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

The \_\_iter\_\_ returns the iterator object and is implicitly called at the start of loops. The \_\_next\_\_ method returns the next value and is implicitly called at each loop increment.

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

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

The slice operator [n:m] returns the part of the string from the n'th character to the m'th character, including the first but excluding the last. In other words, start with the character at index n and go up to but do not include the character at index m.

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 . You can set up the in-place addition behavior for your own class by overriding the magic “dunder” method \_\_iadd\_\_(self, other) in your class definition.

Q5. When is it appropriate to use operator overloading?

, if the user wants to make the operator “+” to add two class objects, the user has to redefine the meaning of the “+” operator such that it adds two class objects. This is done by using the concept “Operator overloading”.