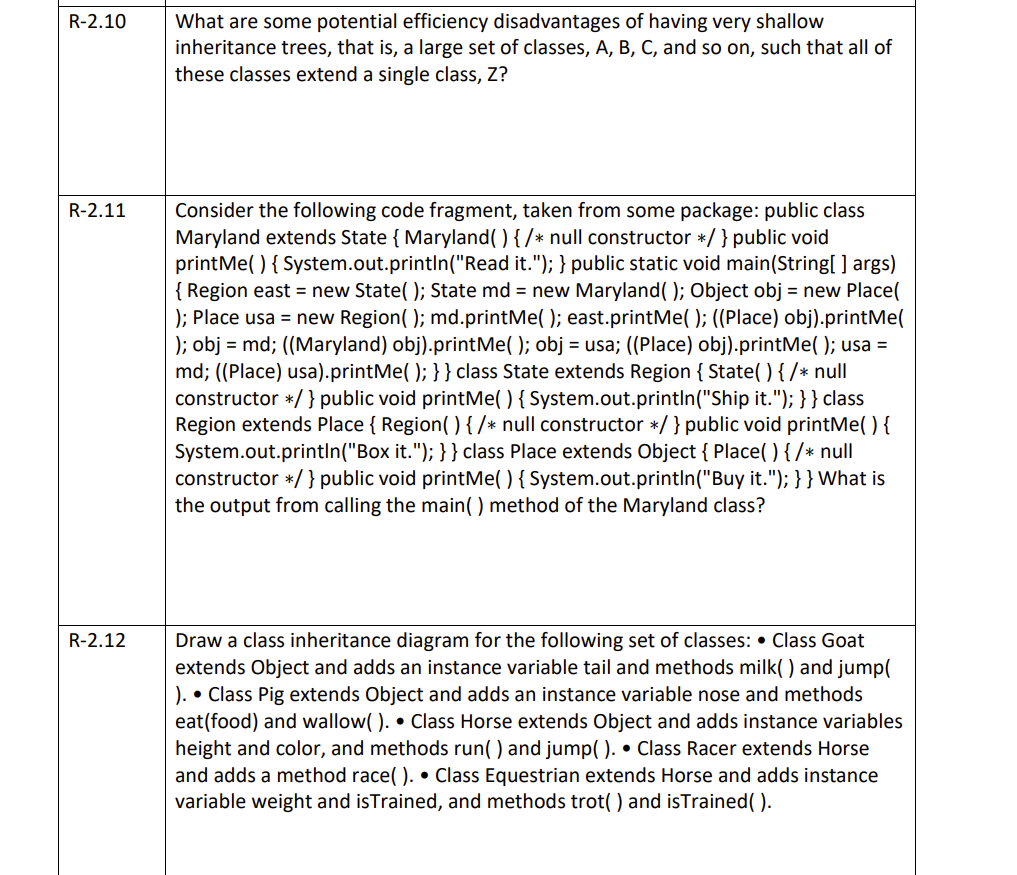


The only problem I can see is all the classes extends to z, becomes tightly coupled and if there is a chance to optimize the code in z class, it becomes cumbersome for developers, as they have the change the code in all extending classes. As a result, the code becomes less performant.

0 0



Read it.

