**Warm up discussion problem:**

*The Wine Problem*
A king is having a party with 1000 bottles of wine. One month before the party, someone sneaks into the winery, and poisons one bottle of wine with poison that will show no signs that the person is poisoned, but 3 weeks later they will drop dead instantly.

The king has 10 servants that he do not care if they die, or live. Using these 10 servants, how does the king find the poisoned bottle of wine before the party?

# Workshop: *Classes*

1. Explain the similarities and differences between *instance* variables and *"regular"* variables.

2. Explain the following in terms of actual code that might be found in a class definition:

   (a) method
   (b) instance variable
   (c) constructor
   (d) initializer
   (e) accessor
   (f) mutator

3. Write code to declare a class `Person`. A person has a name and a phone number. What is the best way to represent a phone number? Include accessor, mutator, and initializer methods.

4. Assume that you have the `Person` class defined from the previous question. Write the code to declare a class `RockBand`. A rock band has four people (a singer, a drummer, a guitar player, and a bass player). Include accessor, mutator, and initializer methods. Are there different ways to define the initializer?

   Assume that you have the two classes defined in the previous questions. Write a main function that builds two different rock bands.

5. Modify the `Student` class, below, by adding a mutator method that records a grade for the student. Here is the specification of the new method:

   `addGrade(self, gradePoint, credits)` where `gradePoint` is a **`float`** value that represents a grade (e.g., A = 4.0, A- = 3.7 B+ = 3.3, etc.), and `credits` is a

**float** value indicating the number of credit hours for the class. Modify the student class by adding this grade information.

```
class Student:
    def __init__(self, name, hours, qpoints):
        self.name = name
        self.hours = float(hours)
        self.qpoints = float(qpoints)
    def getName(self):
        return self.name
    def getHours(self):
        return self.hours
    def getQPoints(self):
        return self.qpoints
    def gpa(self):
        return self.qpoints/self.hours
```

Use the updated class to implement a simple program for calculating GPA. Your program should create a new student object that has 0 credits and 0 quality points (the name is irrelevant). Then prompt the user to enter course information (`gradePoint` and `credits`) and then print out the final GPA achieved.

6. Extend the previous exercise by implementing an `addLetterGrade` method. This is similar to `addGrade` except that it accepts a letter grade as a string (instead of a grade point). Use the updated class to improve the GPA calculator by allowing the entry of letter grades.

7. Write a class to represent the geometric solid sphere. Your class should implement the following methods:

```
__init__(self, radius)  # Creates a sphere of a radius.
getRadius(self)  # Returns the radius of this sphere.
surfaceArea(self)  # Returns the surface area of the sphere.
volume(self)  # Returns the volume of the sphere.
```

Use your new class to write a program that calculates the surface and the volume of a sphere given its radius as input from the user.