**Python advance assignment -5**

**Q1. What is the meaning of multiple inheritance?**

Multiple inheritance refers to the ability of a class to inherit properties and methods from more than one parent class. This allows a subclass to inherit behavior from multiple superclasses and combine them in a single class.

**Q2. What is the concept of delegation?**

Delegation is a design pattern in which an object (the delegate) is given the responsibility of performing a certain action on behalf of another object (the delegator). The delegator forwards the request to the delegate, which performs the action and returns the result. This allows the delegator to remain unaware of the details of the action being performed.

**Q3. What is the concept of composition?**

Composition is a design pattern in which an object (the composite) is made up of other objects (the components). The composite has an "has-a" relationship with the components, meaning it has a reference to one or more of them and can use their properties and methods. This allows the composite to reuse the behavior of the components and build more complex objects.

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

Bound methods are methods that have been bound to a specific instance of a class, meaning they have access to the instance's attributes and methods. They are created when an instance method is accessed through an instance of the class, and they can be used to call the method on that specific instance. They are typically used when passing methods as arguments to other functions.

**Q5. What is the purpose of pseudoprivate attributes?**

Pseudoprivate attributes are attributes that are intended to be used only within the class, but not directly accessed or modified by external code. They are typically prefixed with a single underscore (\_) to indicate that they are intended for internal use only. This is a convention, they can be accessed and modified from the outside but it is not recommended. It serves as a indication to the developer to use them with cautio