Q1. Which two operator overloading methods can you use in your classes to support iteration?

Ans

the \_\_iter\_\_ and \_\_getitem\_\_ methods can be used to make your class iterable and support iteration. The choice between them depends on the specific requirements and design of your class.

The \_\_iter\_\_ method provides more control and flexibility over the iteration process, while the \_\_getitem\_\_ method allows for indexing and is suitable when the class represents a collection-like object

Q2. In what contexts do the two operator overloading methods manage printing?

Ans

the \_\_str\_\_ and \_\_repr\_\_ methods manage printing by providing different representations of an object. The \_\_str\_\_ method focuses on providing a human-readable representation, while the \_\_repr\_\_ method focuses on providing an unambiguous representation for debugging and internal purposes.

Q3. In a class, how do you intercept slice operations?

By implementing the \_\_getitem\_\_ method and handling slice objects appropriately, you can intercept and customize slice operations in your class to control how data is retrieved from your objects.

Q4. In a class, how do you capture in-place addition?

Ans the \_\_iadd\_\_ method. The \_\_iadd\_\_ method allows objects to support the += operator, which performs in-place addition.

Q5. When is it appropriate to use operator overloading?