Questions 1 - 4, 6

1
Has-a means that a class has an instance of another class in it, for example the account class makes and uses the customer class within it. Is-a describes a superclass-subclass relationship that means inheritance and using abstract and extends when making the classes.

2
The derived class will be able to use go() and stop() since it inherits it from the superclass/base class.

3
Implementing an abstract method

- When the body for the method has been declared abstract in the superclass.
- Ex: in the vehicle superclass the method, abstract String honkSound();.
- These methods are “empty”/ they have no code, and must be implemented in a subclass.
- Ex: In the truck subclass,
  public String honkSound()
  {
  return(“BEEP”);
  }

Overriding a method

- A new implementation of the abstract method in the superclass such as using the honkSound method in a car class and returning “beeeeep” instead of “BEEP”.
- The method still has the same structure like the parameters, return type, and name.

Similarities

- Both use the same outline of an abstract method (parameters, return type, and name) but may return something different such as a different String.

4
Abstract classes can’t be instantiated whereas interfaces have methods a class must implement and are abstract by default. Abstract classes can have public and private methods but interfaces can only have public methods. Variables in abstract classes can change, they can be final, static or non-final, non-abstract but interfaces only have constant variables (public, static, final). Interfaces also can’t have constructors, they’re just methods.

6

a)
Wo is abstract and public.

b)
Wo is an interface.

c)
It's implemented to returns n integer but since it's private and not public the method won't work.

d)
doThat()? doThis(), doNow()

e)
The implementation of doThis() in Roo overrides the implementation of doThis() in Bo. This is method overriding. doThis() stays the same in Bo but is overridden to fit the needs of Roo.

f)
What it should do is: private int x in Bo is set to 1 since int z in Bo's constructor is set to 1 because of the keyword super in Roo.

But since there's no () after Roo in the constructor it just leads to an error and doesn't do anything.

g)
No.

h)
Yes it can.