**Python advance assignment-4**

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

The two operator overloading methods that can be used in classes to support iteration are the \_\_iter\_\_ and \_\_next\_\_ methods. The \_\_iter\_\_ method returns an iterator object and the \_\_next\_\_ method returns the next item from the iterator.

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

The \_\_str\_\_ method manages printing when the built-in print() function or the str() function is called. The \_\_repr\_\_ method manages printing when an object is represented in the interactive interpreter or in a repr() function call.

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

In a class, you can intercept slice operations by defining a \_\_getitem\_\_ method and handling the slice operations within that method.

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

In a class, you can capture in-place addition by defining the \_\_iadd\_\_ method and handling the in-place addition operation within that method.

**Q5. When is it appropriate to use operator overloading?**

It is appropriate to use operator overloading when you want to define custom behavior for built-in operators such as +, -, \*, /, etc. for objects of your class. This allows your objects to behave in a similar way to built-in types and make your code more readable.