1. What is the concept of an abstract superclass?

Ans🡺 Abstract classes are classes that contain one or more abstract methods. An abstract method is a method that is declared, but contains no implementation. Abstract classes cannot be instantiated, and require subclasses to provide implementations for the abstract methods. The abstract superclass always needs to be implemented its abstract methods in subclasses.

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

Ans🡺 The statements and declarations at top level of class are available throughout the class.

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

Ans🡺 \_\_init\_\_() of the superclass will be called automatically.

super() returns a delegate object to a parent class, so you call the method you want directly on it.

This save us from having to rewrite the same code again, but it also allows us to change the working of constructor for child in another way.

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

Ans🡺 Just use the super keyword inside subclass implementations to get functionality from Super class into subclass

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

Ans🡺 The scope of names defined in a class block is limited to the class block; it does not extend to the code blocks of methods.

Also the local variables of function are limited to function only and won’t be available to class.