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

**Answer 1:** To support iteration in your classes, you can use the \_\_iter\_\_ and \_\_next\_\_ methods for operator overloading.

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

**Answer 2:** The \_\_str\_\_ and \_\_repr\_\_ methods in operator overloading are used for managing printing in different contexts.

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

**Answer 3:** To intercept slice operations in a class, you can implement the \_\_getitem\_\_ method.

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

**Answer 4:** To capture in-place addition in a class, you can use the \_\_iadd\_\_ method for operator overloading.

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

**Answer 5:** Operator overloading is appropriate when you want to define custom behavior for standard operators in your class, providing a more intuitive and expressive interface for instances of your class.