1. What is the concept of an abstract superclass?

**Ans**. An abstract superclass is one way to provide re-usable code. You can extend the abstract class and inherit the code. This is sometimes more convenient than using static methods or object composition to share code.

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

**Ans.** When a Class statement's top level contains a basic assignment statement, its usually treated as a class attribute or class level variable. where as assignment statements inside methods are treated as instance attributes or local attributes. When an instance of a class is created a single copy of class attributes is maintained and shared to all instances of class. where as each instance object maintains its own copy of instance variables.

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

**Ans.**superclass is needed to create the member and initialize them to defaults.

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

**Ans**.With the help of super method().

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

**Ans.** A Variable which is defined inside a function is local to that function. it is accesible from the point at which it is defined until the end of the function.

A variable inside of a class also has a local variable scope. Variables which are defined in the class body (but outside all methods) are called as class level variables or class attributes. they can be referenced by there bare names within the same scope, but they can also be accessed from outside this scope if we use the attribute access operator (.). on a class or an instance of the class.