Assignment 12 – Dictionaries

Dictionary

A dictionary is similar to a list that stores a collection of items. While a list stores ordered items, a dictionary stores unordered items using keys. These items in the dictionary are known as “key”, “value” pairs.

Creating a Dictionary

1. Create a python file
2. Code the following:

dictionary.py

```
#! /usr/bin/env python3

# strings as keys and values
phones = {"AND": "Android",
          "BB": "Black Berry",
          "IOS": "iPhone",
          "WIN": "Windows"}
print("Strings as Keys and Values")
print(phones)
print()

# numbers as keys and strings as values
print("Numbers as Keys and Strings as Values")
numbers = {5: "Five", 6: "Six", 7: "Seven"}
print(numbers)
print()

# strings as keys, values as mixed types
print("Strings as Keys and Mixed Types as Values")
movie = {"name": "The Princess Bride",
         "year": 1987,
         "price": 9.99}
print(movie)
print()
```

3. Run and capture the following
   Screen Capture #1 (1 points)

```
dictionary
C:\Users\Saddleback\PycharmProjects\2018_Spring\Assignment_12\venv\Scripts\
Strings as Keys and Values
{'AND': 'Android', 'BB': 'Black Berry', 'IOS': 'iPhone', 'WIN': 'Windows'}

Numbers as Keys and Strings as Values
{5: 'Five', 6: 'Six', 7: 'Seven'}

Strings as Keys and Mixed Types as Values
{'name': 'The Princess Bride', 'year': 1987, 'price': 9.99}


Process finished with exit code 0
```

Accessing a Dictionary by key

4. Code the following:

```
24      print()
25
26      # Accessing an item by key
27      print("Accessing an item by Key")
28      phone = phones["IOS"]
29      print(phone)
30      print()
31
32      # Updating an item in the dictionary
33      print("Updating an item in the dictionary")
34      phones["IOS"] = "iPhone X"
35      phone = phones["IOS"]
36      print(phone)
37      print()
```

5. And test

```
{'name': 'The Princess Bride', 'year':

Accessing an item by Key
iPhone

Updating an item in the dictionary
iPhone X


Process finished with exit code 0
```

Validating Key in Dictionary

6. Code the following:

```
37      print()
38
39      # Testing for a key in the dictionary
40      print("Testing for a key in the dictionary")
41      phone = "WIX"
42      if phone in phones:
43          print(phones[phone])
44      else:
45          print(phone, "is not a valid phone key.")
46
47      print()
```

7. And test

```
iPhone X

Testing for a key in the dictionary
WIX is not a valid phone key.


Process finished with exit code 0
```

Dictionary get() Method

8. Code the following:

```
47      print()
48
49      # Dictionary get method
50      print("Using the dictionary get method")
51      phone = phones.get("AND")
52      print(phone)
53      phone = phones.get("IOX")
54      print(phone)
55      phone = phones.get("IOX", "Unknown")
56      print(phone)
57      print()
58
```

9. Run and capture the following:
   Screen Capture #2 (2 points)

```
dictionary
Accessing an item by Key
iPhone

Updating an item in the dictionary
iPhone X

Testing for a key in the dictionary
WIX is not a valid phone key.

Using the dictionary get method
Android
None
Unknown


Process finished with exit code 0
```

Deleting Dictionary Items

10. Code the following:

```
56    print(phone)
57    print()
58
59    # Deleting a dictionary item
60    print("Deleting a dictionary item")
61    print("Starting Count", str(len(phones)))
62    del phones["WIN"]  # This uses square brackets
63    print("Count after delete", str(len(phones)))
64    phone = phones.pop("BB")
65    print(phone, "was deleted. Count after delete", str(len(phones)))
66    phones.clear()
67    print("Count after clear", str(len(phones)))
68    print()
69
```

