

CompSci101 Laboratory 08 — Dictionaries

EXERCISES

EXERCISE 8.1 – USING A DICTIONARY

This program prompts for and inputs a letter from the user. The program then prints a baby girl name and a baby boy name that starts with the input letter. Complete the Lab08Ex1 program by writing the following functions:

- **get_letter_from_user()**

Returns: the letter entered by the user converted to uppercase.

This function prompts for and inputs a letter from the user as in the screenshot below. It then returns the letter in uppercase.

- **print_baby_name(letter, baby_names_dictionary)**

Parameters: The user's letter and a baby names dictionary

Prints: The baby name that starts with the parameter letter

Example outputs from program if the user enters "a" then "b" at the prompts:

```
Enter a letter: a
Baby name : Alex
Baby name : Abby
```

```
Enter a letter: b
Baby name : Benjamin
Baby name : Bella
```

EXERCISE 8.2 CREATING A DICTIONARY

This program creates a baby names dictionary. Complete the Lab08Ex2 program by writing the following functions:

- **get_list_of_names(filename)**

Parameter: the filename of a file of names

Returns: a list of names

This function takes a filename as a parameter, opens and reads the contents of the file, and returns a list of all the names in the file. You will need to use the string `split()` method to split the file contents (by whitespace) into a list.

- **create_baby_names_dict(names_list)**

Parameter: a list of baby names

Returns: a baby names dictionary

This function takes a list of baby names as a parameter, and creates a dictionary by looping through each element in the list and creating a corresponding **key:value** item. The “key” of each item in the dictionary is the first letter of the baby name, and the “value” is the baby name (e.g. key of “A”, value of “Alex”). You can assume that the file contains 26 lines and each line contains only one name. Each name starts with a unique first letter.

- **display_baby_names(baby_names_dictionary)**

Parameters: The baby names dictionary (**key : value** is **initial letter : name**)

Prints: The contents of the baby names dictionary, formatted as in the example below.

This function takes the baby names dictionary as a parameter, and prints the contents of the dictionary. Note that there is a comma and a space after each name (including the last name).

Example outputs from program if the user enters “baby_boy_names.txt” at the prompts:

```
Enter a filename: baby_boy_names.txt
A Alex, B Benjamin, C Connor, D Daniel, E Ethan, F Finn, G
George, H Hunter, I Isaac, J Joshua, K Kingston, L Liam, M Max, N
Noah, O Oliver, P Phoenix, Q Quinn, R Riley, S Samuel, T Thomas,
U Ute, V Victor, W William, X Xavier, Y Yukio, Z Zachary,
```

EXERCISE 8.3 – BABY NAMES COUNTS ANALYSIS

This program produces a summary of the top 10 baby names. Complete the Lab08Ex3 program by writing the following functions:

- **get_list_of_names(filename)**

Parameter: the filename of a file of names

Returns: a list of words

This function takes a filename as a parameter, opens and reads the contents of the file, and returns a list of all the names in the file. You will need to use the string `split()` method to split the file contents (by whitespace) into a list.

- **create_names_count_dictionary(names_list)**

Parameter: a list of baby names

Returns: a dictionary of baby name counts (**key : value** is **name : count**)

This function takes a list of baby names as a parameter and creates a dictionary of baby name counts. Each dictionary item will contain a **key** of the baby name and the corresponding **value** will be a count of all the times the baby name appeared in the list.

- **produce_name_counts_report(name_counts_dict)**
 Parameter: a dictionary of name counts (*key : value* is **name : count**)
 Prints: a report of baby name counts

This function is passed a dictionary of baby name counts as a parameter. It produces an analysis of the baby name counts, formatted as in the screenshot on the next page. The names are to be printed in ascending alphabetical order.

Expected output from program if you have coded it correctly:

```
Enter a filename: top10_boy_names.txt
Report
=====
Charlie appears 192 times
George appears 232 times
Jack appears 251 times
James appears 210 times
Leo appears 257 times
Lucas appears 247 times
Noah appears 297 times
Oliver appears 335 times
Thomas appears 202 times
William appears 210 times
```

EXERCISE 8.4 – ALPHABET-BABY NAMES DICTIONARY

This program reads in a file of baby names and prints a list of baby names by initial letter as in the screenshot on the next page. Complete the following functions:

- **get_list_of_names(filename)**
 Parameter: the filename of a file of baby names
 Returns: a list of baby names

This function takes a filename as a parameter, opens and reads the contents of the file, and returns a list of all the baby names in the file. You will need to use the string `split()` method to split the file contents by whitespace into a list.

- **create_baby_names_dictionary(names_list)**

Parameter: a list of baby names

Returns: a dictionary of baby names
(*key : value* is **initial letter : a list of baby names**).

This function takes a list of baby names as a parameter and creates a dictionary. Each dictionary item will contain a “**key**” of the initial letter of the baby name and the corresponding “**value**” will be a list containing all the baby names that start with that particular letter.

- **display_baby_names(baby_names_dict)**

Parameter: a dictionary of baby names
(*key : value* is **initial letter : a list of baby names**)

Prints: baby names grouped by initial letter in ascending order.

This function takes a dictionary of baby names keyed by the initial letter of the baby names, and prints the names as per the screenshot below. Note that you will first need to produce a list of the keys and use the **sort()** method to sort the keys, then loop through this list of sorted keys. Each list of names will also need to be sorted.

Expected output from program if you have coded it correctly:

```
Enter a filename: 2019Top_boy_names.txt
A: Adam Aiden Alexander Angus Archer Archie Arlo Arthur Asher Ashton
Austin
B: Beau Benjamin Blake Bodhi Braxton
C: Caleb Carter Charles Charlie Connor Cooper
D: Daniel David Dylan
E: Edward Eli Elijah Elliot Ethan Ezekiel Ezra
F: Felix Finn Flynn
G: Gabriel George Grayson
H: Harley Harrison Harry Harvey Henry Hudson Hugo Hunter
I: Isaac Isaiah
J: Jack Jackson Jacob James Jasper Jaxon Jayden John Jordan Joseph Joshua
L: Lachlan Leo Leon Levi Liam Lincoln Logan Louie Louis Luca Lucas Luka
Luke
M: Mason Matthew Max Micah Michael
N: Nathan Nikau Noah
O: Oliver Oscar
P: Phoenix
R: Reuben Riley Roman Ryan Ryder
S: Samuel Sebastian
T: Theo Theodore Thomas Toby Tyler
W: William Wyatt
X: Xavier
Z: Zachary Zion
```