Q1. Which two operator overloading methods can you use in your classes to support iteration?
Answer: __iter___ and __next___ are the operator overloading methods in python that support iteration. They are collectively called iterator protocol.
__iter__ returns the iterator object and is called at start of loop in our respective class.
__next__ is called at each loop increment returns the incremented value. Also StopIteration is raised when there is no value to return.

Q2. In what contexts do the two operator overloading methods manage printing?
Answer: __str__ and __repr__ are two operator overloading methods that manage printing.
Difference between two methods is , __str__ prints the informal string representation of an object, one that is useful for printing the object.
__repr__ is used to print official string representation of an object , so it includes all information for debugging and development.
E.g.
str(2.0/11.0) = 0.181818181818
repr(2.0/11.0) = 0.18181818181818182

Q3. In a class, how do you intercept slice operations?
Answer: Use of "slice" in __getitem__ method is used for intercept slice operation. This slice constructor is provided with start integer number, stop integer number and step interger number.
__getitem__(slice(start, stop, step))

Q4. In a class, how do you capture in-place addition?
Answer : "a+b" is normal addition operation, while "a+=b" is in-place addition operation. In this in-place addition a itself will store value of the addition.
__iadd__ method is used for this in-place operation.

Q5. When is it appropriate to use operator overloading?
Answer: Operator Overloading is used when we want to use an operator other than its normal operation to have different meaning according to the context required in user defined function.