Q1. What is the meaning of multiple inheritance?

Multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit features from more than one parent object or parent class. It is distinct from single inheritance, where an object or class may only inherit from one particular object or class.

Q2. What is the concept of delegation?

1. Delegation can be an alternative to inheritance.
2. Delegation means that you use an object of another class as an instance variable, and forward messages to the instance.
3. It is better than inheritance for many cases because it makes you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn’t force you to accept all the methods of the superclass: you can provide only the methods that really make sense.
4. Delegation can be viewed as a relationship between objects where one object forwards certain method calls to another object, called its delegate.

Q3. What is the concept of composition?

The main reason to use composition is that it allows you to reuse code without modeling an is-a association as you do by using inheritance.

The concept of composition is often used in the real world, and it should be the same in software development. A car is not an engine; it has one. And a coffee machine has a grinder and a brewing unit, but it is neither of them. The car and the coffee machine integrate an engine, grinder and brewing unit via their external APIs to compose a higher level of abstraction and provide more significant value to their users.

Q4. What are bound methods and how do we use them?

A bound method is the one which is dependent on the instance of the class as the first argument. It passes the instance as the first argument which is used to access the variables and functions.

Q5. What is the purpose of pseudo private attributes?

The problem that the pseudo-private attribute feature is meant to alleviate has to do with the way instance attributes are stored. In Python, all instance attributes wind up in the single instance object at the bottom of the class tree. This is very different from the C++ model, where each class gets its own space for data members it defines.

Within a class method in Python, whenever a method assigns to a self attribute (e.g., self.attr=value), it changes or creates an attribute in the instance (inheritance search only happens on reference, not assignment). Because this is true even if multiple classes in a hierarchy are assigned to the same attribute, collisions are possible.