1. What is the concept of an abstract superclass?

Abstraction is implemented using the abstract classes. An abstract class in Python is typically created to declare a set of methods that must be created in any child class built on top of this abstract class. Similarly, an abstract method is one that doesn't have any implementation.

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

“Top-level code” is the first user-specified Python module that starts running. It’s “top-level” because it imports all other modules that the program needs. Sometimes “top-level code” is called an entry point to the application.

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

It's because one needs to define something that is NOT done in the base-class' \_\_init\_\_ , and the only possibility to obtain that is to put its execution in a derived-class' \_\_init\_\_ function.

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?

Local scope just refers to the scope available to a given variable, but function scope would refer to variables inside a function