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. \_\_next\_\_ raises a StopIteration exception when there are no more value to return, which is implicitly captured by looping constructs to stop iterating.

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

\_\_str\_\_() method used to do the two operator overloading methods manage the printing

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

Using the method of \_\_setslice\_\_() ,\_\_getslice\_\_(),\_\_delslice\_\_() to intercept the slice operation in class

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

assignment and computation in a single statement using “operator” module

Example:

x += y is equivalent to x = operator.iadd(x, y)

Some operations in- place addition is:

iadd() :- This function is used to assign and add the current value.

iconcat() :- This function is used to concat one string at end of second.

isub() :- This function is used to assign and subtract the current value.

imul() :- This function is used to assign and multiply the current value.

itruediv() :- This function is used to assign and divide the current value.

imod() :- This function is used to assign and return remainder .

The in-place assinging is not performed on immutable containers like string, numbers,tuple

Q5. When is it appropriate to use operator overloading?

Operator overloading means adding extra functionality for a certain operator. When an operator is overloaded, the operator has different meanings, which depends on the type of its operands