Ungraded Quiz on OO concepts – what do you know?

**Don’t give too much detail. Your answers do not have to be long. Don’t try to look answers up (unless you want to). Just tell me what you know.**

1. In the Board class you should have declared the instance variables as private. Why are instance variables typically declared private while the methods are typically public?
   1. Instance variables are typically declared private because only the programmer should have access to those. The methods are what you want the users of the program to be using, so therefore those are made public.
2. What does it mean to instantiate a class? Give a line of code that is an example of this (instantiate the Board class for example)
   1. To instantiate a class means to create that class as an object to be used later. An example could be:
   2. Board gameBoard = new Board(xDim, yDim, numHotSpots);
3. What is the purpose of the keyword static? The main method is declared as static – why?
   1. The purpose of the keyword static is to make the method declared at the class level. This allows us to save memory and other things as we don’t have to create a new instance when we want to use it.
4. What does new actually do?
   1. I’m not entirely sure, but I think new just creates a new object of whatever you are calling.
5. You are asked to create two Board constructor methods. What is the term we use for declaring two methods with the same name, but different sets of parameters?
   1. I am not sure if this is what you’re referring to, but there is something called method overloading, where you can submit different amounts of parameters to the “same” method. But if you’re asking about just different values of parameters to the same method, I guess I’m not entirely sure.
6. What is the difference between .equals methods and using ==?
   1. Well part of it you already said, the .equals is a method, while the == is an operator. But the .equals compares the contents, while the == compares the memory addresses.
7. Suppose our game was going to also have a Player class and there were going to be 2 types of Players. How could we use inheritance to define our classes?
   1. I’m not entirely sure, but basically we could create a class that has all the main components that we want all players to be able to accomplish, and then we could use inheritance to create two different classes that might behave differently.