1. What is the concept of an abstract superclass?

Ans. An abstract class can be considered as a blueprint for other classes. It allows you to

create a set of methods that must be created within any child classes built from the

abstract class. A class which contains one or more abstract methods is called an abstract

class. An abstract method is a method that has a declaration but does not have an

implementation.

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

Ans. It is act a simple variable having some value stored in it which can be used as a normal variable while calling any function or variables inside the class. It is global in nature for the class. It can be used directly by with class or with class object or with any other subclass.

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

Ans. The main reason for always calling base class \_init\_\_ is that base class may typically create member variable and initialize them to defaults. So if you don't call base class init, none of that code would be executed and you would end up with base class that has no

member variables.

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. In case of class, local scope means, whatever the variable is declared under the class or inside the class that will be used within a class only either by binding it with object of a class or with class itself. Unlike the function, local scope of a variable is not restricted with in one function only. In class it can be used with any number of function which are defined inside the class.