Q1. What is the meaning of multiple inheritance?

Multiple inheritance refers to a feature in object-oriented programming where a class inherits attributes and behaviors from more than one parent class. This allows a subclass to inherit from multiple superclasses, potentially leading to complex inheritance hierarchies.

Q2. What is the concept of delegation?

Delegation is a design principle where an object forwards or delegates some of its responsibilities to another object. Instead of inheriting behavior, an object maintains a reference to another object and invokes its methods to achieve certain functionalities. It promotes code reusability and modularity.

Q3. What is the concept of composition?

Composition is a design concept where complex objects are built by combining simpler objects or components. It emphasizes creating relationships between objects rather than inheriting behavior. Composition allows for more flexibility and reduces tight coupling between classes.

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

Bound methods are methods of a class that are associated with an instance of that class. When a bound method is called, the instance is automatically passed as the first argument (usually referred to as self). This enables the method to operate on the specific instance's data and behavior.

Q5. What is the purpose of pseudoprivate attributes?

Pseudoprivate attributes (often referred to as name mangling) in Python are attributes with names that begin with double underscores (\_\_). They are used to make attributes less likely to collide with attributes defined in subclasses or other classes. While not truly private, they provide a level of name separation and help avoid accidental name clashes in subclasses.