Section 10

Classes & Hw6

Logistics

HW5 Part 2 due on Wednesday (11/30/2022)

Classes

What is a class?

a class encapsulates and abstracts data and logic that you can use in your programs

We want to make a program that can keep track of phone numbers.

A class contains...

Variables to hold the necessary data

A phone number should probably store:

- Area code
- Exchange
- Number

```
EX:  (415)  552-7909
       ^      ^    ^
       |      |    |
       |      |     number
       |     exchange
      area  code
```

Creating an instance of a class

When we want to start using a class, we typically want to set some of its variables to their correct values

This is done through a special method called __init__

When we create a phone number, we want to specify it’s area code, exchange, and number.

```python
def __init__(self, area_code, exchange, number):
    """
    Creates a new PhoneNumber from the provided area code,exchange and number.
    """
    self.area_code = area_code
    self.exchange = exchange
    self.number = number
```

```python
# Make some new phone numbers
num1 = PhoneNumber(916, 272, 8010)
num2 = PhoneNumber(916, 274, 2805)
num3 = PhoneNumber(415, 552, 7909)
```

A class contains...

Methods to create, query, and modify an instance of a class

(Methods are basically functions inside classes)

A phone number should probably be able to:

- Be created
- Be called
- Print its number

```
def call(self):

    """
    Calls this PhoneNumber.
    """
    print("Calling (" + str(self.area_code) + ") " + str(self.exchange) + "-" + str(self.number))
    print("ring... ring... Hello?")

def print_number(self):
    """
    Prints a pretty version of this PhoneNumber.
    """
    print("(" + str(self.area_code) + ") " + str(self.exchange) \
        + "-" + str(self.number))
```

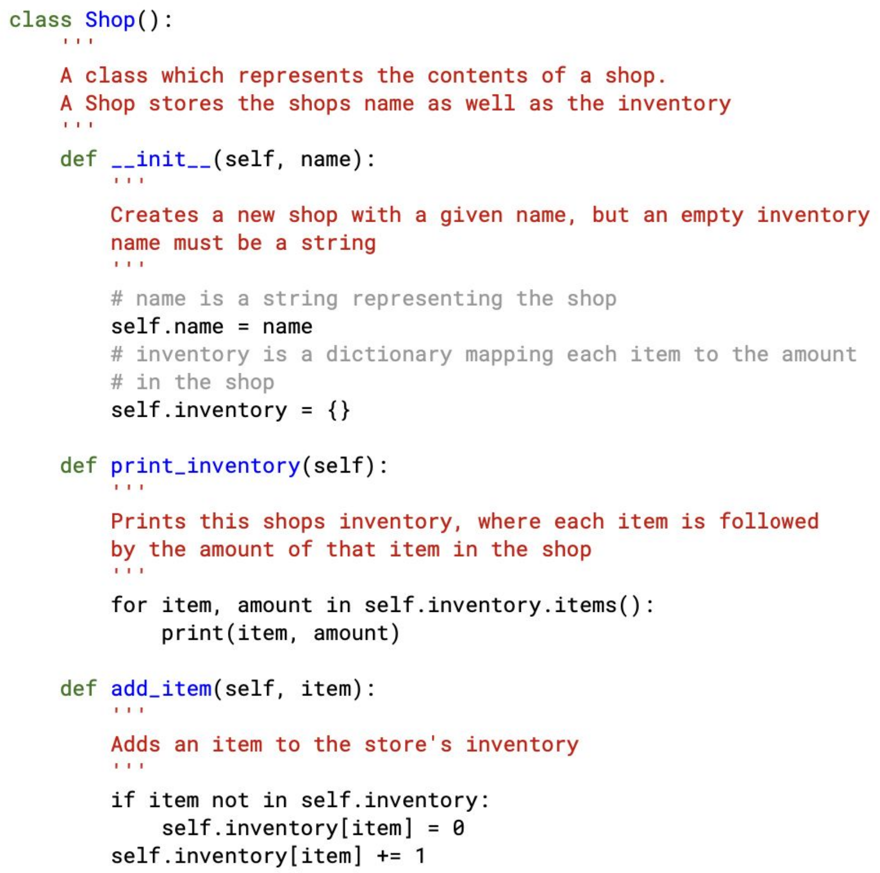
Section Problems

Link

Problem 1a

What would the following client code print?

```
flower_shop = Shop("Swansons")
flower_shop.add_item("Daisy")
flower_shop.print_inventory()
```



```
class Shop():
    '''
    A class which represents the contents of a shop.
    A Shop stores the shops name as well as the inventory
    '''
    def __init__(self, name):
        '''
        Creates a new shop with a given name, but an empty inventory
        name must be a string
        '''
        # name is a string representing the shop
        self.name = name
        # inventory is a dictionary mapping each item to the amount
        # in the shop
        self.inventory = {}

    def print_inventory(self):
        '''
        Prints this shops inventory, where each item is followed
        by the amount of that item in the shop
        '''
        for item, amount in self.inventory.items():
            print(item, amount)

    def add_item(self, item):
        '''
        Adds an item to the store's inventory
        '''
        if item not in self.inventory:
            self.inventory[item] = 0
        self.inventory[item] += 1
```

Problem 1b

What code would you write to make a new shop called Arcane Comics that has 5 comics in it, and then print out it's inventory?

