# CSC121 Lab 09: Dictionaries and Sets

## Goals

In this lab assignment, students will demonstrate the ability to:

* Create and use sets
* Create and use dictionaries

NOTE: This lab document is a copyrighted work of Wake Tech Community College and the course instructor. Any posting of this document outside of Wake Tech is considered a copyright violation. Students who post these documents outside of Wake Tech are subject to academic and possible legal actions.

## Instructions

In this lab, you will demonstrate your mastery of dictionaries and sets.

Follow the instructions in each problem and submit the specified files.

Problems 1 and 3 includes a starter file you will modify. Problem 2 will consist of a program that you create from scratch that meets the problem specification.

## Problems

### Problem 1

This program will demonstrate your ability to work with dictionaries.

A starter file called **Lab09P1-starter.py** has been provide on Blackboard. Download that file and rename it to **Lab09P1.py**.

The file consists of two functions.

* main() – DO NOT UPDATE THIS FUNCTION. This function reads in the file **words.txt** and stores the contents in a variable. That data is passed to the process\_file function.
* process\_file(file\_text) – This function contains several TODO comments that will direct you on what needs to be done.

The TODO comments in the process\_file function indicate to do the following:

* Create an empty dictionary.
* Create a list of words from the data passed to the function.
* Update the dictionary with the words as keys and the counts of the words as values.
* Print the dictionary.
* Determine the words with the maximum count, put them in a list, and print the list.
* Use the list to help remove all the words with maximum count from the dictionary.
* Put the words from the dictionary in a new list, sort it, and print that list.
* Remove the last line with the pass statement.

There is a sample word list on Blackboard called **words.txt**. It looks like this:

This is a word list

Giving this word list a test

See how this word list works

Test it and see how a word list file works

Using this file, here's the sample output a finished solution should produce:

{'THIS': 3, 'IS': 1, 'A': 3, 'WORD': 4, 'LIST': 4, 'GIVING': 1, 'TEST': 2, 'SEE': 2, 'HOW': 2, 'WORKS': 2, 'IT': 1, 'AND': 1, 'FILE': 1}

Words with max count of 4: ['WORD', 'LIST']

Dictionary with max removed: {'THIS': 3, 'IS': 1, 'A': 3, 'GIVING': 1, 'TEST': 2, 'SEE': 2, 'HOW': 2, 'WORKS': 2, 'IT': 1, 'AND': 1, 'FILE': 1}

Words sorted: ['A', 'AND', 'FILE', 'GIVING', 'HOW', 'IS', 'IT', 'SEE', 'TEST', 'THIS', 'WORKS']

Your program should include the additional text that indicates what the maximum count for the words is, and it should include the other lead-in text as well.

Submit the program file **Lab09P1.py** to Blackboard for credit.

### Problem 2

This program will demonstrate your ability to work with sets.

Create a file named **Lab09P2.py**. Write a program that does the following:

1. Generate 8 random integers between 1 and 16 inclusive using the randint function from the random module. Store the random integers in a set named set1. Display the set. Please note that the set may have less than 8 elements because some of the random integers generated may be duplicates.
2. Create another set named set2 in the same way set1 was created.
3. Find and display the union of set1 and set2. Make sure to include something in your output that indicates your output is for this step (see sample output).
4. Find and display the intersection of set1 and set2. Make sure to include something in your output that indicates your output is for this step (see sample output).
5. Find and display the difference of set1 and set2. Make sure to include something in your output that indicates your output is for this step (see sample output).
6. Find and display the symmetric difference of set1 and set2. Make sure to include something in your output that indicates your output is for this step (see sample output).
7. Use set comprehension to select the numbers less than 10 from the union set in Step c and store them in a set. Display this set. Make sure to include something in your output that indicates your output is for this step (see sample output).

Sample output:

set1: {1, 7, 10, 13, 16}

set2: {4, 5, 7, 8, 9, 13, 14}

Union of set1 and set2: {1, 4, 5, 7, 8, 9, 10, 13, 14, 16}

Intersection of set1 and set2: {13, 7}

Difference of set1 and set2: {16, 1, 10}

Symmetric difference of set1 and set2: {1, 4, 5, 8, 9, 10, 14, 16}

Less than 10 in union of set1 and set2: {1, 4, 5, 7, 8, 9}

Submit the program file **Lab09P2.py** to Blackboard for credit.

### Problem 3

In this problem, we redesign the inventory program for Trish's bookstore using dictionaries. Note: We will be continuing this redesign in Lab 10.

Copy the starter code **Lab09P3-starter.py** to a file named **Lab09P3.py**.

Remember in Lab 08, you used a list of lists to store the item information in the inventory system before writing it out to a file. You will change that now so that you are keeping the information in three dictionaries instead.

There are three TODO comments:

* At the first TODO comment, create three empty directories named inventory\_counts, inventory\_costs and inventory\_categories.
* At the second TODO comment, update the dictionaries with the user entered data. Use the item\_name as the key in all dictionaries and store the appropriate data.
  + For example, if the user first entered the item name "Science Books" with an item count of 10, a unit cost of 12.95, and a category "Book" this is what the dictionaries would look like:

inventory\_counts = {"Science Books": 10}

inventory\_costs = {"Science Books": 12.95}

inventory\_categories = {"Science Books": "Book"}

* At the third TODO comment, print the dictionaries.

Sample output:

Welcome to Trish's Inventory Input System

Enter the item name: Science Book

Enter the item count: 8

Enter the unit cost: 14.99

Enter the category: Book

Enter another item? (y/n) y

Enter the item name: Monopoly

Enter the item count: 10

Enter the unit cost: 17.95

Enter the category: Game

Enter another item? (y/n) y

Enter the item name: Die Hard

Enter the item count: 7

Enter the unit cost: 5.00

Enter the category: DVD

Enter another item? (y/n) n

Inventory counts: {'Science Book': 8, 'Monopoly': 10, 'Die Hard': 7}

Inventory costs: {'Science Book': 14.99, 'Monopoly': 17.95, 'Die Hard': 5.0}

Inventory categories: {'Science Book': 'Book', 'Monopoly': 'Game', 'Die Hard': 'DVD'}

Submit the program file **Lab09P3.py** to Blackboard for credit.

## Grading Rubric

### Grading rubric for Problem 1 (45 points)

* Program has a well-formatted and correct header [5 points]
* Program does execute correctly and produces correct results [40 points]

### Grading rubric for Problem 2 (25 points)

* Program has a well-formatted and correct header [5 points]
* Program does execute correctly and produces correct results [20 points]

### Grading rubric for Problem 3 (30 points)

* Program has a well-formatted and correct header [5 points]
* Program does execute correctly and produces correct results [25 points]