```
"""
cars_two_dim_extra_credit.py
@author: Sherri Vaseashta
Created Current Date
This script opens the Automobile_data_ITP150.csv file and stores each
column in the file into a two dimensional list representing manufacturer,
model, mpg, and price with throwback data from the 1970s. We also count
the number of cars for each manufacturer in two different ways. One is
the long way using nested for loops and the other uses Python's collections
library which is much shorter.
"""


import csv
import collections


def main():

    cars_two_dim = [[]]  # Initializing a 2 dimensional list
    manufacturers_dictionary = {}  # Initializing an empty dictionary
    cars_two_dim = read_the_file(cars_two_dim)
    print_the_list(cars_two_dim)
    manufacturers_dictionary = get_a_dictionary_of_manufacturers(cars_two_dim)
    # Method 1 to count the manufacturers which has the most code in it
    manufacturer_counts_using_dictionary(manufacturers_dictionary, cars_two_dim)
    # Method 2 to count the manufacturers using collections library. shorter
    manufacturer_counts_using_collections(manufacturers_dictionary)
    write_manufacturers_dictionary_to_file(manufacturers_dictionary)


def read_the_file(cars_two_dim):
    # Open the input file
    with open("Automobile_data_ITP150.csv", newline='') as cars_file:
        cars_reader = csv.reader(cars_file, delimiter=',')
        cars_two_dim = [row for row in cars_reader]

    # make the mpg and price columns into integers and floats respectively
    for row in range(len(cars_two_dim)):
        cars_two_dim[row][2] = int(cars_two_dim[row][2])
        cars_two_dim[row][3] = float(cars_two_dim[row][3])
    print(cars_two_dim)
    print("Number of rows in the list:", len(cars_two_dim))

    return cars_two_dim


def print_the_list(cars_two_dim):
    print(f'{"Manufacturer":<20} {"Model":>20} {"MPG":>10} {"Price":>10}')
    for row in range(len(cars_two_dim)):
        print(f'{cars_two_dim[row][0]:<20s} {cars_two_dim[row][1]:>20} \
{cars_two_dim[row][2]:>10d} {cars_two_dim[row][3]:>15,.2f}')


def get_a_dictionary_of_manufacturers(cars_two_dim):
    # Create a set with that has the manufacturer names which are in column 0
    manufacturers_set = set()  # This initializes an empty set.
    # The loop below goes through each row in the cars_two_dim list.
    # It saves the manufacturer names which are at current row and column[0]
    # into a variable called item. We then use the set add method to add
    # the item which is the manufacturer name to the manufacturer set.
    # Since sets don't allow duplicate values, we will get the manufacturer
    # name only 1 time, thus unique manufacturer names.
```

```
    for row in range(len(cars_two_dim)):
        item = cars_two_dim[row][0]
        manufacturers_set.add(item)
    print('Manufacturers Set', manufacturers_set)

    # Now that we have all of the manufacturer names, we want to make
    # the manufacturers set into the keys of a dictionary. Then we can
    # count the values for each manufacturer and store that as the value
    # for the key. Remember that dictionaries store items in key value pairs.
    # The line below converts a set to a dictionary and initializes the count
    # for each manufacturer to 0 for now. We will count the counts later.
    # In addition, dictionaries do not allow duplicate keys so each
    # manufacturer is unique.
    manufacturers_dictionary = dict.fromkeys(manufacturers_set, 0)

    # Printing manufacturers dictionary.
    print("Manufacturers Dictionary", manufacturers_dictionary)
    # At this point we have a dictionary with the manufacturer names and
    # we need to count the values for each manufacturer
    return manufacturers_dictionary


def manufacturer_counts_using_dictionary(manufacturers_dictionary, cars_two_dim):
    # Now for each key (manufacturer) in the manufacturers dictionary,
    # we use a for loop to step through each key. We initialize a count
    # = 0. The the inner for loop is stepping through each row in the
    # cars_two_dim_list in order to count the manufacturers. Remember
    # that the manufacturer names are in column 0 for each row in
    # the cars_two_dim list.
    for each_key in manufacturers_dictionary:
        print(each_key)  # Using this print to help you see the logic in action
        count = 0
        for each_row in range(len(cars_two_dim)):
            # The if statement below is comparing the manufacturer name
            # which is at column 0 in whatever row we are at to the
            # key from the manufacturers dictionary to see if they are
            # equal. If so, it will add 1 to the count for the manufacturer.
            if cars_two_dim[each_row][0] == each_key:
                count = count + 1
                print('manufacturer is', each_key, 'count is', count)
                manufacturers_dictionary[each_key] = count  # This stores
                # the count as the value for the key which is the manufacturer

    print('Manufacturers dictionary', manufacturers_dictionary)
    x = sum(manufacturers_dictionary.values())  # Verifying by summing values
    # from the dictionary to see if the sum is equal to the length of the
    # cars_two_dim list of 61. They should be equal since we are counting
    # each time we have a row for a manufacturer in the cars_two_dim list.
    print(x)
    # Printing prettily the counts for each manufacturer
    print("Counts for each Manufacturer")
    print(f'{"Manufacturer":<20s}{"Count":>20s}')
    for x, y in manufacturers_dictionary.items():
        print(f'{x:<20s}{y:>20d}')


def manufacturer_counts_using_collections(manufacturers_dictionary):
    print('Counts for each Manufacturer with Collections')
    manufacturers_dict = collections.Counter(manufacturers_dictionary)
    print(manufacturers_dict)
    sum_values = sum(manufacturers_dict.values())  # checking for sum of 61
    print('sum values', sum_values)
    print("Counts for each Manufacturer")
```

```
    # printing prettily
    print(f'{"Manufacturer":<20s}{"Count":>20s}')
    for x, y in manufacturers_dict.items():
        print(f'{x:<20s}{y:>20d}')


def write_manufacturers_dictionary_to_file(manufacturers_dictionary):
    manufacturers_file = csv.writer(open('manufacturers_file.csv', 'w'))
    for key, val in manufacturers_dictionary.items():
        manufacturers_file.writerow([key, val])
    print("The manufacturers_file.csv has been updated.")


# Call the main function
if __name__ == '__main__':
    main()

```