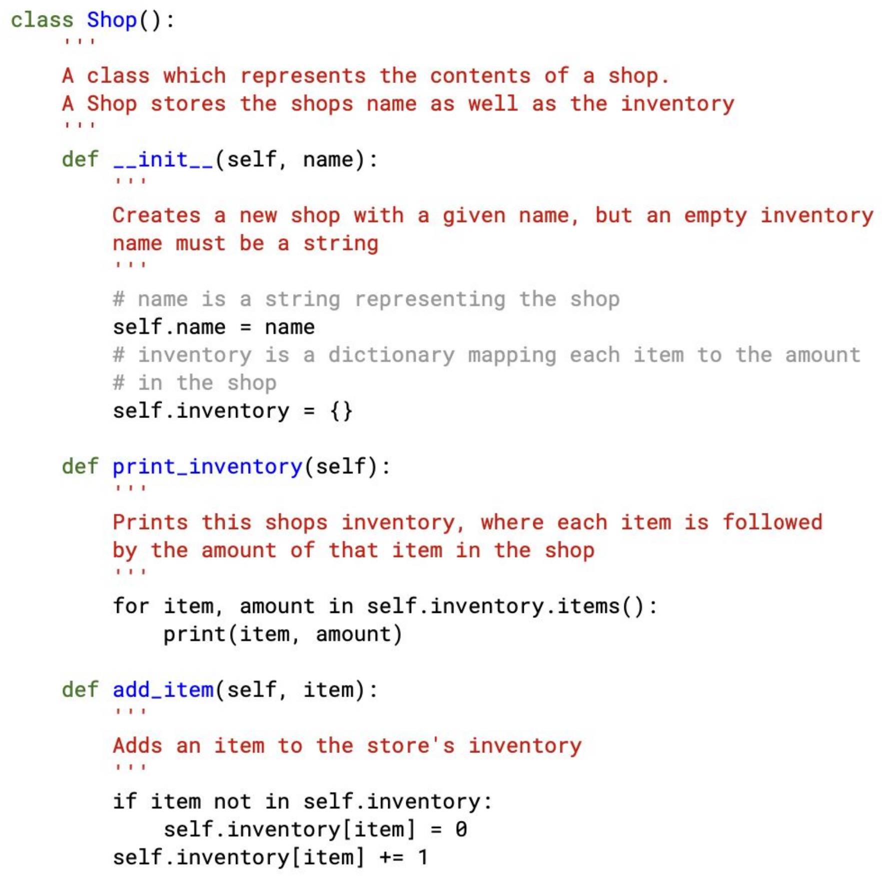
```python
class Shop():
    '''
    A class which represents the contents of a shop.
    A Shop stores the shops name as well as the inventory
    '''
    def __init__(self, name):
        '''
        Creates a new shop with a given name, but an empty inventory
        name must be a string
        '''
        # name is a string representing the shop
        self.name = name
        # inventory is a dictionary mapping each item to the amount
        # in the shop
        self.inventory = {}

    def print_inventory(self):
        '''
        Prints this shops inventory, where each item is followed
        by the amount of that item in the shop
        '''
        for item, amount in self.inventory.items():
            print(item, amount)

    def add_item(self, item):
        '''
        Adds an item to the store's inventory
        '''
        if item not in self.inventory:
            self.inventory[item] = 0
        self.inventory[item] += 1
```

Problem 1c

Adding one item at a time is extremely tedious. Write a function for the class Shop called add_items that given an item and an amount, adds that amount to the store's inventory.



```python
class Shop():
    '''
    A class which represents the contents of a shop.
    A Shop stores the shops name as well as the inventory
    '''
    def __init__(self, name):
        '''
        Creates a new shop with a given name, but an empty inventory
        name must be a string
        '''
        # name is a string representing the shop
        self.name = name
        # inventory is a dictionary mapping each item to the amount
        # in the shop
        self.inventory = {}

    def print_inventory(self):
        '''
        Prints this shops inventory, where each item is followed
        by the amount of that item in the shop
        '''
        for item, amount in self.inventory.items():
            print(item, amount)

    def add_item(self, item):
        '''
        Adds an item to the store's inventory
        '''
        if item not in self.inventory:
            self.inventory[item] = 0
        self.inventory[item] += 1
```

