William Escamilla HW2 WPs

Weiss 3.2, 3.4, 4.1, 4.9, 5.11.

3.2) Public methods and variables of a class can be used and accessed by methods not associated with that class itself. For instance, if there is a private variable ‘name’ in a class, to access that variable’s data you would need to use a ‘getter’ method defined in that class… you cannot access it directly.

3.4) If you don't write the constructor explicitly, compiler will generate a ‘no-args’ constructor by default which then calls the ‘no-args’ constructor of its superclass

4.1) The derived class can use methods from the inherited class to access private variables in the inherited class but cannot outright access the private variables themselves. Of course, the derived class can also use anything that is public from the inherited class. As said above, private functions become public for users of the derived class so that private variables can be accessed

4.9) An interface is a blueprint or contract for all classes that implement it. All methods in an interface are abstract, where an abstract class can have some non-abstract method definitions. A class can extend only one abstract class, but a class can implement more than one interface. An interface cannot declare private, protected, transient or volatile variables.

5.11) Θ(N)