**Python Code Challenges: Classes**

**Python Code Challenges Involving Classes**

This article will help you review Python classes by providing some interesting code challenges.

Some of these challenges are difficult! Take some time to think about them before starting to code.

You might not get the solution correct on your first try — look at your output, try to find where you’re going wrong, and iterate on your solution.

Finally, if you get stuck, use our solution code! If you “Check Answer” twice with an incorrect solution, you should see an option to get our solution code. However, truly investigate that solution — experiment and play with the solution code until you have a good grasp of how it is working. Good luck!

**Class Syntax**

As a refresher, class syntax looks like this:

class MyClass:  
    # ... class variables ...  
   
    def \_\_init\_\_(self):  
        # ... instance variables ...

For example, a class which defines a rectangle using a class variable, instance variables, and a method looks like this:

class Rectangle:  
    sides = 4  
   
    def \_\_init\_\_(self, width=0, height=0):  
        self.width = width  
        self.height = height  
   
    def calculate\_area(self):  
        return self.width \* self.height  
   
   
rectangle\_1 = Rectangle(5, 10)  
rect\_area = rectangle\_1.calculate\_area()

The last two lines in the above example show how to initialize an object of the class as well as calling one of the methods.

**Challenges**

You have decided to use your programming knowledge to create a new robotics company. Your idea for micro driving robots which are able to pick up and deliver objects was promising and now you want to start programming the logic. These code challenges will use your knowledge of Classes to solve some example scenarios. Try solving the five challenge problems below!

**1. Setting Up Our Robot**

Let’s begin by creating the class for our robot. We will begin by setting up the instance variables to keep track of important data related to the robot. Here are the steps we need to do:

1. Create a new class called **DriveBot**
2. Set up a basic constructor (no parameters needed)
3. Initialize three instance variables within our constructor which all default to 0: **motor\_speed**, **direction**, and **sensor\_range**