Q1. What is the meaning of multiple inheritance?

Multiple inheritance is a feature in object-oriented programming languages where a class can inherit characteristics and behaviors from more than one parent class. In other words, a class can have multiple direct base classes.

Q2. What is the concept of delegation?

The concept of delegation in programming refers to the act of one object or class assigning responsibility for a particular task or behavior to another object. Instead of directly implementing the behavior itself, the delegating object forwards the task to a delegate object and relies on it to perform the desired functionality.

Q3. What is the concept of composition?

The concept of composition in programming refers to the practice of combining or assembling multiple objects to create more complex or specialized objects. It involves creating complex structures by incorporating simpler objects as components or parts.Composition allows for building complex systems or objects by combining simpler and more specialized components. Instead of inheriting behavior directly from a base class (as in inheritance), composition involves creating relationships between objects based on their roles and responsibilities.

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

In object-oriented programming, a bound method is a method that is associated with an instance of a class. When a method is called on an instance, it is automatically passed the instance itself as the first argument (typically named self), allowing the method to access and operate on the instance's data.Bound methods are created when a class is instantiated, and they maintain a reference to the instance on which they are called. This reference allows the method to access the instance's attributes and perform operations specific to that instance.

Q5. What is the purpose of pseudoprivate attributes?

Pseudoprivate attributes, often denoted by a double underscore prefix (e.g., \_\_attribute), are a convention in Python for name mangling, intended to provide a form of name privacy or "weak" encapsulation within a class. The purpose of pseudoprivate attributes is to make the attribute less likely to be accidentally accessed or overridden by subclasses or external code.