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

\_\_lt\_\_() or \_\_gt\_\_() we can use for the iterations support.

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

It will print in case when we add the print statement while overloading a method or by putting the return statement and returning a statement we can manage the printing.

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

With the help of \_\_getitem\_\_() we can perform the slicing operations. By calling a slice() constructor.

sliced ='abcde'.\_\_getitem\_\_(slice(0, 2, 1))

print(sliced)

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

With the help of iadd () method we can do that as below:

Import operator

X= operator.iadd(2,3)

Or

X +=1

Q5. When is it appropriate to use operator overloading?

Its appropriate when we need to give some comments or need to add some additional functionality into the existing operator behavior.