1. What is the concept of an abstract superclass?

An abstract superclass is a superclass that cannot be instantiated and is used to state or define general characteristics. An abstract class serves as a basis for a group of related subclasses

2. What happens when a class statement's top level contains a basic assignment statement?

When a class statement's top level contains a basic assignment statement, the assignment statement is executed first, and then the class statement is executed

3. Why does a class need to manually call a superclass's \_\_init\_\_ method?

To ensuring that objects are properly initialized when there are multiple \_\_init\_\_ methods that need to be called. If the \_\_init\_\_ method of a superclass is not called during object initialization it is likely that that object will end up in an incorrect state.

4. How can you augment, instead of completely replacing, an inherited method?

A more sophisticated way to augment an inherited method involves forwarding. Message forwarding allows you to augment an inherited method in such a way that it can perform its inherited action and some new action.

5. How is the local scope of a class different from that of a function?

Usually, the local scope references the local names of the (textually) current function. Outside functions, the local scope references the same namespace as the global scope: the module's namespace. Class definitions place yet another namespace in the local scope.