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

**Answer:** 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?**

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

**ANSWER:** 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?**

**ANSWER:** Message Forwarding. 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?**

**ANSWER:** A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.