1. What are the four pillars of Object-Oriented Programming? Explain each pillar.
    a. **Abstraction – exposing essential bits of information while hiding details**
    b. **Encapsulation – getter/setter methods – can change attributes through method calls rather than directly.**
    c. **Inheritance – allows objects or classes to inherit from parent classes**
    d. **Polymorphism – any child class object can take any form of a class in it's parent hierarchy**

***** from class video**

2. What is the relationship between a Class and an Object?
    a. **A class is a blueprint and the Object is the instance of that blueprint**

3. What are the differences between checked and unchecked exceptions? **Checked exceptions exist at compile time, such as not defining a variable, whereas an unchecked exception happens during run time such as when trying to reference an index that is greater than the size of an array.**

4. What are the differences between abstract classes and interfaces? When should you use one over the other?

***** From reddit**

**In terms of code, they're very similar, which often raises confusion like yours.**

**Conceptually, I think of it this way. When you inherit from an abstract class, you're saying that you *are* a type of the class. You have the same *attributes*.**

**When you implement an interface, you're saying that you *do* the things that that interface does.**

**In one sentence, abstract classes allow you to inherit state/variables/data, interfaces allow you inherit behaviors.**

5. What is unit testing and why is it important?

**Unit testing validates the basic functionality that you have written works as expected.**

6. What is your favorite thing you learned this week?

**I feel like this is the first time that I am truly grasping why OOP is powerful. The videos on the four pillars were exceptional.**