11. Run and capture the following:

Screen Capture #3 (1 point)

```
Unknown

Deleting a dictionary item
Starting Count 4
Count after delete 3
Black Berry was deleted. Count after delete 2
Count after clear 0


Process finished with exit code 0
```

Looping Through Dictionary Items – Key

12. Code the following:

```
68      print()
69
70      # Reload the dictionary
71      phones = {"AND": "Android",
72                "BB": "Black Berry",
73                "IOS": "iPhone",
74                "WIN": "Windows"}
75
76      # Loop through all items by key
77      print("Loop through all items by key")
78      for code in phones.keys():
79          print(code, phones[code])
80
81      print()
82
83      # Alternate loop through all items by key
84      print("Alternate loop through all items by key")
85      for code in phones:
86          print(code, phones[code])
87
88      print()
89
```

13. And test

```
Count after clear 0

Loop through all items by key
AND Android
BB Black Berry
IOS iPhone
WIN Windows

Alternate loop through all items by key
AND Android
BB Black Berry
IOS iPhone
WIN Windows


Process finished with exit code 0
```

14. Code the following:

```
88  print()
89
90  # Loop through all items by item
91  print("Loop through all items by item")
92  for code, phone in phones.items():
93      print(code, phone)
94
```

15. And test

```
WIN Windows

Loop through all items by item
AND Android
BB Black Berry
IOS iPhone
WIN Windows


Process finished with exit code 0
```

Looping Through Dictionary Items - Values

16. Code the following:

```
94
95  print()
96
97  # Loop through all values by value
98  print("Loop through all values by value")
99  for phone in phones.values():
100     print(phone)
101
102 print()
103
```

17. Run and capture the following:

Screen Capture #4 (2 points)

dictionary

```
Loop through all items by key
AND Android
BB Black Berry
IOS iPhone
WIN Windows

Alternate loop through all items by key
AND Android
BB Black Berry
IOS iPhone
WIN Windows

Loop through all items by item
AND Android
BB Black Berry
IOS iPhone
WIN Windows

Loop through all values by value
Android
Black Berry
iPhone
Windows


Process finished with exit code 0
```

Convert Dictionary to List

18. Code the following:

```
100         print(phone)
101
102     print()
103
104     # Convert dictionary to list and sort
105     print("Convert dictionary to list and sort")
106     codes = list(phones.keys())
107     codes.sort()
108     for code in codes:
109         print(code, phones[code])   # Square Bracket
110
111     print()
```

19. And test

```
Windows

Convert dictionary to list and sort
AND Android
BB Black Berry
IOS iPhone
WIN Windows


Process finished with exit code 0
```

Convert Two-Dimensional List to Dictionary

20. Code the following:

```
112     print()
113
114     # Create dictionary from two-dimensional array
115     phones = [["BB", "Black Berry"],
116               ["WIN", "Windows"],
117               ["IOS", "iPhone"],
118               ["AND", "Android"]]
119
120     print("Convert two-dimensional list to dictionary")
121     # Convert two-dimensional list to dictionary
122     phones = dict(phones)
123     print(phones)
124     print()
```

21. Run and capture the following:
    Screen Capture #5 (2 points)

```
dictionary
Convert dictionary to list and sort
AND Android
BB Black Berry
IOS iPhone
WIN Windows

Convert two-dimensional list to dictionary
{'BB': 'Black Berry', 'WIN': 'Windows', 'IOS': 'iPhone', 'AND': 'Android'}

Process finished with exit code 0
```

The State Program

Create an application that accepts four commands (*view, add, delete, and exit*). The application should allow you to view a list of states by two-character state code and state name. You should also be able to add states to the list as well as delete by entering the two-character state code.

