject();.**