

GitHub link:

<https://github.com/CWestLBCC/CS161>

For this project I created a program that manages a community garden and the garden information.

I will use this project and expand on it for the Final Project. I want to make sure every line item is covered for this Project #6 assignment that may not be in the final program.

1. Using dictionaries show each of the following.

- 1) Create

In this section the Garden Dictionary was created and then printed.

Code:

```
6 #Source: https://www.w3schools.com/python/python_dictionaries.asp
7 def create_garden_dict():
8     """This function creates a dictionary of garden items."""
9     garden_dict = {
10         "flowers": ["rose", "dahlia", "scotch broom", "daffodil"],
11         "vegetables": ["tomato", "zucchini", "carrot"],
12         "fruits": ["apple", "plum", "cherry"]
13     }
14     return garden_dict
```

```
116 #Here is the main functions
117 def main():
118     #Create garden dictionary.
119     garden_dict = create_garden_dict()
120     print("Garden Dictionary:", garden_dict)
```

Output:

```
Garden Dictionary: {'flowers': ['rose', 'dahlia', 'scotch broom', 'daffodil'], 'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['apple', 'plum', 'cherry']}
```

- 2) Add more data

In this section -Sunflower- was added to the Garden dictionary.

Code:

```
16 def add_data(garden_dict, category, data):
17     """This function adds more data to the garden dictionary."""
18     if category in garden_dict:
19         garden_dict[category].append(data)
20     else:
21         print(f"Category '{category}' does not exist in the garden dictionary.")
22
```

```

88     ...# This will add -Sunflower- to the flowers set.
89     ...flower_set.add("sunflower")

```

```

149     ...# This is how I could add more data -Sunflower- to the garden dictionary.
150     ...add_data(garden_dict, "flowers", "sunflower")
151     ...print("\nAfter adding -Sunflower- to the flowers dictionary:", garden_dict)

```

Output:

```

After adding -Sunflower- to the flowers dictionary: {'flowers': ['rose', 'dahlia', 'scotch broom', 'daffodil', 'sunflower'],
'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['apple', 'plum', 'cherry']}

```

- 3) Change a value associated with a key in the dictionary.

In this section -Scotch Broom- was changed to -Peony- in the Flowers dictionary.

Code:

```

23 def change_value(garden_dict, category, old_data, new_data):
24     ..."""This function changes a value that is associated with a key in the dictionary."""
25     ...if category in garden_dict:
26     ...    ...if old_data in garden_dict[category]:
27     ...        ...index = garden_dict[category].index(old_data)
28     ...        ...garden_dict[category][index] = new_data
29     ...    ...else:
30     ...        ...print(f"'{old_data}' does not exist in the '{category}' category.")
31     ...    ...else:
32     ...        ...print(f"Category '{category}' does not exist in the garden dictionary.")

```

```

153     ...# This is how I am able to change a value associated with a key in the garden dictionary.
154     ...change_value(garden_dict, "flowers", "scotch broom", "peony")
155     ...print("\nAfter changing -Scotch Broom- to -Peony- in the flowers dictionary:", garden_dict)

```

Output:

```

After changing -Scotch Broom- to -Peony- in the flowers dictionary: {'flowers': ['rose', 'dahlia', 'peony', 'daffodil', 'sunflower'],
'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['apple', 'plum', 'cherry']}

```

- 4) Remove data

In this section -Apple- was removed from the Fruits dictionary.

Code:

```

34 def remove_data(garden_dict, category, data):
35     ..."""Function to remove data from the garden dictionary"""
36     ...if category in garden_dict:
37     ...    ...if data in garden_dict[category]:
38     ...        ...garden_dict[category].remove(data)
39     ...    ...else:
40     ...        ...print(f"'{data}' does not exist in the '{category}' category.")
41     ...    ...else:
42     ...        ...print(f"Category '{category}' does not exist in the garden dictionary.")

