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

1. \_\_iter\_\_
2. \_\_next\_\_

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

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

Using a.\_\_getitem\_\_(slice(i, j)) or \_\_setitem\_\_() or \_\_delitem\_\_()

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

## iadd() function. This function is used to assign the current value and add them. This operator does x+=y operation. In the case of strings, numbers assigned are not performed.

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

When you have to add, subtract, multiply values of multiple objects.

**Example:**

| Class Operations :  def \_\_init\_\_(self, a, b):  self.a = a  self.b = b   def \_\_add\_\_(self, other):  return self.a + other.a, self.b + other.b  def \_\_sub\_\_(self, other):  return self.a - other.a, self.b - other.b   Ob1 = Operations(1, 2) Ob2 = Operations(2, 3) Ob3 = Ob1 + Ob2  Ob4 = Ob1 - Ob2 print(Ob3)  print(Ob4) |
| --- |