# Chapter Review Questions

Name: Aamir Khan

Date: 7-12-2022

Chapter 10 Review Questions (10 points):

1. What type of class contains at least one abstract method?

**An abstract class.**

1. What type of classes can have objects which are instantiated?

**Concrete classes.**

1. What OOP concept involves the use of a superclass variable to invoke methods on superclass and subclass objects?

**Polymorphism.**

1. Explain how polymorphism enable you to program “in the general” sense rather than “in the specific.”

**Since we can have objects identify as different types or forms, we can treat a subclass object as if it were a superclass object. This allows us to treat the subclass object in a general sense as if it were a superclass object.**

1. Describe the OOP scenarios where having an abstract method would be appropriate.

**If you know you’ll be overriding your methods in your subclasses, then abstract methods would be appropriate.**

1. Explain how polymorphism promotes extensibility in programming/software engineering.

**It promotes extensibility because we can create objects that can take on various forms. We can then treat these objects in different ways, which promotes our ability to handle related objects.**

1. Describe three ways in which you can assign superclass and subclass references to instance variables of superclass and subclass types.
2. **A subclass reference can be stored in a superclass variable.**
3. **A subclass reference can be stored in a subclass variable.**
4. **A superclass reference can be stored in a superclass variable.**
5. Compare and contrast abstract classes and interfaces in Java. When would you use an abstract class or an interface instead of the other?

**Abstract classes and interfaces can’t be instantiated. Interfaces promote multiple inheritance whereas classes can derive from only one abstract class. If you need to have one base class that others will derive from, use an abstract class. If you need to inherit from multiple classes, use interfaces.**

1. Describe what a functional interface is in Java.

**A functional interface is an interface that contains only one abstract method.**

1. In Java 9, the ability of have private methods for interfaces was introduced. Why is it useful to be able to add private methods to interfaces?

**This adds a layer of protection to your methods, and it prevents you from accessing these methods outside of their scope.**