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

The \_\_iter\_\_() method is used to define the start of the iteration, and the \_\_next\_\_() method is used to iterate through the elements of the collection.

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

The two operator overloading methods that handle printing are \_\_str\_\_ and \_\_repr\_\_.

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

**slice** is a constructor in Python that creates slice object to represent set of indices that the range(start, stop, step) specifies.

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

Python provides the operator x += y to add two objects in-place by calculating the sum x + y and assigning the result to the first operands variable name x

Q5. When is it appropriate to use operator overloading?

It allows us to provide an intuitive interface to our class users, plus makes it possible for templates to work equally well with classes and built-in types