**Assignments, W1D1**

Algorithmic Coding:

Problem 1: <https://leetcode.com/problems/reverse-string/> - Recursion, Strings

Problem 2: <https://leetcode.com/problems/swap-nodes-in-pairs/> - Recursion, Linked Lists

Problem 3: <https://leetcode.com/problems/reverse-linked-list/> - Recursion, Linked List

Object Oriented Programming:

Problem 1: Restaurant Class

Make a class called Restaurant. The \_\_init\_\_() method for Restaurant should store two attributes: a restaurant\_name and a cuisine\_type. Make a method called describe\_restaurant() that prints these two pieces of information, and a method called open\_restaurant() that prints a message indicating that the restaurant is open.

Make an instance called restaurant from your class. Print the two attributes individually, and then call both methods.

Problem 2: Restaurant Instances

Start with your class from Exercise 1. Create three different instances from the class, and call describe\_restaurant() for each instance.

Problem 3: User Class

Make a class called User. Create two attributes called first\_name and last\_name, and then create several other attributes that are typically stored in a user profile (username, email, location). Make a method called describe\_user() that prints a summary of the user’s information. Make another method called greet\_user() that prints a personalized greeting to the user.

Create several instances representing different users, and call both methods for each user.

Problem 4: Expanding the Restaurant Class

Start with your program from Exercise 1. Add an attribute called number\_served with a default value of 0. Create an instance called restaurant from this class. Print the number of customers the restaurant has served, and then change this value and print it again.

Add a method called set\_number\_served() that lets you set the number of customers that have been served. Call this method with a new number and print the value again.

Add a method called increment\_number\_served() that lets you increment the number of customers who’ve been served. Call this method with any number you like that could represent how many customers were served in, say, a day of business.

Problem 5: Login Attempts for Users

Add an attribute called login\_attempts to your User class from Exercise 9-3 (page 166). Write a method called increment\_login\_attempts() that increments the value of login\_attempts by 1. Write another method called reset\_login\_attempts() that resets the value of login\_attempts to 0.

Make an instance of the User class and call increment\_login\_attempts() several times. Print the value of login\_attempts to make sure it was incremented properly, and then call reset\_login\_attempts(). Print login\_attempts again to make sure it was reset to 0.