22. Create **state.py** and code the following:

Code to Display Menu

```
#!/usr/bin/env python3


def display_menu():
    print("COMMAND MENU")
    print("view - View State")
    print("add  - Add a State")
    print("del  - Delete a State")
    print("exit - Exit program")
    print()

```

Code to Display State Codes

```

def display_codes(states):
    codes = list(states.keys())
    codes.sort()
    codes_line = "State codes: "
    for code in codes:
        codes_line += code + " "
    print(codes_line)

```

Code to Display States

```
21
22  def view(states):
23      display_codes(states)
24      code = input("Enter state code: ")
25      code = code.upper()
26      if code in states:
27          name = states[code]
28          print("State name: " + name + ".\n")
29      else:
30          print("There is no state with that code.\n")
31
```

Code to Add States

```
32
33  def add(states):
34      code = input("Enter state code: ")
35      code = code.upper()
36      if code in states:
37          name = states[code]
38          print(name + " is already using this code.\n")
39      else:
40          name = input("Enter state name: ")
41          name = name.title()
42          states[code] = name
43          print(name + " was added.\n")
44
```

Code to Delete States

```
45
46  def delete(states):
47      code = input("Enter state code: ")
48      code = code.upper()
49      if code in states:
50          name = states.pop(code)
51          print(name + " was deleted.\n")
52      else:
53          print("There is no state with that code.\n")
54
```

Code to Control the Flow of the Application

```
55
56  def main():
57      states = {"CA": "California",
58                "AZ": "Arizona"}
59
60      display_menu()
61      while True:
62          command = input("Command: ")
63          command = command.lower()
64          if command == "view":
65              view(states)
66          elif command == "add":
67              add(states)
68          elif command == "del":
69              delete(states)
70          elif command == "exit":
71              print("Bye!")
72              break
73          else:
74              print("Not a valid command. Please try again.\n")
75
76
77  if __name__ == "__main__":
78      main()
79
```

Display State Codes and State Names

Screen Capture #6 (1 point)

```
C:\Users\Saddleback\venv\Scri
COMMAND MENU
view - View State
add  - Add a State
del  - Delete a State
exit - Exit program

Command: view
State codes: AZ CA
Enter state code: ca
State name: California.

Command:
```

Add States

Screen Capture #7 (1 point)

```
C:\Users\Saddleback\venv\Scri
COMMAND MENU
view - View State
add  - Add a State
del  - Delete a State
exit - Exit program

Command: view
State codes: AZ CA
Enter state code: ca
State name: California.

Command: add
Enter state code: NV
Enter state name: Nevada
Nevada was added.

Command: view
State codes: AZ CA NV
Enter state code:
```

Delete States

Screen Capture #8 (1 point)

Code Validation - states.py (2 points)

```
C:\Users\Saddleback\venv\Scripts\pyth
COMMAND MENU
view - View State
add  - Add a State
del  - Delete a State
exit - Exit program

Command: view
State codes: AZ CA
Enter state code: az
State name: Arizona.

Command: del
Enter state code: az
Arizona was deleted.

Command: view
State codes: CA
Enter state code: ca
State name: California.

Command: exit
Bye!

Process finished with exit code 0
```

Complex Dictionary Objects

23. Code the following:

```
123     print(phones)
124
125     # Create complex dictionary objects
126     phones = {"iPhone": ["6", "6s", "SE", "7", "8", "X"],
127               "Android": ["Lollipop", "Marshmallow", "Nougat", "Oreo"],
128               "Windows": ["7", "7.5", "7.8", "8", "8.1", "10"],
129               "Blackberry": ["1.0", "3.6", "5.0", "6.0", "7.0", "7.1"]}
130
131     print("Creating complex dictionary objects")
132     phones = dict(phones)
133     print(phones)
134
```

24. Run and capture the following:

Screen Capture #9 (1 point)

```
dictionary ×

Creating complex dictionary objects
{'iPhone': ['6', '6s', 'SE', '7', '8', 'X'], 'Android': ['Lollipop', 'Marshmallow', 'Nou

Process finished with exit code 0
```

The Movie Catalog Program

Create an application that accepts five commands (*show, add, edit, delete, and exit*). The application should allow you to lookup a movie by title, add movies, edit movies, and delete movies from a catalog.

