ASSIGNMENT - 3

1. What is the concept of an abstract superclass?

Ans: An abstract superclass in Python is a class that is not meant to be instantiated directly. It often contains abstract methods that must be implemented by its subclasses. It serves as a blueprint for its subclasses, defining a common interface that its subclasses should adhere to. Python's abc module provides tools for creating abstract classes and abstract methods using the ABC (Abstract Base Class) and abstractmethod decorators.

2. What happens when a class statement’s top level contains a basic assignment statement?

Ans: When a basic assignment statement, like x = 10, is present at the top level of a class statement, it creates a class-level variable, accessible to all instances of that class. These variables are shared among instances unless they are modified on an instance.

3. Why does a class need to manually call a superclass’s \_\_init\_\_ method?

Ans: In Python, when a subclass defines its \_\_init\_\_ method, it does not automatically call the superclass's \_\_init\_\_ method. To ensure that the superclass's initialization logic is executed, the subclass needs to explicitly call the superclass's \_\_init\_\_ method using super().\_\_init\_\_(). This allows the subclass to inherit and initialize attributes from the superclass.

4. How can you augment, instead of completely replacing, an inherited method?

Ans: To augment rather than completely replace an inherited method, you can achieve this by using method overriding. Inside the overriding method in the subclass, call the superclass method using super() and then add or modify functionality before or after the call to the superclass method. This way, you retain the original behavior of the inherited method while extending or modifying its functionality.

5. How is the local scope of a class different from that of a function?

Ans: Class scope: Variables defined within a class (outside of any method) are class attributes and are accessible throughout the class. These class attributes are shared among all instances of the class.

Function scope: Variables defined within a function have a local scope and are accessible only within that function. They are not accessible outside the function unless explicitly returned or stored elsewhere.