

## **COP 2930 - Introduction to Computing Dictionaries - Suggested Exercises**

### Objective:

1. Practice writing programs that use dictionaries.

1) Write a program that manages a simple grocery store. Make the program menu driven where the user can add some quantity of an item to a dictionary storing the stock (if it's in stock, add to the current stock, if it's not, then make a new entry), or buy some quantity of an item, or ask how much of an item the store has left. Do reasonable error checking.

2) Write a function that takes in two dictionaries, where both dictionaries map people to their credit rating by two different bureaus. The function should return a new composite dictionary that has a single entry for each person in either dictionary. In particular, if a person is in both of the input dictionaries, their new mapped credit score should be the maximum of the two entries. Otherwise, just use the one credit score available. For example, if the first dictionary is {"Arup":798, "Bob": 799, "Carol": 844, "Danielle": 689, "Eddy": 756} and the second dictionary is {"Arup": 813, "Carol": 750, "Fran": 784, "George": 732} then the function should return the following dictionary: {"Arup":813, "Bob": 799, "Carol": 844, "Danielle": 689, "Eddy": 756, "Fran": 784, "George": 732}. The prototype for the function is below:

```
def creditMerge(experian, Equifax) :
```

3) Write a program that reads in a list of names, not all necessarily distinct, and assigns each name an ID number, starting with 0, in the order the names first appeared. First ask the user how many names they will enter. Then read in the names. After reading in each name, either print out a message that says the name was previously entered and print out its ID number, OR if the name wasn't previously entered, print out its newly assigned ID number. After all names are entered, let the user enter names and for each name entered, provide the ID number for that name, or state that the name isn't part of the database.