**ADITYA AMIN**

**ASSIGN : 03**

1. What is the concept of an abstract superclass?

The concept of an abstract superclass is related to abstract classes and is a fundamental concept in object-oriented programming (OOP). An abstract superclass, also known as an abstract base class, is a class that is designed to be inherited by other classes, but it cannot be instantiated on its own.

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

When a class statement's top level contains a basic assignment statement, it defines a class attribute. A class attribute is a variable or value that is shared among all instances of the class. It is associated with the class itself rather than with individual instances.

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

In Python, a class needs to manually call a superclass's \_\_init\_\_ method when it wants to initialize the attributes and behavior inherited from the superclass in addition to its own attributes and behavior.

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

To augment, or extend, an inherited method instead of completely replacing it, you can follow these steps in Python:

Define the subclass that inherits from the superclass.

Declare a method in the subclass with the same name as the method you want to augment from the superclass.

Inside the subclass method, call the superclass method using the super() function.

Add additional functionality before or after calling the superclass method.

Optionally, return or modify the result of the superclass method if needed.

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

The local scope of a class and the local scope of a function differ in several ways:

Accessible Variables: In a class, variables defined within methods are accessible throughout the class. These variables are referred to as instance variables or attributes and can be accessed using the self keyword. In contrast, in a function, variables defined within the function are only accessible within that function's scope and cannot be accessed from outside the function.