1. What is the concept of an abstract superclass?

Ans: - An abstract superclass in Python is a class that cannot be instantiated and is meant to be subclassed by other classes1. It allows you to define a set of methods that must be created within any child classes built from the abstract class1. A class that contains one or more abstract methods is called an abstract class.



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

Ans: - When a class statement’s top level contains a basic assignment statement, it creates a class attribute. This attribute is shared by all instances of the class.

A screenshot of a computer screen

Description automatically generated

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

Ans: - A class needs to manually call a superclass’s \_\_init\_\_ method to ensure that the initialization code in the superclass gets executed3. This is important because the superclass’s \_\_init\_\_ method might set up variables or state that is necessary for the methods in the superclass.

A screenshot of a computer program

Description automatically generated

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

Ans: - You can augment an inherited method by calling the method from the superclass within the overridden method in the subclass. This allows you to add additional functionality in the subclass’s method without losing the behavior defined in the superclass’s method.

A screen shot of a computer program

Description automatically generated

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

Ans: - In Python, the local scope of a function includes all variables defined within the function. These variables are only accessible within the function. On the other hand, the local scope of a class includes all variables and methods defined within the class. These can be accessed by all methods within the class and by instances of the class.