**Tutorial 13**

**CSE101: Introduction to Programming**

**Question 1:**

* Create a class, Triangle. Its \_\_init\_\_() method should take self, angle1, angle2, and angle3 as arguments. Make sure to set these appropriately in the body of the \_\_init\_\_()method.
* Create a variable named number\_of\_sides and set it equal to 3.
* Create a method named check\_angles which checks the sum of a triangle's three angles. It should return True if the sum of self.angle1, self.angle2, and self.angle3 is equal 180, and False otherwise.
* Create a variable named my\_triangle and set it equal to a new instance of your Triangle class. Pass it three angles that sum to 180 (e.g. 90, 30, 60).
* Print out my\_triangle.number\_of\_sides and print out my\_triangle.check\_angles().

**Question 2:**

* Define a new class named "Car". Inside your Car class, define a variable ‘condition’ and give it a default value of the string "new".
* Define the \_\_init\_\_() function of the Car class to take four inputs: self, model, color, and mpg. Assign the last three inputs to member variables of the same name by using the self keyword.
* Inside the Car class, add a method named display\_car() to Car that will reference the Car's member variables to return the string, "This is a [color] [model] with [mpg] MPG." You can use the str() function to turn your mpg into a string when creating the display string. Inside the Car class, add a method drive\_car()that sets self.condition to the string "used".
* Create a class ElectricCar that inherits from Car. Give your new class an \_\_init\_\_() method that includes a "battery\_type" member variable in addition to the model, color, and mpg. Then, create an electric car named "my\_car" with a "molten salt" battery\_type. Supply values of your choice for the other three inputs (model, color, and mpg).
* Inside ElectricCar add a new method drive\_car() that changes the car's condition to the string "like new".