Introduction to Computer Science (ICS 3U)
Dictionaries

Save the programs in a folder called **Dictionaries**.

1. Create a program which will create a dictionary with at least 4 keys (fields) for a person, place or thing - make it up. It will then have the user input data and have it outputted. It will then ask the user to input a key and have the corresponding value outputted. Call your program **Dictonary1.py**

2. a) Create a program that will allow the user to input data into a file called **Music.txt** to keep track of the albums they have. The data inputted into the file must include the following keys (fields): name, artist(s), format (ie. record, mp3, cd) and favourite song. The user must be able to add as many albums as they wish. Re-save your program **Dictonary2.py**

   b) Add a section to your program to output the contents of the **Music.txt** file using a dictionary. Re-save your program **Dictonary2.py**

3. Create a program that will accept data from a file called **Teachers.txt** located in the Student Share Folder. The file contains the name of the teacher, the department they teach in and the subject they teach for 6 teachers. The user should be able to input the name of a department and have the names of all the teachers who teach in that department and their subjects outputted. Output an appropriate message if there are no teachers in the department inputted by the user. Use a dictionary. Call your program **Dictonary3.py**

   Hint: See an example program that will help with this program.

   Level 4: Have the list of the departments in the file appear before the user inputs their choice. Each department name should only appear once.

Note that you must watch the video about “Lists of Dictionaries” before moving on.

4. a) Create a program that will input the data from **Teachers.txt** . It will then output all the data in a table sorted in alphabetical order by teacher name. Call your program **Dictonary4.py**

   b) Allow the program to work for a file with any number of users.

5. Create a menu driven program that will load the data from the a **Students.txt** file found in the Student Share folder. This file has the names of students, the number of absences they have and their marks in English, math, religion and science. The number of students changes constantly so the program should handle any number of students and allow the user to do any or all of the following:

   - Have the overall average of all the students displayed on the monitor.

   - Allow the user to input the student name and have that student’s name and average outputted on the monitor.

Continued on next page

- Output the names and averages of all of the students who made the Honour Roll (an average of 80 or better) to the monitor and to a file called **Honour.txt**

- Output a list of the student names, marks and averages sorted by highest to lowest average.

- Add a student to the file

When the program ends, the Students.txt file should be overwritten with any new data that the user has inputted. Call your program **Dictonary5.py**

Hints: 1) Create a menu-driven program for this question.

2) Create a void function to input the data and do all of the tasks listed in the menu items. If required, the parameter should be a list of dictionaries.

3) The **Students.txt** data file should only be accessed when the list is loaded and at the end of the program when the file is updated.

4) **Hint**: See an <u>example program</u> that will help with this program.

6. a) Create a data file called **Questions.txt** which will store information in record form about the questions and answers to a multiple choice quiz. The for each question should be organized as follows:
   - the question
   - the four choices (one answer per line)
   - the correct answer

   An example record would be:

   Who will win the Stanley Cup this year?
   Toronto Maple Leafs
   Ottawa Senators
   Detroit Red Wings
   Chicago Black Hawks
   Ottawa Senators

   Make sure you have at least **five** questions.

   b) Create a program which will read the data from the file and store it <u>in a dictionary</u>. The program will display the question and the possible answers on the monitor. The user will then input their answer and the program will output whether they are correct or not. If they are incorrect, the correct answer will be outputted. Finally, the number of correct answers they have will be outputted. Why is a list **not** required for this program? Call your program Call your program **Dictonary6.py**

7. a) Create a program which will be an improved version of your quiz program. It will output the questions <u>in random order</u>. Why will you have to use a list in this program? Call your program **Dictonary7.py**

   b) Improve your quiz program so that there are no questions repeated *(Level 4)*.
   Hint: This will require another list to keep track of the questions asked.