CIS-298 Intro to Python

With Professor Robert Mann

HW #2

Student: Demetrius Johnson

19 January 2023

Due: 24 January 2023 at 4pm

Submit your code in a report: question number, code copy/pasted, snippet of output.

You may answer all questions in one program but only submit the code for that question, separately from any other code.

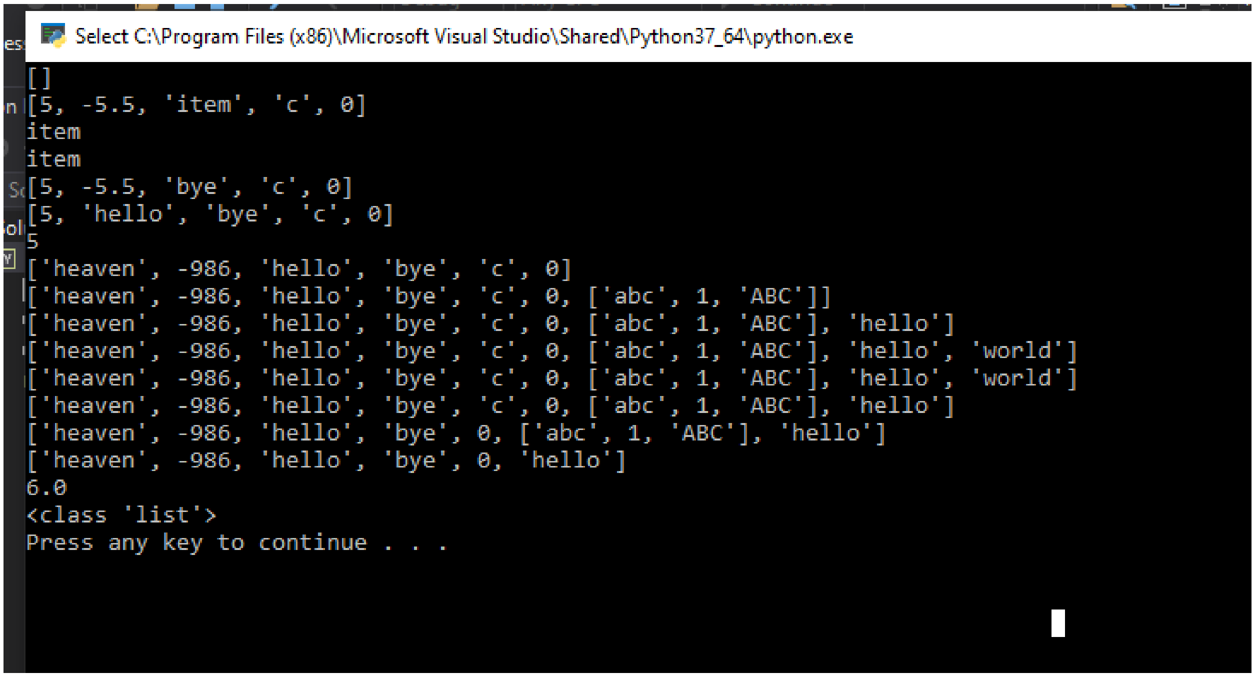
****** Lists has 3 questions, tuples has 3 questions, dictionary has 8 questions. ******

Question 1 - Lists

```
#* Lists []

#Create an empty list and print the list
my_list = []
print(my_list)

#Create a singleton list and print the list
my_list_1 = [1]

#Create a list of 5 items of mixed types and print the list
my_list_5 = [5, -5.5, 'item', 'c', 0]
print(my_list_5)

#Print the 3rd item in the list
print(my_list_5[2])

#Print the item at index -3
print(my_list_5[-3])

#Change the 3rd item in the list to “bye” and print the whole list
my_list_5[2] = "bye"
print(my_list_5)

#Change the -4th item in the list to ‘hello’ and print the whole list
my_list_5[-4] = "hello"
print(my_list_5)

#Print the length of the list
print(len(my_list_5))

#Find the min and max of the list
#print(min(my_list_5)) --> error, requires list to have elements all of the same type
#print(max(my_list_5)) --> error, requires list to have elements all of the same type

#Delete the -5th item in the list
my_list_5.pop(-5)

#Add list [‘heaven’, -986] to the beginning of your list
my_list_5 = ['heaven', -986] + my_list_5
print(my_list_5)

#Append list [‘abc’,1,”ABC” ] to the end of your list
my_list_5.append(['abc', 1, 'ABC'])
print(my_list_5)

#Add ‘hello’ to the end of your list.  What happened? → added ‘hello’ to end of list as
a new element
my_list_5.append('hello')
```

```
print(my_list_5)

#Append “world” to the end of your list. What happened? → added ‘world’ to end of list
as a new element
my_list_5.append("world")
print(my_list_5)

#Print your list, perform pop() on your list, and print the list again
print(my_list_5)
my_list_5.pop()
print(my_list_5)

#Perform pop(4) on your list and print the list
my_list_5.pop(4)
print(my_list_5)

#Perform pop(-2) on your list and print the list
my_list_5.pop(-2)
print(my_list_5)

#Print the length of your list as a float
print(float(len(my_list_5)))

#Print the type and ord of your list
print(type(my_list_5))
#print(ord(my_list_5)) error--> ord() function requires a single character
```

Screenshot Output


Select C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

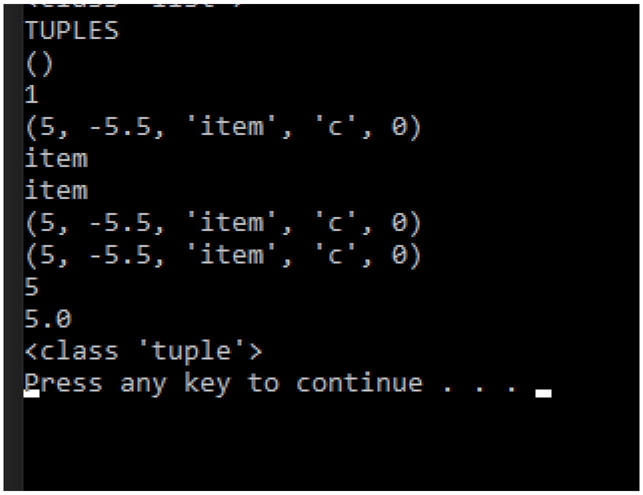
```
[]
[5, -5.5, 'item', 'c', 0]
item
item
[5, -5.5, 'bye', 'c', 0]
[5, 'hello', 'bye', 'c', 0]
5
['heaven', -986, 'hello', 'bye', 'c', 0]
['heaven', -986, 'hello', 'bye', 'c', 0, ['abc', 1, 'ABC']]
['heaven', -986, 'hello', 'bye', 'c', 0, ['abc', 1, 'ABC'], 'hello']
['heaven', -986, 'hello', 'bye', 'c', 0, ['abc', 1, 'ABC'], 'hello', 'world']
['heaven', -986, 'hello', 'bye', 'c', 0, ['abc', 1, 'ABC'], 'hello', 'world']
['heaven', -986, 'hello', 'bye', 'c', 0, ['abc', 1, 'ABC'], 'hello']
['heaven', -986, 'hello', 'bye', 0, ['abc', 1, 'ABC'], 'hello']
['heaven', -986, 'hello', 'bye', 0, 'hello']
6.0
<class 'list'>
Press any key to continue . . .
```

Question 2 – Tuples

```
#* Tuples ()
print("TUPLES")
#Create an empty tuple and print the tuple
empty_tuple = ()
print(empty_tuple)

#Create a singleton tuple and print the tuple
singleton_tuple = (1)
print(singleton_tuple)

#Create a tuple of 5 items of mixed types and print the tuple
my_tuple_5 = (5, -5.5, 'item', 'c', 0)
print(my_tuple_5)

#Print the 3rd item in the tuple
print(my_tuple_5[2])

#Print the item at index -3
print(my_tuple_5[-3])

#Change the 3rd item in the tuple to “bye” and print the whole tuple
# my_tuple_5[2]= 'bye' #ERROR: NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE
IMMUTABLE
print(my_tuple_5)

#Change the -4th item in the tuple to ‘hello’ and print the whole tuple
#my_tuple_5[2]= 'hello' #ERROR: NO FUNCTION TO REMOVE OR ADD ELEMENTS  --> TUPLES ARE
IMMUTABLE
print(my_tuple_5)

#Print the length of the tuple
print(len(my_tuple_5))

#Find the min and max of the tuple
#print(min(tmy_tuple_5)) ERROR: MIN AND MAX REQUIRES ALL ELEMENTS OF THE SAME TYPE
#print(max(tmy_tuple_5)) ERROR: MIN AND MAX REQUIRES ALL ELEMENTS OF THE SAME TYPE

#Delete the -5th item in the tuple
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Add tuple [‘heaven’, -986] to the beginning of your tuple
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Append tuple [‘abc’,1,”ABC” ] to the end of your tuple
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Add ‘hello’ to the end of your tuple.  What happened?
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Append “world” to the end of your tuple. What happened?
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Print your tuple, perform pop() on your tuple, and print the tuple again
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Perform pop(4) on your tuple and print the tuple
```

```
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Perform pop(-2) on your tuple and print the tuple
#NO FUNCTION TO REMOVE OR ADD ELEMENTS --> TUPLES ARE IMMUTABLE

#Print the length of your tuple as a float
print(float(len(my_tuple_5)))
#Print the type and ord of your tuple
print(type(my_tuple_5))
#print(ord(my_tuple_5)) error--> ord() function requires a single character
```

Screenshot Output

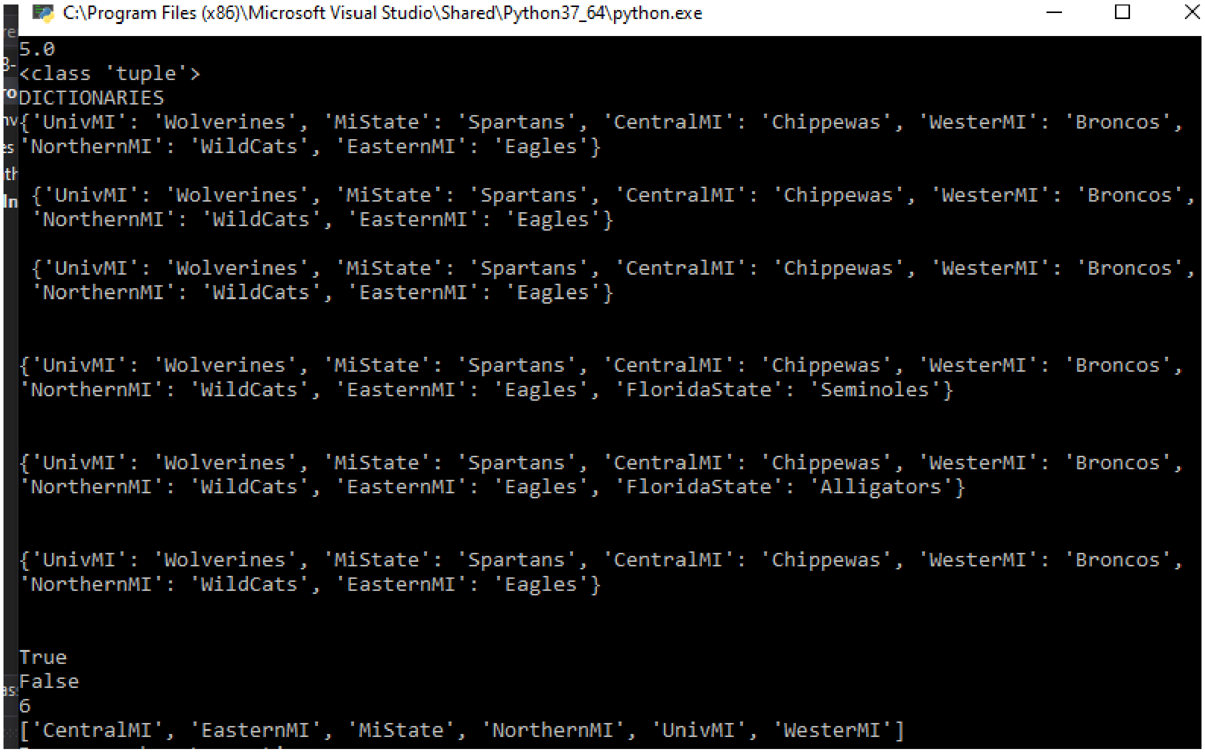


```
TUPLES
()
1
(5, -5.5, 'item', 'c', 0)
item
item
(5, -5.5, 'item', 'c', 0)
(5, -5.5, 'item', 'c', 0)
5
5.0
<class 'tuple'>
Press any key to continue . . .
```

Question 3 – Dictionaries

```
#* Dict { }

#Build a dictionary with 6 college & mascot associations, using all three methods

#                 A = { key : value, key : value }
Dictionary1 = {"UnivMI":"Wolverines",
               "MiState":"Spartans",
               "CentralMI":"Chippewas",
               "WesterMI":"Broncos",
               "NorthernMI":"WildCats",
               "EasternMI":"Eagles"
               }

#      A = dict( [(key, value), (key,value)] )
Dictionary2 = dict([("UnivMI","Wolverines"),
               ("MiState","Spartans"),
               ("CentralMI","Chippewas"),
               ("WesterMI","Broncos"),
               ("NorthernMI","WildCats"),
               ("EasternMI","Eagles")]
               )

#                 A=dict( key=value, key=value )
Dictionary3 = dict(UnivMI = "Wolverines",
               MiState = "Spartans",
               CentralMI = "Chippewas",
               WesterMI = "Broncos",
               NorthernMI = "WildCats",
               EasternMI = "Eagles"
               )
print(Dictionary1, Dictionary2, Dictionary3)
#Access the dictionary using a key that doesn’t exist
#print(Dictionary1["FloridaState"]) -->exception thrown --> key not in dictionary

#Add a new value to the dictionary and print the dictionary
Dictionary1["FloridaState"] = "Seminoles"
print(Dictionary1)

#Change the value of an existing entry and print the dictionary
Dictionary1["FloridaState"] = "Alligators"
print(Dictionary1)

#Del an entry from the dictionary
del Dictionary1["FloridaState"]
print(Dictionary1)

#Check to see if a value is ‘in’ the dictionary and also ‘not in’ the dictionary
print ('Wolverines' in Dictionary1.values())
print ('Wolverines' not in Dictionary1.values())

#Get the length of the dictionary
print(len(Dictionary1))

#Print a list of sorted keys to the dictionary
print(sorted(Dictionary1))
```

Screenshot Output


C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe

```
5.0
<class 'tuple'>
DICTIONARIES
{'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
'NorthernMI': 'WildCats', 'EasternMI': 'Eagles'}

 {'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
 'NorthernMI': 'WildCats', 'EasternMI': 'Eagles'}

 {'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
 'NorthernMI': 'WildCats', 'EasternMI': 'Eagles'}


{'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
'NorthernMI': 'WildCats', 'EasternMI': 'Eagles', 'FloridaState': 'Seminoles'}


{'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
'NorthernMI': 'WildCats', 'EasternMI': 'Eagles', 'FloridaState': 'Alligators'}


{'UnivMI': 'Wolverines', 'MiState': 'Spartans', 'CentralMI': 'Chippewas', 'WesterMI': 'Broncos',
'NorthernMI': 'WildCats', 'EasternMI': 'Eagles'}


True
False
6
['CentralMI', 'EasternMI', 'MiState', 'NorthernMI', 'UnivMI', 'WesterMI']
```