

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

1. An abstract superclass is a class that is designed to be inherited from and extended by other classes, but is not meant to be instantiated on its own. It usually contains abstract methods or attributes that the subclass must implement, and provides a basic structure or behavior that the subclass can build upon. Abstract superclasses are commonly used in object-oriented programming to provide a common interface or set of behaviors for a group of related classes.
2. When a class statement's top level contains a basic assignment statement, it creates a class attribute that is shared by all instances of the class. This attribute is not tied to any particular instance, but is part of the class itself. This can be useful for defining default values or shared behavior that applies to all instances of the class.
3. A class needs to manually call a superclass's **`init`** method when it overrides the superclass's **`init`** method with its own implementation. This is because the subclass's **`init`** method will not automatically call the superclass's **`init`** method unless explicitly instructed to do so. By calling the superclass's **`init`** method from the subclass's **`init`** method, the subclass can inherit any behavior or attributes that the superclass provides.
4. To augment an inherited method, you can override the method in the subclass and call the superclass's version of the method using the `super()` function. This allows you to add additional behavior to the method without completely replacing the original behavior. For example, if you inherit a method that performs a calculation, you could override the method in the subclass to perform some additional validation or formatting before calling the superclass's method to perform the calculation.
5. The local scope of a class is different from that of a function in that it includes both class attributes and instance attributes, as well as any methods defined within the class. The local scope of a function, on the other hand, only includes the function's parameters and any variables defined within the function. Additionally, the local scope of a class is accessible from any instance of the class, while the local scope of a function is only accessible within the function itself.