CS 111 *Session 11: Classes and Objects* *EM, November 7, 2019*

1. Write a class to represent a sphere. Implement a constructor, $getRadius(self)$, $volume(self)$, and $surfaceArea(self)$.

2. Consider the following description for a *Student* class.
   A student has a name, id number, graduation year, and a dictionary of classes they have taken and the grade points (4.0, 3.7, etc.) they received in the class.
   A student can graduate when they are a senior, taken 12 classes, and have a GPA of at least 2.0.
   When a *Student* object is created, assume they have not taken any classes yet.
   Implement a constructor, $addNewCourse(self, courseName, grade)$, $getGrade(self, courseName)$, $caluclateGPA(self)$, and $canGraduate(self)$

3. Will the following code produce an error? If not, what will it print?

```
class Fun():
    def __init__(self, foo, bar, fizz):
        self.x = foo
        self.y = bar
        self.z = fizz

    def total(self):
        return self.x + self.y + self.z

    def switch(self):
        self.x = self.y
        self.y = self.z
        self.z = self.x

    def __repr__(self):
        return "x: " + str(self.x) + ", y: " + str(self.y) + ", z: " + str(self.z)


firstFun = Fun(2,4,6)
secondFun = Fun(3,5,7)
thirdFun = firstFun
print(firstFun.total())
secondFun.total()
thirdFun.switch()
print(firstFun)
print(secondFun)
print(thirdFun)

apples = Fun([1,2], [4,5], [8,7,11])
oranges = Fun([0,0], [19,21], [1])
bananas = apples
bananas.total()
apples.switch()
print(apples.total())
print(oranges.total())
print(oranges)
print(apples)
print(bananas)
```

4. Write a simple *Book* class that has two instance variables: the name of the book and the year it was published.

5. Now, consider a dictionary, *authorBooks*, where the keys are authors' names and the keys are lists of *Book*s that they have written. Write snippets of code that accomplish the following things (assume that *authorBooks* has been defined appropriately):

   (a) Print out the names of all the authors in *authorBooks*

   (b) Print out the titles of all the books in *authorBooks*

   (c) Find the author that has written the most books and print out their name and the number of books they have written.

   (d) Find the book in *authorBooks* that was published first (earliest publication year) and print its author, title, and publication year

6. Write a function that takes as input a string and returns a list that contains the 3 letters that appear most frequently in the string:

7. What does the following code snippet print out?

```
xs = ["cs", "math", "econ", "history", "biology", "sociology"]

for x in xs:
    print(x)

for x in range(len(xs)):
    print(x)
```

8. Does the following code create an error? If not, what does it output?

```
pets = {
    "cats": 5,
    "dogs": 9,
    "birds": 2,
    "hamsters": 1
    "all pets": ["cats", "dogs", "birds", "hamsters"]
}

print("chinchilla" in pets)
pets["cats"] = 7
print("birds")
pets["dogs"] = pets["dogs"] + 1
for pet in pets["all pets"]:
    print(pets[pet])
```