```

```

157     ...# This is how data can be removed from the garden dictionary.
158     ...remove_data(garden_dict, "fruits", "apple")
159     ...print("\nAfter removing 'Apple' from fruits dictionary:", garden_dict)

```

Output:

```

After removing 'Apple' from fruits dictionary: {'flowers': ['rose', 'dahlia', 'peony', 'daffodil', 'sunflower'], 'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['plum', 'cherry']}

```

- 5) Index the dictionary to find some value stored at a key within it.
In this section the value at index 2 of the Flowers dictionary was located and identified as -Peony-.

Code:

```
44 def index_dictionary(garden_dict, category, index):
45     """Function to index the dictionary to find some value stored at the key within it."""
46     if category in garden_dict:
47         if index < len(garden_dict[category]):
48             return garden_dict[category][index]
49         else:
50             print(f"Index '{index}' is out of range for the '{category}' category.")
51     else:
52         print(f"Category '{category}' does not exist in the garden dictionary.")

161 # Index the dictionary to find some value stored at a key within it.
162 print("\nValue at index 2 of flowers dictionary: ", index_dictionary(garden_dict, "flowers", 2))
163
```

Output:

```
Value at index 2 of flowers dictionary: peony
```

- 6) Create a function that takes a dictionary and accomplish something similar to a built in function (min, max, mean, sum or comparison) without using any building math methods.

In this section I could not figure out how to make the code work no matter how much I troubleshoot. It throws off the other code. I've included the troublesome commented out code and the code I settled on using. It calculates the shortest category in the Garden Dictionary.

Code:

Troublesome code.

```
54 # I could not get this to work no matter how much I troubleshoot... It's throwing off the other code.
55 #def garden_no_builtin_function(garden_dict, operation):
56 #    """Function that creates a function that takes a dictionary and accomplishes something similar to a built-in function."""
57 #    if operation == "min":
58 #        return no_builtin_min(garden_dict)
59 #    elif operation == "max":
60 #        return no_builtin_max(garden_dict)
61
62 #def no_builtin_min(garden_dict):
63 #    min_value = None
64 #    for key, value in garden_dict.items():
65 #        if min_value is None or len(value) < min_value:
66 #            min_value = len(value)
67 #    return min_value
68
69 #def no_builtin_max(garden_dict):
70 #    max_value = None
71 #    for key, value in garden_dict.items():
72 #        if max_value is None or len(value) > max_value:
73 #            max_value = len(value)
74 #    return max_value
75
```

Code I resolve to use instead.

```
76 def garden_function(garden_dict, operation):
77     """Function to create a function that takes a dictionary and accomplishes something similar to a built-in function."""
78     if operation == "min":
79         return min(len(v) for v in garden_dict.values())
80     elif operation == "max":
81         return max(len(v) for v in garden_dict.values())
```

Output:

```
The length of the shortest category in garden dictionary is: 2
```

- 7) Use a couple of methods to accomplish some task (or your own functions).

In this section I totaled the different types of plants grown in the community garden. This did not include the fruit trees since they are a Tuple.

Code:

```
167 ...# The total quantity of inventory.
168 ...total_quantity_garden = sum(len(v) for v in (garden_dict["flowers"], garden_dict["vegetables"], garden_dict["fruits"]))
169 ...print("\nTotal Quantity using sum:", total_quantity_garden)
```

Output:

```
Total Quantity using sum: 10
```

2. Use Sets in some tasks similar to the above, make special note in code when we cannot perform task exactly the same due to differences between Dictionary and Set objects behave.

In this section

Code:

```
83 def garden_set():
84     ..."""Function to create and use a set."""
85     ...# Create a set of flowers.
86     ...flower_set = {"rose", "dahlia", "scotch broom", "daffodil"}
87
171 ...# Create and use a set.
172 ...flower_set, custom_set_function = garden_set()
173 ...print("\nFinal flower set:", flower_set)
174 ...print("Minimum flower in set:", custom_set_function("min"))
175
```

Output:

```
Minimum flower in set: daffodil
```

3. Use both Dictionary and Sets as arguments to functions that you create, show how they behave similar and different from each other.

The fruit tree could not be included in the Garden Inventory since it is classified as a Tuple. The "Pear" could not be added to the Set since Tuples are immutable and not changeable. Lists are changeable.

