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

Function overloading and Operator overloading

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

Overriding built in functions and overriding user defined function.

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

The **\_\_getitem\_\_** method is used for accessing list items, array elements, dictionary entries etc. **slice** is a constructor in Python that creates slice object to represent set of indices that the range(start, stop, step) specifies. \_\_getitem\_\_ method can be implement in a class, and the behavior of slicing can be defined inside it.

\_\_getitem\_\_(slice(start,stop,step))

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

Using iadd()) to add and assign value.This operation does a+=b is equivalent to a=operator.iadd(a,b)

Q5. When is it appropriate to use operator overloading?

Operator overloading is mostly useful when you're making a new class that falls into an existing "Abstract Base Class" (ABC) -- indeed, many of the ABCs in standard library module [collections](http://docs.python.org/library/collections.html#module-collections) rely on the presence of certain special methods (and special methods, one with names starting and ending with double underscores AKA "dunders", are exactly the way you perform operator overloading in Python.