

## Recitation 09

### -

## Dictionaries & Sets

### Exercise 1 - (Gaddis 9.1) Course information

Write a program that creates a dictionary containing course numbers and the room numbers of the rooms where the courses meet. The dictionary should have the following **key** / **value** pairs:

Course Number ( <b>key</b> )	Room Number ( <b>value</b> )
CS101	3004
CS102	4501
CS103	6755
NT110	1244
CM241	1411

The program should also create a dictionary containing course numbers and the names of the instructors that teach each course. The dictionary should have the following **key** / **value** pairs:

Course Number ( <b>key</b> )	Instructor ( <b>value</b> )
CS101	Haynes
CS102	Alvarado
CS103	Rich
NT110	Burke
CM241	Lee

The program should also create a dictionary containing course numbers and the meeting times of each course. The dictionary should have the following **key** / **value** pairs:

Course Number ( <b>key</b> )	Meeting Time ( <b>value</b> )
CS101	8:00 a.m.
CS102	9:00 a.m.
CS103	10:00 a.m.
NT110	11:00 a.m.
CM241	1:00 p.m.

The program should let the user enter a course number, and then it should display the course's room number, instructor, and meeting time.

---

**Exercise 2 - (Gaddis 9.2) Capital Quiz**

---

Write a program that creates a dictionary containing the Chinese provinces as keys and their capitals as values. (See table below or parse file `chinese-provincial-capitals.txt` ([link](#)) The program should then randomly quiz the user by displaying the name of a province and asking the user to enter that province's capital. The program should keep a count of the number of correct and incorrect responses.

PROVINCE	CAPITAL
Anhui	Hefei
Fujian	Fujian
Gansu	Lanzhou
Guangdong	Guangzhou
Guizhou	Guiyang
Hainan	Haikou
Hebei	Shijiazhuang
Heilongjiang	Harbin
Henan	Zhengzhou
Hubei	Wuhan
Hunan	Changsha
Jiangsu	Nanjing
Jiangxi	Nanchang
Jilin	Changchun
Liaoning	Shenyang
Qinghai	Xining
Shaanxi	Xian
Shandong	Jinan
Shanxi	Taiyuan
Sichuan	Chengdu
Yunnan	Kunming
Zhejiang	Hangzhou

---

**Exercise 3 - (Gaddis 9.3) File Encryption and Decryption**

---

Write a program that uses a dictionary to assign "codes" to each letter of the alphabet. For example:

```
codes = {'A': '%', 'a': '9', 'B': '@', 'b': '#', ...}
```

Using this example, the letter **A** would be assigned the symbol **%**, the letter **a** would be assigned the number **9**, the letter **B** would be assigned the symbol **@**, and so forth.

The program should open a specified text file, read its contents, and then use the dictionary to write an encrypted version of the file's contents to a second file. Each character in the second file should contain the code for the corresponding character in the first file.

Write a second program that opens an encrypted file and displays its decrypted contents on the screen.

---

## Exercise 4 - (Gaddis 9.4) Unique Words

---

Write a program that opens a specified text file and then displays a list of all the unique words found in the file.

Hint: Store each word as an element of a set.

---

## Exercise 5 - (Gaddis 9.5) Word Frequency

---

Write a program that reads the contents of a text file. The program should create a dictionary in which the keys are the individual words found in the file and the values are the number of times each word appears. For example, if the word `the` appears 128 times, the dictionary would contain an element with `'the'` as the key and `128` as the value. The program should either display the frequency of each word or create a second file containing a list of each word and its frequency.

---

## Exercise 6 - (Gaddis 9.6) File Analysis

---

Write a program that reads the contents of two text files and compares them in the following ways:

- It should display a list of all the unique words contained in both files.
- It should display a list of the words that appear in both files.
- It should display a list of the words that appear in the first file but not the second.
- It should display a list of the words that appear in the second file but not the first.
- It should display a list of the words that appear in either the first or second file but not both.

Hint: Use set operations to perform these analyses.

---

## Exercise 7 - (Gaddis 9.7) World Series Winners

---

Start by downloading the data file `WorldSeriesWinners.txt` ([link](#)). This file contains a chronological list of the World Series' winning teams from 1903 through 2016. The first line in the file is the name of the team that won in 1903, and the last line is the name of the team that won in 2016. (Note that the World Series was not played in 1904 or 1994. There are entries in the file indicating this.)

Write a program that reads this file and creates a dictionary in which the keys are the names of the teams and each key's associated value is the number of times the team has won the World Series. The program should also create a dictionary in which the keys are the years and each key's associated value is the name of the team that won that year.

The program should prompt the user for a year in the range of 1903 through 2016. It should then display the name of the team that won the World Series that year and the number of times that team has won the World Series.