Code:

```
102 def dict_set_in_functions(garden_dict, flower_set):
103     """Function to use both dictionary and set as argument to functions."""
104     # Behavior with dictionaries:
105     print("Behavior with dictionaries:")
106     print(f"Garden dictionary before the function: {garden_dict}")
107     add_data(garden_dict, "fruit tree", "pear")
108     print(f"Garden dictionary after adding -Pear- to the fruit tree dictionary: {garden_dict}")
109
110     # Behavior with sets:
111     print("\nBehavior with sets:")
112     print(f"Flower set before the function: {flower_set}")
113     flower_set.add("borage")
114     print(f"Flower set after adding -Borage-: {flower_set}")
115
176     # Use both dictionary and set as arguments to functions.
177     dict_set_in_functions(garden_dict, flower_set)
178
```

Output:

```
Behavior with dictionaries:
Garden dictionary before the function: {'flowers': ['rose', 'dahlia', 'peony', 'daffodil', 'sunflower'], 'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['plum', 'cherry']}
Category 'fruit tree' does not exist in the garden dictionary.
Garden dictionary after adding -Pear- to the fruit tree dictionary: {'flowers': ['rose', 'dahlia', 'peony', 'daffodil', 'sunflower'], 'vegetables': ['tomato', 'zucchini', 'carrot'], 'fruits': ['plum', 'cherry']}
```

```
Behavior with sets:
Flower set before the function: {'daffodil', 'sunflower', 'rose', 'dahlia'}
Flower set after adding -Borage-: {'daffodil', 'sunflower', 'rose', 'borage', 'dahlia'}
```

4. Show a simple Dictionary and a simple Set comprehension. (zip up two different lists, maybe). Attempt to show more complicated Dictionary and Set comprehensions.

In this section I have merged the set information into a comprehensive inventory of packets of seeds or plants and trees for the community garden. This inventory includes the plant name, season of planting, and quantity.

Code:

```
116 # Source: https://www.w3schools.com/python/python_sets.asp
117 def set_comprehensions():
118     """Function to show simple dictionary and set comprehensions."""
119     # Define sets representing the plant information
120     flowers = [("rose", "summer", 5), ("dahlia", "summer", 10), ("scotch broom", "spring", 5), ("daffodil", "spring", 50)]
121     garden = [{"tomato", "summer", 5}, {"zucchini", "summer", 1}, {"carrot", "summer", 5}, {"green bean", "summer", 3}, {"pumpkin", "fall", 2}, {"squash", "fall", 3}]
122     fruit_trees = [{"apple", "fall", 3}, {"plum", "summer", 2}, {"cherry", "summer", 1}]
123
124     def merge_sets(*sets):
125         """Simple dictionary comprehension."""
126         merged_set = {}
127         for x in sets:
128             for plant, season, quantity in x:
129                 if plant not in merged_set:
130                     merged_set[plant] = {"season": season, "quantity": quantity}
131                 else:
132                     merged_set[plant]["quantity"] += quantity
133         return merged_set
134
```

```

135     ...# Call the merge_sets function with the defined sets
136     ...merged_information = merge_sets(flowers, garden, fruit_trees)
137
138     ...# Print the merged information
139     ...print(f"\nInventory of seeds or plants and trees for the community garden is:")
140     ...for plant, details in merged_information.items():
141     ...|...print(f"{plant}: Season: {details['season']} - Quantity: {details['quantity']}")
142

```

Output:

```

Inventory of seeds or plants and trees for the community garden is:
rose: Season: summer - Quantity: 5
dahlia: Season: summer - Quantity: 10
scotch broom: Season: spring - Quantity: 5
daffodil: Season: spring - Quantity: 50
tomato: Season: summer - Quantity: 5
zucchini: Season: summer - Quantity: 1
carrot: Season: summer - Quantity: 5
green bean: Season: summer - Quantity: 3
pumpkin: Season: fall - Quantity: 2
squash: Season: fall - Quantity: 3
apple: Season: fall - Quantity: 3
plum: Season: summer - Quantity: 2
cherry: Season: summer - Quantity: 1

```