1. What is the concept of an abstract superclass?

**Ans:** 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:** An assignment statement sets and/or re-sets the value stored in the storage location(s) denoted by a variable name; in other words, it copies a value into the variable. In most imperative programming languages, the assignment statement (or expression) is a fundamental construct.

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:** In Python method overriding occurs by simply defining in the child class a method with the same name of a method in the parent class. When you define a method in the object you make this latter able to satisfy that method call, so the implementations of its ancestors do not come in play.

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

**Ans:** A global variable is a variable that is accessible globally. A local variable is one that is only accessible to the current scope, such as temporary variables used in a single function definition.