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

A common superclass for several subclasses. Factor up common behavior Define the methods they all respond to. Methods that subclasses should implement are declared abstract Instances of the subclasses are created, but no instances of the superclass

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

When class statement's top level contains a basic assignment statement, it is considered as class attribute. Change in the value of class attribute will affect all the instances of the class.

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

A class need to manually call a superclass's init method doing so that we can access those methods of the super-class (parent class) which have been overridden in a sub-class (child class) that inherits from it.

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

The way to do that in Python is by calling to the original version directly, with augmented arguments.

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

In class, if the variable is declared without self then it is accessible within that function only, kinda local variable. However if it declared using self like self.variable\_name = 'somevalue', then it is accessible via any object but not via the class name.

Whereas, if a variable is declared within a function then it is a local variable and is accessible to that function only.