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

**2. What happens when a class statement&#39;s top level contains a basic assignment statement?**

**3. Why does a class need to manually call a superclass&#39;s \_\_init\_\_ method?**

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

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

***SOLUTIONS***

1. An abstract superclass is a class that is designed to be inherited from, and it is not intended to be instantiated directly. It defines common attributes and behaviors that are shared by its subclasses.
2. If a class statement's top level contains a basic assignment statement, it creates a class-level variable that is shared by all instances of the class.
3. A class needs to manually call a superclass's **init** method to ensure that the superclass's initialization code is executed. If the superclass has important initialization logic, it should be invoked by the subclass.
4. To augment an inherited method, you can call the superclass's method using the super() function and then add additional functionality before or after the superclass method is called.
5. The local scope of a class is different from that of a function in that it includes the class's attributes and methods. When a method is called on an instance of a class, the instance is passed as the first argument (self) and is available in the method's local scope. In contrast, a function's local scope only includes the function's arguments and local variables.