1. What is the concept of an abstract superclass?

An abstract superclass is one way to provide re-usable code. You can extend the abstract class and inherit the code. This is sometimes more convenient than using static methods or object composition to share code.

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

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

The main reason for always calling base class \_init\_\_ is that base class or super class may typically create member variable and initialize them to defaults. So if you don't call base class init, none of that code would be executed and you would end up with base class that has no member variables.

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?

Variables that are declared inside a function are called local variables and in the function scope. Local variables are accessible anywhere inside the function. But Variable that is declared inside a specific block & can't be accessed outside of that block