# PRACTICE WITH LISTS

CS 125

**Working with a partner/group, use the following steps to solve each of the following problems.**

(a) Plan your function on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases for your function. What input will you send to your function? What value should the function return?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your function, run your test cases. Does your function work? If not, how can you fix it?

1. Write a function `minimum(alist)` that returns the minimum of a list of numbers. Python has a built-in `min` function, but do not use it. Instead, iterate over the numbers in the list and keep track of the smallest number found.

   For example, `minimum([3, 7, 2, 4, 10])` returns `2`.

2. Write a function `isSorted(alist)` that determines whether a list of numbers is sorted in increasing order; that is, whether each number is less than or equal to the next number in the list. Your function should return `True` or `False`, depending on whether the list is sorted or not. For example:

   `isSorted([2, 5, 7, 8, 12, 15])` returns `True`
   `isSorted([2, 2, 2, 3, 3])` returns `True`
   `isSorted([2, 5, 10, 8, 12, 15])` returns `False`

3. Write a program that asks the user to enter some text. Then use the Python's string `split` method to split the text into a list of words. Then print out all of the five-character words that the user entered.

4. A standard deck of playing cards contains 52 cards. Each card has a *suit* and a *value*. The suits are *spades*, *hearts*, *diamonds*, and *clubs*. The values are 2, 3, 4, 5, 6, 7, 8, 9, 10, *Jack*, *Queen*, *King*, *Ace*. Write a program that produces a shuffled deck of cards.

   Begin by writing a function `makeDeck`. This function should use loops to build a list representing a 52-card deck. In the list, each card should be represented by a string such as `"2Clubs"` or `"JackDiamonds"`.

   Then write a function `shuffle(deck)` that accepts a list as a parameter and shuffles it. One way to do this is to take each item in the list and swap it with another item at a randomly-chosen index. (Do not use Python's `random.shuffle` function.)