1. What is the concept of an abstract superclass?

ANS:

An abstract class, in the context of Java, is a superclass that cannot be instantiated and is used to state or define general characteristics.

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

ANS:

Python’s assignment operators allow you to define **assignment statements**. This type of statement lets you create, initialize, and update variables throughout your code. Variables are a fundamental cornerstone in every piece of code, and assignment statements give you complete control over variable creation and mutation.

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

ANS:

The \_\_init\_\_ function is used to set everything up when you create an instance of the class, so users of class B will assume that this has been done and if not, then it may produce bugs or side effects the user does not understand.

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

ANS:

A more sophisticated way to augment an inherited method involves forwarding. Message forwarding allows you to augment an inherited method in such a way that it can perform its inherited action and some new action.

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

ANS:

## Local Scope

A variable created inside a function belongs to the local scope of that function, and can only be used inside that function.

### Example:-

A variable created inside a function is available inside that function:

def myfunc():  
  x = 300  
  print(x)  
  
myfunc()