COSC 1336 – Program 6

- Use the course material located at:
  - COSC 1336
- Program 6 can be completed using content from the following chapters:
  - Welcome! through Data Types - Sets

Requirements (remember to identify the requirements by *number*):

- Create a file using PyCharm.
- Name the source file program_6.py.
- Format code like the examples.
- Include comments in your code like the examples.
- Code the requirements in the numbered list below.

COSC 1336 – Program 6

Requirements to be numbered in program_6.py:

Write a program for a Honda regional office manager. Note: There is no data hard-coded in the program except for the two tuples. *All data is entered by the user. See the code example at the end of this document.*

1. Output a header in the console: “This is Program 6 - <yournamehere>”
2. Print “This program practices working with lists, strings, tuples, and dictionaries.”
3. Create two tuples - one for two weeks named “Week One” and “Week Two” and one for three days of the week (Thursday, Friday, Saturday).
4. Use a loop to record the first name and last name of sales people in one string (e.g. “Bill Simms”) and the name of the dealership where they work in another string (e.g. “Hero Honda”).
5. In each iteration of the loop, create a dictionary entry for the salesperson data entered where the name is the ‘key’ and the dealership is the ‘value’.
6. After the dictionary has been populated, create two nested loops to process sales data.
7. The outer loop uses the weeks tuple and the inner loop uses the days tuple.
8. In the inner loop, ask the user to enter the number of cars sold by salesperson (display the salesperson’s name to the user). Store the user’s response in a list of sales data. For each salesperson, there should be six entries in the list.
9. Use a loop to have the user populate a list of 20 customer compliments. Why 20? It is just a number for testing. Include sales rep names in each of the compliments.
10. Use nested loops to process the customer compliments. The outer loop cycles through the sales reps and the inner loop cycles through the compliments. Use a sequence operation to search for each sales rep’s name in each of the compliments.

For instance, the outer loop is set to the first sales rep. The inner loop searches for the sales rep's name (the key in the dictionary) in compliment one, then compliment two, etc. Record the number of compliments associated ***per each sales rep*** (must be ***per each sales rep***, NOT total).

More on the compliments section:

- User enters a compliment
- The compliments include the sales rep names
- After collecting all compliments
- Will need nested loops (outside loop for each sales rep; inner loop for each compliment)
- Search for the name of each sales rep in each compliment
- Count the number of times you find a compliment for each sales rep

At the end of the program, produce output like that below for each of the sales reps processed.

11. All data is based on user entry (except tuples). Sample output:

Bill Simms of Hero Honda
Cars Sold
Week One
Thursday: 3
Friday: 4
Saturday: 7

Week Two
Thursday: 5
Friday: 8
Saturday: 9

Total Cars Sold (per rep): _____
Average Cars Sold (per rep): _____
Number of Compliments (per rep): ____

There should be a report like that shown at the END of the program that displays sales data for each of the sales reps.

12. Print a statement explaining your experiences with Program 6.

TEST – TEST – TEST your application to ensure the specific program requirements are met.

- Use the list above and the common requirements as a confirmation checklist.
- Not meeting all requirements = 0 points for the assignment.

Code example:

Note: This is NOT the solution. However, it should be VERY helpful if you have questions about how to implement program_6.

program_6.py ×

```
weeks = ('Week One', 'Week Two')
days = ('Thursday', 'Friday', 'Saturday')
# This is just test data. You will need to write a loop to enter data like this.
sales_reps = {'Frances Fleming': 'Hero Honda',
              'Domingo Depue': 'Highlight Honda',
              'Ema Endicott': 'Home Run Honda'}

cars_sold_thursday = []
cars_sold_friday = []
cars_sold_saturday = []

for week in weeks:
    print('\n-----Entering cars for {} -----'.format(week))
    for key in sales_reps.keys():
        cars_sold_thursday.append(int(input('\nThursday cars sold by {}: '.format(key))))
        cars_sold_friday.append(int(input('\nFriday cars sold by {}: '.format(key))))
        cars_sold_saturday.append(int(input('\nSaturday cars sold by {}: '.format(key))))

print('\n############ Printing Sales Results ############')
for week in weeks:
    print('\n############ {} Results ############'.format(week))
    j = 0
    for key in sales_reps.keys():
        print('\n{} sold {} on Thursday'.format(key, cars_sold_thursday[j]))
        print('\n{} sold {} on Friday'.format(key, cars_sold_friday[j]))
        print('\n{} sold {} on Saturday'.format(key, cars_sold_saturday[j]))
        j += 1
```