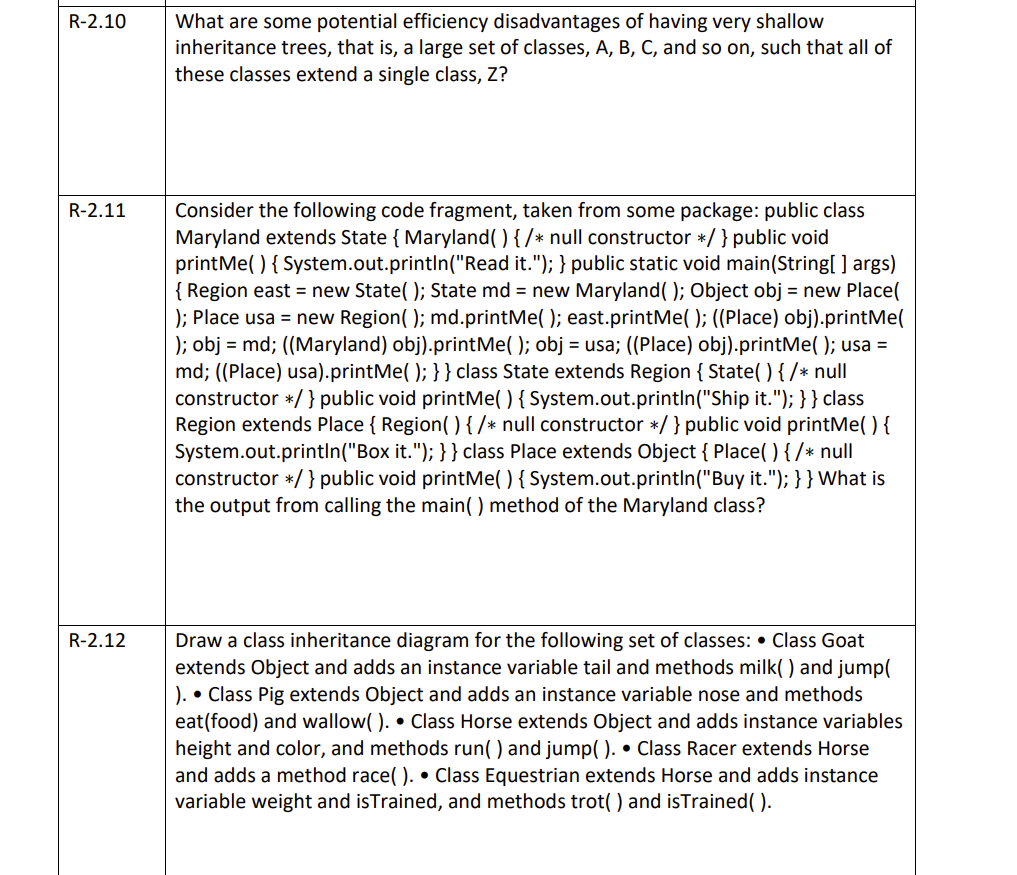
Ship it.

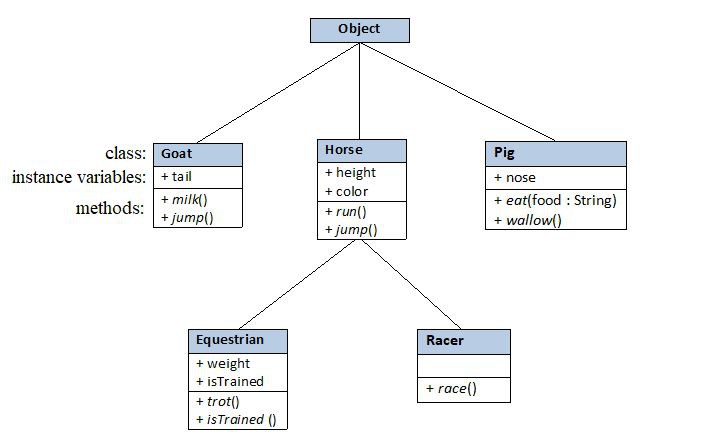
Buy it.

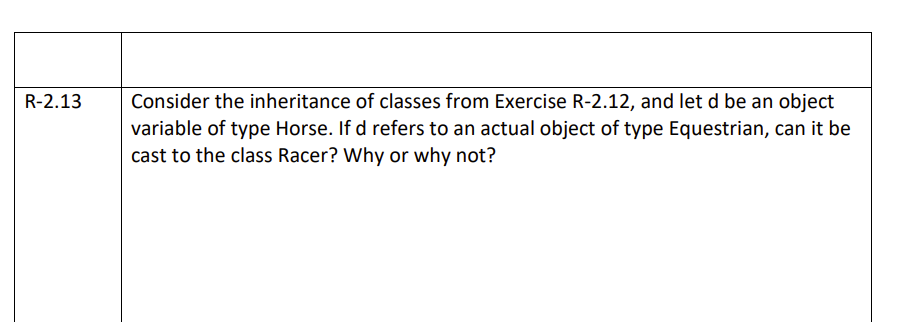
Read it.