22. Create **movie_catalog.py** and code the following:

Code Validation – movie_catalog.py (2 points)

Code to Display Menu

```
#!/usr/bin/env python3


# display the full menu list
def display_menu():
    print("The Movie Catalog program")
    print()
    print("Menu")
    print("show - Show Movie Info")
    print("add  - Add Movie")
    print("edit - Edit Movie Info")
    print("del  - Delete Movie")
    print("exit - Exit Program")

```

Code to Display an Individual Movie

```

# show an individual movie
def show_movie(movie_catalog):
    title = input("Enter the title for the movie: ")
    if title in movie_catalog:
        movie = movie_catalog[title]
        print("Title:       ", title)
        print("Lead Actor:  ", movie["actor"])
        print("Release Year:", movie["year"])
    else:
        print("Sorry,", title, "does not exist in the catalog")

```

Code to Add or Edit a Movie

```
28  # add or edit a movie in the catalog
29  def add_edit_movie(movie_catalog, mode):
30      title = input("Enter the title for the movie: ")
31      if mode == "add" and title in movie_catalog:
32          print(title, "already exists in the catalog.")
33          response = input("Would you like to edit it? (y/n): ").lower()
34          if response != "y":
35              return
36      elif mode == "edit" and title not in movie_catalog:
37          print(title, "does not exist in the catalog.")
38          response = input("Would you like to add it? (y/n): ").lower()
39          if response != "y":
40              return
41
42      # get movie data and create a dictionary for the data
43      actor = input("Enter the actor name: ")
44      year = input("Enter release year: ")
45      movie = {"actor": actor, "year": year}
46
47      # add the movie data to the catalog using title as key
48      movie_catalog[title] = movie
49
```

Code to Delete a Movie

```
51  # delete a movie from catalog
52  def delete_movie(movie_catalog):
53      title = input("Enter the title for the movie: ")
54      if title in movie_catalog:
55          del movie_catalog[title]
56          print(title, "removed from the catalog")
57      else:
58          print(title, "does not exist in the movie catalog")
59
```

Code to Control the Flow of the Application

```
61  # main function which controls the program flow
62  def main():
63      movie_catalog = {
64          "Big":
65              {"actor": "Tom Hanks",
66               "year": "1988"},
67          "Toy Story":
68              {"actor": "Tim Allen",
69               "year": "1994"},
70          "Man on Fire":
71              {"actor": "Denzel Washington",
72               "year": "2004"}
73      }
74
75      display_menu()
76
77      while True:
78          print()
79          command = input("Command: ").lower()
80          if command == "show":
81              show_movie(movie_catalog)
82          elif command == "add":
83              add_edit_movie(movie_catalog, mode="add")
84          elif command == "edit":
85              add_edit_movie(movie_catalog, mode="edit")
86          elif command == "del":
87              delete_movie(movie_catalog)
88          elif command == "exit":
89              print("That's all folks!")
90              break
91          else:
92              print("Unknown command. Please try again.")
```

```
94
95  if __name__ == "__main__":
96      main()
97
```

Display a Movie by Title

Screen Capture #10 (1 point)

```
C:\Users\Saddleback\venv\Scripts\python
The Movie Catalog program

Menu
show - Show Movie Info
add  - Add Movie
edit - Edit Movie Info
del  - Delete Movie
exit - Exit Program

Command: show
Enter the title for the movie: Big
Title:        Big
Lead Actor:   Tom Hanks
Release Year: 1988

Command:
```

Add Movie

Screen Capture #11 (1 point)

```
C:\Users\Saddleback\venv\Scripts\python.exe "C:/Us
The Movie Catalog program

Menu
show - Show Movie Info
add  - Add Movie
edit - Edit Movie Info
del  - Delete Movie
exit - Exit Program

Command: add
Enter the title for the movie: Cast Away
Enter the actor name: Tom Hanks
Enter release year: 2000

Command: show
Enter the title for the movie: Cast Away
Title:        Cast Away
Lead Actor:   Tom Hanks
Release Year: 2000

Command:
```

