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

- add() and **mul**() methods can be used for both integers and string data objects. String is iterable object. When two strings are passed to + operator, it will return concatenated string. When * (number) is preceeded by string, then that string is repeated those many number of time.

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

-It depends on the input parameter result is printed. Example, If both inputs are string function will print string output.

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

-The slice () function returns a slice object. A slice object is used to specify how to slice a sequence. You can specify where to start the slicing, and where to end. You can also specify the step, which allows you to e.g., slice only every other item.

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

-The In-place operator functions perform computation & assignment in a single statement. For example, the standard operator functions like add (), mul () etc take two parameters, perform the operation of them & return the resultant. They do not modify the parameters or arguments. But this is slightly different in the case of in place operator functions.

Q5. When is it appropriate to use operator overloading?

-When we have two objects which are a physical representation of a class (user-defined data type) and we have to add two objects with binary '+' operator it throws an error, because compiler don't know how to add two objects. So we define a method for an operator and that process is called operator overloading.