Q1. What is the meaning of multiple inheritance?

Q2. What is the concept of delegation?

Q3. What is the concept of composition?

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

Q5. What is the purpose of pseudoprivate attributes?

A1. Multiple inheritance is a feature of object-oriented programming where a class can inherit properties and methods from multiple parent classes. This allows for greater flexibility in designing classes and can help to avoid code duplication.

A2. Delegation is a design pattern in which an object passes on a request to another object that is better equipped to handle it. This allows for greater flexibility in designing classes and can help to avoid code duplication.

A3. Composition is a design pattern in which a class is composed of one or more instances of other classes, rather than inheriting from them. This allows for greater flexibility in designing classes and can help to avoid the pitfalls of multiple inheritance.

A4. Bound methods are methods that have been bound to an instance of a class, so that they can access and modify the attributes of that instance. We use bound methods by calling them on an instance of a class, rather than on the class itself.

A5. The purpose of pseudoprivate attributes is to provide a mechanism for data hiding within a class. By prefixing an attribute name with two underscores (e.g., `__attribute`), we indicate to Python that the attribute should not be accessed directly from outside the class. Instead, we can provide accessor methods (e.g., `get_attribute()` and `set_attribute()`) that allow controlled access to the attribute.