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

\_\_iter\_\_ and \_\_next\_\_

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

\_\_str\_\_() manage printing using format

def \_\_str\_\_(self):

return "({0}, {1})".format(self.x, self.y)

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

Using “setitem” and “delitem” we can intercept slice operations.

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

Below are some operations are used for in-place addition

Isadd(), isub(), imul(), itruediv(),imod()

Q5. When is it appropriate to use operator overloading?

It is mostly useful when we are making a new class that falls into an existing "Abstract Base Class" (ABC)