You can find multiple choice review questions (when they are available) in D2L under:
`Assessments->Quizzes`

For all questions, unless otherwise specified assume that there are no syntax errors in any programs or program fragments.

Short answer (code tracing):

JT's hint: Before looking at the answer make a real attempt at providing an answer. If you are drawing a blank then it indicates that you need to study the material from lecture as well as looking at the in class notes that you should be taking as you are following along in class. You will get far more out of the process if you try it out as if you were writing an actual exam question rather than just directly looking at the answer.

(Solution is on the next page)

#Solution:
```
WORTHLESS = 0

class Company:
    def __init__(self,aName,newNetWorth):
        self.name = aName
        self.netWorth = newNetWorth

    def __str__(self):
        aState = ""
        aState = aState + "Company: " + self.name
        aState = aState + "  " + "Worth $" +
            str(self.netWorth)
        return(aState)
```

The 'self' is needed to access the attributes (or methods) of the class whose 'takeOver' method has been called. (Review the recorded video lecture covering Object-Orientation if you need a refresher). The attributes (or methods) of other comparator object can be accessed since it's passed as a parameter. Access to the comparator object occurs via the name of the parameter (in this case the identifier name is 'target').

```
    def takeOver(self,target):
        self.netWorth = self.netWorth + target.netWorth
        target.netWorth = WORTHLESS
```

JT: not much coding is required but the student must be familiar with how to access attributes of an object within that class definition. And one must know why the self reference is needed inside the methods but also how it works.

The gist of the algorithm for adding the worth from one variable to another is trivial and that comes from the early part of the course (introduction to programming: accessing identifiers and basic math and other operators).