1. What is the concept of an abstract superclass?

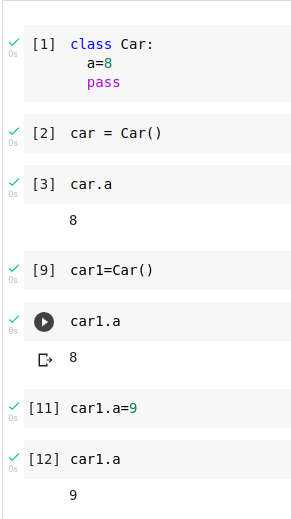
Ans: Abstract superclass is classes where some methods are defined but not implemented. This means that these classes cannot be instantiated, but we can create subclasses that implement those methods and therefore behave like regular classes.

In python we can implement the Abstract class concept by importing ABC module from abc library. We also need “@abstractmethod” and “@property” decoraters to create abstractmethods.

[for more details](<https://medium.com/python-in-plain-english/level-up-your-python-code-with-abstract-classes-7f7f6bdcbb5c>)

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

Ans: If there is a basic assignment statement at the top level of a class statement, then all the objects or instances will get that assigned variable and value by default. We can also reassign the value for any particular instance as follows.

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

Ans: After Inheriting a parent class into a child class, if we define the \_\_init\_\_ of the child class, then it will automatically override the \_\_init\_\_ of the inherited or parrent class.  
  
If we want something from the super class’s \_\_init\_\_ to be done in addition with the child class’s \_\_init\_\_, then we must call it manually.

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

Ans:

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

Ans: