Assignmnet No: 02

Ahmad waqar

MTN-22-13840

4-11.

My Pizzas, Your Pizzas: Start with your program from Exercise 4-1 (page 60). Make a copy of the list of pizzas, and call it friend_pizzas. Then, do the following: • Add a new pizza to the original list. • Add a different pizza to the list friend_pizzas. • Prove that you have two separate lists. Print the message, My favorite pizzas are:, and then use a for loop to print the first list. Print the message, My friend's favorite pizzas are:, and then use a for loop to print the second list. Make sure each new pizza is stored in the appropriate list.

```
pizza_ahmadwaqar=['Chicago Pizza','Siclian Pizza','Greek
Pizza','DeepDisha Pizza']
# Copying List to an other List
friend_pizza_ahmadwaqar=pizza_ ahmadwaqar [:]

#Inserting new item in original List
pizza_ ahmadwaqar.append('Fajeeta')
#inserting mew item to firends Pizza list
friend_pizza_
ahmadwaqar.append('BoneFire')

print("My Favorite pizzas are:")
for pizzas in pizza_
    ahmadwaqar:
    print(pizzas)

print("\nMy friend's Favorite pizzas are:")
   for friend_pizzas in friend_pizza_ ahmadwaqar:
                              print(friend_pizzas)
```

```
My Favorite pizzas are:
Chicago Pizza
Siclian Pizza
Greek Pizza
Deep Disha Pizza
Fajeeta

My friend's Favorite pizzas are:
Chicago Pizza
Siclian Pizza
```

```
Greek Pizza
Deep Disha Pizza
BoneFire
```

4-13.

Buffet: A buffet-style restaurant offers only five basic foods. Think of five simple foods, and store them in a tuple. • Use a for loop to print each food the restaurant offers. • Try to modify one of the items, and make sure that Python rejects the change. • The restaurant changes its menu, replacing two of the items with different foods. Add a block of code that rewrites the tuple, and then use a for loop to print each of the items on the revised menu.

```
buffet_ ahmadwaqar =('Mutton','Chicket','rice','kabab','fish')
print("Our Resturant Offers following Foods:")
# Part 1
for foods in buffet_kashif:
    print(foods)
#part 2
buffet_ ahmadwaqar [0]='Tikka Boti'
```

```
Our Resturant Offers following Foods:
Mutton
Chicket
rice
kabab
fish
```

```
---------------------------------------------------------------------------
------
TypeError                                 Traceback (most recent call
last)
e:\Assignments\Assignment_02.ipynb Cell 5 in <cell line: 7>()
      <a
href='vscode-notebook-cell:/e%3A/Assignments/Assignment_02.ipynb#W4sZm
lsZQ%3D%3D?line=4'>5</a>     print(foods)
      <a
href='vscode-notebook-cell:/e%3A/Assignments/Assignment_02.ipynb#W4sZm
lsZQ%3D%3D?line=5'>6</a> #part 2
.....> <a
href='vscode-notebook-cell:/e%3A/Assignments/Assignment_02.ipynb#W4sZm
lsZQ%3D%3D?line=6'>7</a> buffet_kashif[0]='Tikka Boti'

TypeError: 'tuple' object does not support item assignment
```

```
# Part 3 buffet_ ahmadwaqar
=('Mutton','Chicket','rice','kabab','fish','Tikka Boti')print("Our
Resturant Offers following Foods after adding new items:")for foods
in buffet_ ahmadwaqar:
    print(foods)
```

```
Our Resturant Offers following Foods after adding new items:
Mutton
Chicket
rice
kabab
fish
Tikka Boti
```

5-2.

More Conditional Tests: You don't have to limit the number of tests you create to 10. If you want to try more comparisons, write more tests and add them to conditional_tests.py. Have at least one True and one False result for each of the following: • Tests for equality and inequality with strings • Tests using the lower() function • Numerical tests involving equality and inequality, greater than and less than, greater than or equal to, and less than or equal to • Tests using the and keyword and the or keyword • Test whether an item is in a list • Test whether an item is not in a list

```
# Part 1 cell_ ahmadwaqar
="Samsung" print(cell_
ahmadwaqar =='Nokia')
print(cell_ ahmadwaqar
=='Samsung')print(cell_
ahmadwaqar!='Nokia')
print(cell_
ahmadwaqar!='Samsung')

# Part 2 Test using the Lower function
print("\n") print(cell_
ahmadwaqar.lower()=='Samsung')
print(cell_
ahmadwaqar.lower()=='samsung')

# Part 3
print("\n")
a_
ahmadwaqar
=15
print(a_ ahmadwaqar ==10) # Equality print(a_
ahmadwaqar!=10) # Non Equality print(a_
ahmadwaqar >15) # Greater then print(a_
ahmadwaqar <15) # Less then print(a_kashif>=10)
#Greater then equal toprint(a_ ahmadwaqar <=15)
# less then equal to

#Part 4
print("\n") age_
ahmadwaqar =15
height_
ahmadwaqar =5.7
print(age_ ahmadwaqar >18 and height_ ahmadwaqar
>=5.6)print(age_ ahmadwaqar >18 or height_
ahmadwaqar >=5.6)
```

