**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?**

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

The \_\_getitem\_\_ method is used for accessing list items, array elements, dictionary entries etc. slice is a constructor in Python that creates slice object to represent set of indices that the range(start, stop, step) specifies. \_\_getitem\_\_ method can be implement in a class, and the behavior of slicing can be defined inside it.

class Demo:

def \_\_getitem\_\_(self, key):

# print a[1], a[1, 2],

# a[1, 2, 3]

print(key)

return key

a = Demo()

# => slice 1

a[1]

# => slice(1, 2)

a[1, 2]

# => (1, 2, 3)

a[1, 2, 3]

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

Python in its definition provides methods to perform inplace operations, i.e doing assignment and computation in a single statement using “operator” module. For example,

**x += y is equivalent to x = operator.iadd(x, y)**

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

When multiple task can be executed by same operator