CSV Dictreader

CSV Dictreader

Say you have the following csv called people.csv with the following contents:

```
id,name,age,height,weight
1,Alice,20,62,120.6
2,Freddie,21,74,190.6
3,Bob,17,68,120.0
```

people.csv

```
id,name,age,height,weight
1,Alice,20,62,120.6
2,Freddie,21,74,190.6
3,Bob,17,68,120.0
```

CSV Dictreader

Open the file by calling open and then csv.DictReader.

```
import csv
f = open("people.csv")
input_file = csv.DictReader(f)
```

people.csv

```
id,name,age,height,weight
1,Alice,20,62,120.6
2,Freddie,21,74,190.6
3,Bob,17,68,120.0
```

CSV Dictreader

When you iterate over input_file, you will see each row as a dictionary:

```
import csv
f = open("people.csv")
input_file = csv.DictReader(f)

for row in input_file:
    print(row)
```

people.csv

```
id,name,age,height,weight
1,Alice,20,62,120.6
2,Freddie,21,74,190.6
3,Bob,17,68,120.0
```

CSV Dictreader

When you iterate over input_file, you will see each row as a dictionary:

```
import csv
f = open("people.csv")
input_file = csv.DictReader(f)

for row in input_file:
    print(row)
```

Output:

```
{'age': '20', 'height': '62', 'id': '1', 'weight': '120.6', 'name': 'Alice'}
{'age': '21', 'height': '74', 'id': '2', 'weight': '190.6', 'name': 'Freddie'}
{'age': '17', 'height': '68', 'id': '3', 'weight': '120.0', 'name': 'Bob'}
```

CSV Dictreader

people.csv

```
id,name,age,height,weight
1,Alice,20,62,120.6
2,Freddie,21,74,190.6
3,Bob,17,68,120.0
```

When you iterate over input_file, you will see each row as a dictionary:

```python
import csv
f = open("people.csv")
input_file = csv.DictReader(f)

for row in input_file:
    print(row)
```

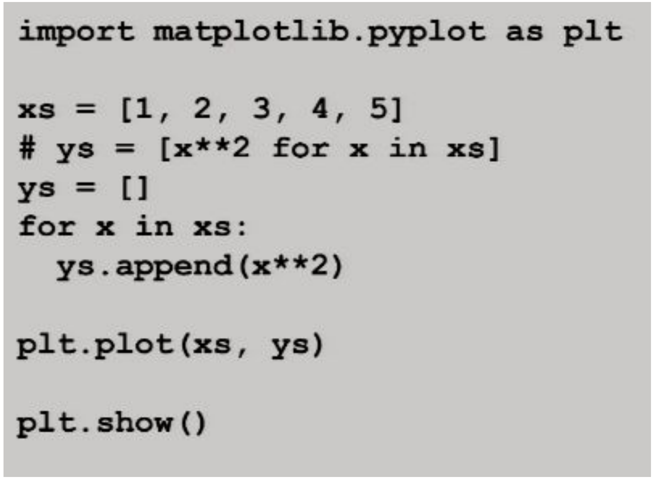
Output:

```
{'age': '20', 'height': '62', 'id': '1', 'weight': '120.6', 'name': 'Alice'}
{'age': '21', 'height': '74', 'id': '2', 'weight': '190.6', 'name': 'Freddie'}
{'age': '17', 'height': '68', 'id': '3', 'weight': '120.0', 'name': 'Bob'}
```

Note how the values in these dictionaries are always strings!

Plotting

- Use matplotlib to plot and visualize data
  - Pro: Powerful
  - Con: Not the best design for a plotting library

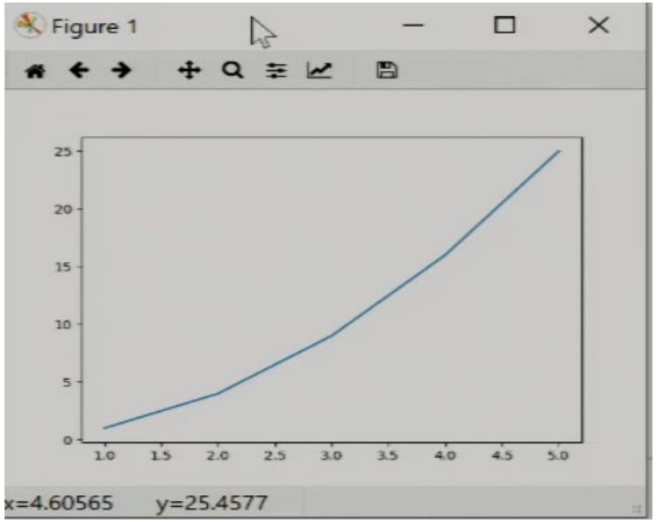


```
import matplotlib.pyplot as plt

xs = [1, 2, 3, 4, 5]
# ys = [x**2 for x in xs]
ys = []
for x in xs:
  ys.append(x**2)

plt.plot(xs, ys)

plt.show()
```



```
x=4.60565     y=25.4577
```

Problem 2

Download Starter Code from website