Edit Movie

Screen Capture #12 (1 point)

```
C:\Users\Saddleback\venv\Scripts\python.exe "C:
The Movie Catalog program

Menu
show - Show Movie Info
add  - Add Movie
edit - Edit Movie Info
del  - Delete Movie
exit - Exit Program

Command: show
Enter the title for the movie: Toy Story
Title:        Toy Story
Lead Actor:   Tim Allen
Release Year: 1994

Command: edit
Enter the title for the movie: Toy Story
Enter the actor name: Tom Hanks
Enter release year: 1995

Command: Show
Enter the title for the movie: Toy Story
Title:        Toy Story
Lead Actor:   Tom Hanks
Release Year: 1995

Command:
```

Delete Movie

Screen Capture #13 (1 point)

```
C:\Users\Saddleback\venv\Scripts\python.exe
The Movie Catalog program

Menu
show - Show Movie Info
add  - Add Movie
edit - Edit Movie Info
del  - Delete Movie
exit - Exit Program

Command: show
Enter the title for the movie: Big
Title:        Big
Lead Actor:   Tom Hanks
Release Year: 1988

Command: del
Enter the title for the movie: Big
Big removed from the catalog

Command: show
Enter the title for the movie: Big
Sorry, Big does not exist in the catalog

Command:
```

Extra Credit

To get full points for each extra credit, you must include screen captures of the running output as well as the python (.py) code files. Extra credit is only available for assignments submitted on time.

Extra Credit #1 – Bird Counter (+1 Extra Credit)

Create a program for birdwatchers that stores a list of birds along with a count of the number of times each bird has been spotted.

```
Bird Counter program

Enter 'x' to exit

Enter name of bird: red-tailed hawk
Enter name of bird: killdeer
Enter name of bird: snowy plover
Enter name of bird: western gull
Enter name of bird: killdeer
Enter name of bird: western gull
Enter name of bird: x

Name            Count
=============== ===============
Killdeer        2
Red-Tailed Hawk 1
Snowy Plover    1
Western Gull    2
```

Specifications:

- Use a dictionary to store the list of sighted birds and the count of the number of times each bird was sighted.
- Use the pickle module to read the dictionary from a file when the program starts and to write the dictionary to a file when the program ends. That way, the data that's entered by the user isn't lost.

Extra Credit #2 – Champion Counter (+1 Extra Credit)

Create a program that reads a text file that contains a list of FIFA World Cup champions and determines the country that has won the most championships.

```
FIFA World Cup Winners

Country          Wins  Years
=======          ====  =====
Argentina        2     1978, 1986
Brazil           5     1958, 1962, 1970, 1994, 2002
England          1     1966
France           1     1998
Germany          4     1954, 1974, 1990, 2014
Italy            4     1934, 1938, 1982, 2006
Spain            1     2010
Uruguay          2     1930, 1950
```

Specifications:

- You can download the world_cup_champions.txt from canvas. It will contain data like this:
  Year,Country,Coach,Captain
  1930,Uruguay,Alberto Suppici,Jose Nasazzi
  1934,Italy,Vittorio Pozzo,Gianpiero Combi
  1938,Italy,Vittorio Pozzo,Giuseppe Meazza
  …
  …
  2002,Brazil,Luiz Felipe Scolari,Cafu
  2006,Italy,Marcello Lippi,Fabio Cannavaro
  2010,Spain,Vicente del Bosque,Iker Casillas
  2014,Germany,Joachim Low,Philipp Lahm
  2018,France,Didier Deschamps,Hugo Lloris
- When the program starts, it should read the text file and use a dictionary to store the required data using the name of each country that has won the World Cup as the key.
- The program should compile the data shown above and display the countries alphabetically.