***1. What is the concept of an abstract superclass?***

***Ans.***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.

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

***Ans.***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.

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

***Ans.***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.

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

***Ans.***Local scope is the area between an { and it's closing }. Function scope is the area between the opening { of a function and its closing }, which may contain more "local" scopes.