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

\_\_iter\_\_ and \_\_next\_\_ operator overloading methods can be used to support iterations

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

Printing the result of the overloaded iteration methods yields the result of the new set of instructions given by the user

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

The \_\_getitem\_\_, \_\_setitem\_\_ and \_\_delitem\_\_ can be used to intercept slicing operations

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

Overloading the \_\_add\_\_ operator

Q5. When is it appropriate to use operator overloading?

Operator overloading can be used when we want the operators to do operations beyond their usual programming. In cases where a user might create their own specific datatypes, operator overloading can be used to tell the program how operators will behave with respect to the new datatype