Chapter 8 CRT

Name: Youdis

1. An Is-a relationship is just inheritance a class inheriting another one so it becomes a subclass of it. For example a Jaguar is an animal so a Jaguar object would inherit from the animal class because it “is-a” animal. A has-a relationship is when an object is made up of different components. For example rooms have tables so when creating a class object a table object will be created within it. Meaning the classroom object “has-a” table object in it.
2. Both methods will be available to the object since the baseclass method is inherited to the subclass meaning the object has access to it. While the method created in the subclass is already available to the object as the object is built upon the classes methods and variables inherited and created in the class.
3. Implementing an abstract method is taking a method from the implemented interface that is defined but has no body and giving the method a concrete definition in the subclass. While overriding a method is copying the superclasses method’s signature in the subclass and giving it a new definition so when called through a subclass object the definition stated in the subclass is performed. They both are inherited and both require writing the method signature and adding something to the method body. Implementing a method is mandatory if there is one to be implemented from the superclass while overriding a method is not mandatory for a class.
4. An abstract class defines what an object is while an interface can define what the object can do via methods. The abstract class as a constructor which can be accessed via derived subclasses while an interface can’t have a constructor method. Also classes can only extend a singular abstract class while they implement many interfaces.
5.
6.
    a. It is an abstract method which is declared but has no implementation
    b. Wo is an interface which defines contracts that implementing classes must implement via concrete definitions
    c. Because it implements Wo and Wo has the abstract method doThat() so Roo must fulfill by the contract by adding implementation to doThat()

d. Roo(), doThat(), doThis(), doNow(), and Bo() constructor method through super().
e. The implementation in Roo overrides the implementation in Bo meaning the implementation stated in the Roo class while being executed instead of the one in the Bo class when called through a Roo object.
f. It calls the constructor method of the superclass in which it is derived from.
g. Yes it can by using super.doThis() this calls the superclasses version of the method rather than the classes version of the method.
h. No it cannot.