Part 5

```
print("\n") list_ ahmadwaqar
=[30,25,56,580,582,100]item_ ahmadwaqar
=990
for item in list_ ahmadwaqar:
    if(item_ ahmadwaqar
        ==item):item_
        ahmadwaqar =True
if(item_ ahmadwaqar ==True):
    print("Item is Available in the list")
else:
    print("Item is Not available in the list")
```

```
False
True
True
False


False
True


False
True
False
False
True
True


False
True


Item is Not available in the list
```

6-3.

Glossary: A Python dictionary can be used to model an actual dictionary. However, to avoid confusion, let's call it a glossary. • Think of five programming words you've learned about in the previous chapters. Use these words as the keys in your glossary, and store their meanings as values. • Print each word and its meaning as neatly formatted output. You might print the word followed by a colon and then its meaning, or print the word on one line and then print its meaning indented on a second line. Use the newline character (\n) to insert a blank line between each word-meaning pair in your output. Looping

```
glossary_ ahmadwaqar ={
    'Variable':'Named memory locations',
    'List':'data structure to store group of items',
```

```
    'Python':'A high level Programing Language',
    'if':'conditional Statement',
    'Loops':'repaetidly executes stament or block of statements'
}
for key,value in glossary_ ahmadwaqar.items():
    print(key+":\t"+value)
```

```
Variable:  Named memory locations
List: data structure to store group of items
Python:    A high level Programing Language
if:  conditional Statement
Loops:     repaetidly executes stament or block of statements
```

6-11.

Cities: Make a dictionary called cities. Use the names of three cities as keys in your dictionary. Create a dictionary of information about each city and include the country that the city is in, its approximate population, and one fact about that city. The keys for each city's dictionary should be something like country, population, and fact. Print the name of each city and all of the information you have stored about it.

```
cities_
    ahmadwaqar
    ={
    'Lahore':{
        'Country':'Pakistan',
        'Population':'11.13 Million',
        'Fact':'Biggest city of the region'
    },
    'New York':{
        'Country':'United States',
        'Population':'8.468 Million',
        'Fact':'About 22 percent of New York City\'s land is used for
public parks '
    },
    'Mumbai':{
        'Country':'India',
        'Population':'20.96 Million',
        'Fact':'Mumbai was earlier called as Bombay'
    }
}
for cities,info in cities_ ahmadwaqar.items():
    print("\nCity Name: "+cities)
    print("\tContry: "+info['Country'])
    print("\tPopulation: "+info['Population'])
    print("\tFact: "+info['Fact'])
```

```
City Name: Lahore
     Contry: Pakistan
     Population: 11.13 Million
```

```
      Fact: Biggest city of the region

City Name: New York
      Contry: United States
      Population: 8.468 Million
      Fact: About 22 percent of New York City's land is used for public
parks

City Name: Mumbai
      Contry: India
      Population: 20.96 Million
      Fact: Mumbai was earlier called as Bombay
```

6-12.

Extensions: We're now working with examples that are complex enough that they can be extended in any number of ways. Use one of the example programs from this chapter, and extend it by adding new keys and values, changing the context of the program or improving the formatting of the output.

```
favorite_languages_
    ahmadwaqar = {'jen':
    'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}
print("Follwing items are in the dictionary.")
for name,language in favorite_languages_kashif.items():
    print(name.title()+"'s favorite Language is: "+language.title())

# adding new keys and values in the dictionaryfavorite_languages_
ahmadwaqar ['Lara']='html' favorite_languages_ ahmadwaqar
['Smith']='php'
print("\nFollwing items are in the dictionary after adding new keys
and values.")
for name,language in favorite_languages_ ahmadwaqar.items():
    print(name.title()+"'s favorite Language is -->
"+language.upper())
```

```
Follwing items are in the dictionary.
Jen's favorite Language is: Python
Sarah's favorite Language is: C
Edward's favorite Language is: Ruby
Phil's favorite Language is: Python

Follwing items are in the dictionary after adding new keys and values.
Jen's favorite Language is --> PYTHON
Sarah's favorite Language is --> C
Edward's favorite Language is --> RUBY
```

```
Phil's favorite Language is --> PYTHON
Lara's favorite Language is --> HTML
Smith's favorite Language is --> PHP
```