MGMT 58600 Python Programming

Lab 3

# Lab Purpose:

The purpose of this lab is to provide more experience with Python lists and dictionaries.

# Lab Instructions:

## Description:

Krannert Specialty Motors is a unique auto dealer that sells both brand new and “lovingly driven” high-end automobiles. In addition to car sales, KSM has been experimenting for the year with the idea of leasing cars (using some of the “lovingly driven” high-end automobiles as its inventory) as well and has a subsidiary called Krannert Specialty Motors Leasing (KSML). Understanding and maintaining the fleet of lease vehicles is important to KSML and sometimes the inventory manager needs to look at historic data to make decisions about the future. The inventory manager needs to be able to look at both aggregate and specific data about their cars and make some decisions. Currently there is a small inventory of rough 20-25 cars, but that may grow in the future if this venture proves success. The cars have unique names based on the type of car or its features (e.g., the name of the Fiat 124 Spider is “spider,” the name of the TR7 is “triangle,” the name of the Alfa Romeo Stelvio Quadrifoglio is “shamrock,” the name of the the Delorean DMC-12 is “marty,” etc. The aspects of each car that are tracked include actions like general maintenance (general upkeep), fixes (minor upkeep), upgrades (major improvements), and others. All cars also have other data about them tracked as well including things like number of oil changes, number of bodywork incidents, number of major mechanical issues, unique items tied to type of car (e.g., an import/export waiver if a foreign car … the value will be 1), number of enhancements (these are items done to the car to increase its value like a new paint job or added enhancements to make the engine more powerful) and the agent id. The agent id is the only data that all cars must have. The other items only have an entry if the issue has happened (e.g., if the car is a domestic car, it will not have an import/export waiver or, if the car has never had the oil changed, there will be no number of oil changes entry).

You will write program that allows a manager to query car data to provide information about:

* A specific car and a specific action/activity
* Aggregate information about an action/activity across all cars (e.g., the total number of oil changes across all cars or the total number of major mechanical issues across all cars)

Some general ideas about the program:

1. Your program must have a single dictionary to contain the data about the properties and the activities/actions (hint: you need a nested dictionary to do this)
2. Your dictionary must be loaded with at least four cars (at a minimum you must include “spider,” “triangle,” “shamrock,” and “marty”). Each car must have at least two different activities/actions associated with it (recall that ALL cars have an agent id so that counts as one) and at least one car must have three activities/actions (in this car at a minimum you must have number of oil changes and number of bodywork incidents entries). Each action will have the number of times that action was completed in the last year on that car. For example, spider may have 2 oil changes and 4 bodywork incidents or shamrock may have an import/export waiver (again, the value will be 1) and 1 oil change. You may have more than the minimum should if desired and time permits.
3. Your program must do two things as specified above:
   1. Aggregate information about an action/activity across all cars (e.g., the total number of oil changes across all cars or the total import/export waivers across all cars)
      1. This part will ask the user what action they wish to get aggregate information about.
      2. After receiving the input action, the program will then look through the dictionary and return a total of how many times that action has been done across all cars.
      3. You must use a function to process the dictionary to get the counts (hint: you’ll need to pass the action that you are looking for as well as the dictionary to accomplish this)
      4. The output should look similar to the following: oil changes ----> 54
   2. Provide a specific car and the number of times a specific action has been done at that car (along with the agent’s name for that car)
      1. This part will ask the user for a car and which action for that car that they wish to get a count for (i.e., how many times that action was performed on that car)
      2. After receiving the property and the action, your program will then get then use the dictionary to see how many times that action was done on that specific car.
      3. You will also need to convert the agent id into the actual name of the agent (agent id at our company start a zero and progress upward as we add agents). The agent names must be maintained in a list and you must use a list to convert them.
      4. The output should look similar to the following: spider had 3 oil changes and the agent is Mary Jones

If a user is looking up oil changes and a specific car has more than 5 oil changes, you should print a warning for the user under the output in 3.b.iv (above) that is similar to “**Warning: You should probably have the mechanic check this engine.**”

**Grading Criteria**

|  |  |
| --- | --- |
| **Element** | **Points** |
| Proper use of function with return to get aggregate information across all cars | 20 |
| Proper dictionary including correct data inclusion | 15 |
| Proper processing reporting of specific action for specific property | 15 |
| Warning for too many oil changes | 10 |
| Proper use of list (including initial values) to convert agent codes into agent names | 15 |
| Program executes correctly | 15 |
| Comments | 10 |
| Total Points Possible | 100 |