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

\_\_iter\_\_ and \_\_next\_\_ are two operator overloading methods can be used in classes to support iteration.

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

\_\_str\_\_ and \_\_repr\_\_ are two methods used to manage the printing context. In \_\_str\_\_ we should return a string in a human readable format. It used to print the output of the class in a string format. \_\_repr\_\_ is used for debugging and development it should return a string that can be used to recreate the object i.e. it should represent an object.

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

To intercept slice operations in a class, you can use the \_\_getitem\_\_() method along with the slice object. The \_\_getitem\_\_() method allows you to customize the behavior of getting an item from an object using indexing or slicing.

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

We use \_\_iadd\_\_ method to modify in-place addition. It can be represented as += operator.

**Q5. When is it appropriate to use operator overloading?**

Operator overloading is appropriate to use in situations where it enhances the readability, expressiveness, and intuitiveness of your code by allowing objects to behave like built-in types or by defining custom behavior for specific operators.