Quiz #2 Answers

1. Contrast variables, parameters, and properties. Examples of uses might help.
You only had to contrast the three, but I'll compare first for educational value. All three of these elements of the programming language are basically chunks of memory where we can store information and give it a name. We get to pick the type of the information we are storing as well as the name we want it called.

So in what ways are they different? They differ in scope and function. A variable has a local scope and only exists inside the method/function we declare it in. We use them just for simple calculations to store something for us for a short period of time. Parameters also have local scope and inside of a method/function a parameter acts just like a variable. However, from outside the method a parameter is something we can use to give information to the method/function. When we call the method/function we have to provide the value that we want the method to have for that parameter. This gives us the ability to write much more flexible methods/functions that can do something for us in multiple situations. A property doesn't have the local scope. It is basically like a variable that we can attach to a whole object. Because it is part of the object, it gets remembered as long as the object is in the world and can be accessed across multiple methods/functions. We typically use these to store information related to the object that needs to "live" longer than just through one method/function call.

2. You are writing a program in which two objects play "hide and go seek". At a certain point in the program the object that is seeking needs to do something special if it is close to the object that is hiding and the hiding object is in front of it, otherwise it keeps on seeking. Outline how you would get this functionality in Alice.
We basically have an if statement here. If the seeker if close to the hider and facing the hider do one thing, otherwise do something else. To use the full syntax of an if it might look like this.

```
if(seeker.isCloseTo(3,soughtObject) && soughtObject.isInFrontOf(seeker)) {
        code for finding
} else {
        code to keep seeking
}
```

Extra Credit: Give an example of validating parameters with an if statement.
The example in the book is that the jumping fish can only jump so far and you can't pass it negative distances. Other classic examples in programming are when you do division making sure the denominator isn't zero or when taking a square root making sure the value isn't negative.