Box it.

Read it.

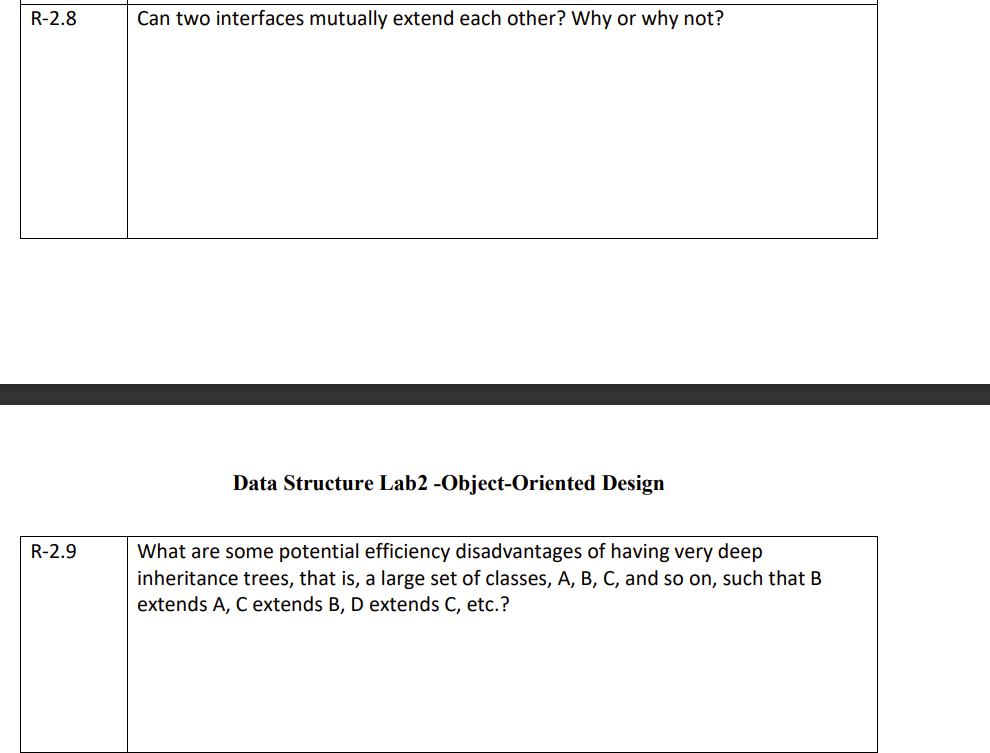






NO,

Beacause Racer and Equestrain are diffrent types,   
         you can resolve the compilation error by Racer extends Equestrain, now Racer becomes   
         parent class to the Equestrain, and you can cast from Equestrain to Racer.



1. One of the drawbacks of deep inheritance is that the classes (parent and child) become tightly coupled.

As a result, when we change the code of the parent class, all of the child classes inheriting from the parent class will be affected, and they cannot function independently.

2. The compiler will take longer to figure out which method is overridden if the method signature is overridden in each class.

3. Classes that extend one another become disorganized.

4. Moreover, it does not support multiple inheritance in classes, but it can be done through interfaces.

Deep inheritance is a bad practise,

Two interfaces can not extend to each other, However an interface can extend multiple interfaces.

Why:

When an interface extends another interface, first one becomes child interface and the later becomes parent interface

in object oriented programming, parent interface doen't mean to inherit child interface methods, if parent interface extends child interface and vice versa, it throws error saying hierarachy of interfaces is inconsistent.

parent interface contains universal methods and child inteface have relatively specific methods.