Code: CSA0810
Subject: python programming

Name: G.Jeevan
Reg.No: 192211714

ASSIGNMENT-II

PYTHON Collections:

There are 4 Data types:

1. list
2. Tuple
3. Set
4. Dictionary

List:

list is a collection which is ordered & changeable. Allows duplicate members.

Tuple:

Tuple is a collection which is ordered & unchangeable. Allows duplicate members.

Set:

Set is a collection which is unordered, unchangeable & unindexed. No duplicate members.

Dictionary:

Dictionary is a collection which is ordered & changeable. NO duplicate members.

→ Set items are unchangeable, but can add or remove elements.

→ python version 3.7, dictionaries are ordered. In python 3.6 & earlier, dictionaries are unordered.

→ the four data types can be distinguished by order, index, mutable, changable & allow duplicates.

→ collections in python are containers used for storing data and are commonly known as data structures.

→ python has built in collections module providing additional data structures for collection of data.

Representing 4 data types in tabular column:

Data structure	Definition	Mutable	Ordered	Indexable	Allows duplicates	Example
Tuples	ordered, immutable collection	NO	yes	yes	yes	Mytuple = (1, 2, 3)
Lists	ordered, mutable collection	yes	yes	yes	yes	Mylist = [1, 2, 3]
sets	unordered, mutable collection	yes	NO	NO	NO	Myset = {1, 2, 3}
Dictionaries	unordered, collection of key value pairs	yes	yes	NO	yes	Mydict = {'key1': 'value1', 'key2': 'value2'}

Examples of list:

Lists are ordered mutable sequences that can be changed after they have been created by adding, removing and changing objects.

```
prices = [20, 10, 40, 15]
```

→ Lists can be created by using square brackets.

→ A list is created & stored data in the form of list.

```
ages = [15, 20, 18, 16]
print(ages)
```

→ print will give or show the data and print it as output.

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)
```

→ first a list is created as fruits.

→ using the "append" function we can add the new element or data to the existed list.

→ the orange is added at the end the list.

output:

```
["apple", "banana", "cherry", "orange"]
```

Examples of Tuple :

Tuples are ordered immutable sequence that store multiple items.

```
fruits = ("apple","banana","cherry")
```

→ Tuples can be created by using '()' round brackets.

→ Single object tuples are referred to as singleton.

→ It can be created by using a trailing common after the object or else python identifies it as a string.

```
fruit = ("orange")
print(type(fruit))
print(fruit)
```

output :

```
<class 'str'>
orange.
```

→ commas are what make a tuple as parentheses are optical.

```
letters = 'a','b','b'
type(letters)
```

→ Here, we create a new tuple.

→ print the type of data as result.

Examples of sets:

set can be created by using curly brackets following the variable names (or) using set() constructor.

```
letters = {"a","b","c"}
letters = set("a","b","c")
```

→ set are used to store multiple items in a single variable.

creating a set

```
fruits = {"apple","banana","cherry"}
print(fruits)
```

→ Here we create a new set like fruits
→ print the result as output.

```
S1 = {"a","b","c"}
S2 = {1,2,3}
S3 = S1.union(S2)
print(S3)
```

→ S1 & S2 will be created as set.
→ By using union or '+' operator we can get the set 3.

output:

```
{'a','b','c',1,2,3}
```

Examples of Dictionaries:

The values associated with a key can be any object like sets, dictionaries are unordered.

```
ages = {"Mike": 10, "Leo": 19, "Den": 5}
print (ages)
```

output:

```
{'Mike': 10, 'Leo': 19, 'Den': 5}
```

→ values in dictionary items can be of any data types.

```
Mydict = {'key1': 'value1', 'key2': 'value2'}
print (mydict)
print (mydict['key1'])
Mydict['key 3'] = 'value 3'
print (mydict)
```

Dictionary items:

Dictionary items are presented in key: value pairs can be referred by using the key name.

```
thisdict = {"brand": "ford", "year": 1964}
print (thisdict ("brand"))
```

→ Here we can create the new dictionaries with each special & its name.

→ print represent brand which will print name of brand.

output:

"ford".