1. A has-a relationship between classes means that a class has an instance of another class like as an example a plane class may have an instance of a propeller class which shows the plane has a propeller. A is-a relationship is where a class is a subtype of another class such as a human is a subclass of mammal so if “human” is a subclass of “mammal” its saying that a “human” is a mammal.
2. An object of a derived class will have access to the both of these methods because the derived class will inherit “go()” from the base class. It also has its own method “stop”
3. When a class inherits from an abstract class it has to show a implementation for every abstract method in which is written out in the abstract class and thus abstract methods will be declared with no implementation in the abstract class. Overriding occurs at the time when a derived class gives a specific implementation for a method in which has already been defined in its base class which means the method in the derived class will have the same signature of the method in said base class.
4. Abstract classes can have member variables and an interface cannot. An interface can only have abstract methods where as an abstract class can have abstract methods and concrete or “normal” methods. Finally in abstract classes a class may only inherit from **one** abstract class where as an interface can have a class implement **multiple** interfaces

    6.
    1. An abstract method
    2. An interface
    3. Because Roo implements Wo (the interface)
    4. doThat() doNow() doThis() and any methods inherited from the object
    5. The implementation in roo overrides the implementation in Wo
    6. This is used to call a constructor from a class via inheritance
    7. Yes it can by using the super to call it from the roo object
    8. Yes a method in roo can call the method in Wo using super