1. What is the concept of an abstract superclass?

Abstract superclass is a class that is declared but does not have implementation. So import the abc module and declare an abstract method by using @abstractmethod.

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

An assignment statement evaluates the expression list (remember that this can be a single expression or a comma-separated list, the latter yielding a tuple) and assigns the single resulting object to each of the target lists, from left to right.

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

The reason we use super is so that child classes that may be using cooperative multiple inheritance will call the correct next parent class function in the Method Resolution Order (MRO)

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

Overriding is an important part of OOP since it makes inheritance utilize its full power. By using the method overriding a class may "copy" another class, avoiding duplicated code, and at the same time enhance or customize part of it. Method overriding is thus a part of the inheritance mechanism.

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

Scope of a local class is limited only within the function definition. The function can use the class as usual as local variables. The class gets destroyed as soon as the function is returned.