Chapter 3 Review Questions

List the foundational principles of OOP. The foundational principles of OOP are Encapsulation, Data Abstraction, Polymorphism, and Inheritance. Encapsulation is the process of separating clients from implementations by hiding information. Data Abstraction is the ability to define new data types and to manipulate objects holding data-type values. Polymorphism's glossary definition is using the same API (or partial API) for different types of data. The intext definition is when a variable of an interface type invokes a method declared in the interface, Java knows which method to call because it knows the type of the invoking object. Inheritance is that Java provides language support for defining relationships among objects.

What's the difference between class and interface?
Class is the template of the code. It also includes the function and the state. The textbook definition states that class is the Java construct to implement a user-defined data type, providing a template to create and manipulate objects holding values of the type, as specified by an API. The textbook definition of interface is a contract for a class to implement a certain set of methods. A difference is that there needs to be a class for an interface to exist.

What's the difference between regular class and abstract class?
Based off of a good explanation on stackoverflow a regular class is used when you want to enforce certain functions which need to be defined. For example, actions like when we did the Toyota example in class, and we gave it the contract start. An abstract class is used to enforce base functions and have base properties. I believe you would use this if you were trying to add a constant. To clarify it says that an abstract class is given to things that shouldn't change the behavior.