1. What is the concept of an abstract superclass?

Ans: **A class is called an abstract class when one of its methods is abstract. Abstract class just behaves like a template for the subclass. There are two important aspects about the abstract class.**

1. **An instance of the abstract class can’t be created.**
2. **If the abstract class is inherited into any sub class then the abstract methods have to be defined in the sub class.**

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

Ans: **A basic assignment is written in the constructor where we use a pointer to assign the passing value to the class while creating an instance which can be used in other methods in the same class.**

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

Ans:

**In the multiclass inheritance, we can import the parent class init method to be able to assign values to the attributes defined in these classes. For example, in the below screenshot, we have to call the init methods of the both classes so that we can retrieve the attributes of both classes when we create an instance of the child class.**



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

Ans: **Mixin approach and multiple inheritance can be used to augment an inherited method.**

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

Ans: **if a variable is defined inside a class but outside of any method/constructor then it can be used with the class\_name.variable\_name but a variable defined inside a method can be used by method\_name.variable\_name inside the same class.**

