Week 1 - In your Interface

**1) Add the \_\_contains\_\_ protocol and show whether or not  'Tim' and 'Sam' are part of our team.**

**2) Add the \_\_iter\_\_ protocol and show how you can print each member of the classmates object.**

Answers of questions 1 and 2:

class Teams:

def \_\_init\_\_(self, members):

self.\_\_myTeam = members

def \_\_len\_\_(self):

return len(self.\_\_myTeam)

# Question 1

def \_\_contains\_\_(self, item):

return item in self.\_\_myTeam

# Question 2

def \_\_iter\_\_(self):

yield from self.\_\_myTeam

def main():

classmates = Teams(['John', 'Steve', 'Tim'])

print(len(classmates))

# Question 1

print()

print('Tim' in classmates)

print('Sam' in classmates)

# Question 2

print()

for classmate in classmates:

print(classmate)

main()

**3) Determine if the class classmates implements the \_\_len\_\_ method.**

Teams is the class while classmates is an object of class Teams. The Teams class implements the \_\_len\_\_ function.

**4) Explain the difference between interfaces and implementation.**

An interface sets what functions need to be implemented in the class. An implementation refers to the actual function definition implemented into the class. An interface is the cookie cutter, while the implementation is the cookie.

**5) Using both visual and written descriptions, think through the interface-implementation of a large-scale storage system.   In many systems today, we have the ability to store information from a single application to a variety of storage devices - local storage (hard drive, usb), the cloud and/or some new medium in the future.   How would you design an interface structure such that all of the possible implementations could store data effectively.**

We can create an interface for all the storage devices that way there is a standardized set of functions. Then we can implement each of the functions differently depending on what the storage device is. This can help ensure that each of the devices are uniform in the structure, but unique in their implementation design.

Submit Assignment