Vincent Curran
OOAD Project 1 Part 1
January 13th 2022

1) Provide a definition of the OO term “class”

A class is a data type. It is defined by the user and has both data members and methods.

Source: https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/

2) Select any three of the following six OO terms: abstraction, encapsulation, polymorphism, coupling, cohesion, identity
Provide:

a) A definition of the term
b) How the term applies to the OO notion of a class
c) What is a possible positive result of applying the term to a class design
d) What is a possible negative result of applying the term incorrectly in a class design

Abstraction: A set of methods provided to solve a particular problem, so that only the important things are displayed to the client. This is done in a class by setting common variables or methods in a superclass. Using abstraction means you need to depend on less concrete classes, which is a pro. If you use it wrong you could end up exposing extra things to the user which you don’t want them to have access to. Also, you could end up with objects that don't do what you think they should.
Encapsulation: Details of the actual language that hide implementation, such as having private variables. For a class, it means that all the data is private and can only be accessed through methods provided by the class. This is good because it means you can restrict clients from changing variables they shouldn't have access to. If implemented poorly, it could lead to an unusable implementation if you don’t provide methods to work with the variables the class could be useless.
Polymorphism: The concept of treating objects as a superclass but having their implementation come from its subclass. For a class this means making a superclass with multiple different subclasses, but creating them in such a way (ie factory), that the client can access the objects as the superclass, while the implementation follows for the objects subclass. This can be good because it allows object creation to be much simpler, also you can write code just for the superclass instead of for each subclass if they had been concrete. It can be bad if done wrong because you may have objects that aren't related enough to come from the same superclass, then it will be hard to keep them straight and used correctly.

Sources:
Class PPT
Head First
https://www.geeksforgeeks.org/abstraction-in-java-2/
https://www.geeksforgeeks.org/encapsulation-in-java/