**ASSIGNMENT 4**

1. **What exactly is []?**

Ans=> [] is an empty list.

**2. In a list of values stored in a variable called spam, how would you assign the value ‘hello’; as the**

**third value? (Assume [2, 4, 6, 8, 10] are in spam.)**

Ans=> spam = [2,4,6,8,10]

spam.insert(3,’hello’)

now spam = [2,4,6,’hello’,8,10]

**## Let’s pretend the spam includes the list [‘a’, ‘b’, ‘c’, ‘d’] for the next three queries.**

1. **What is the value of spam[int(int(‘3’ \* 2) / 11)]?**

Ans=> Value will be ‘d’

1. **What is the value of spam[-1]?**

Ans=> Value will be ‘d’

1. **What is the value of spam[:2]?**

Ans=> [‘a’,’b’]

**##Let’s pretend bacon has the list [3.14, ‘cat’, 11, ‘cat’, True] for the next three questions.**

1. **What is the value of bacon.index(‘cat’)?**

Ans=> Value is 1

1. **How does bacon.append(99) change the look of the list value in bacon?**

Ans=>[3.14, 'cat', 11, 'cat', True, 99]

**8. How does bacon.remove(‘cat’) change the look of the list in bacon?**

Ans=> [3.14, 11, 'cat', True, 99]

**9. What are the list concatenation and list replication operators?**

Ans=> List concatenation operators are – ‘+’

List replication operators are – ‘\*’

**10. What is difference between the list methods append() and insert()?**

Ans=> Insert allows to add a specific element at a specific index of the list. Where as append can add the element only at the end of the list.

**11. What are the two methods for removing items from a list?**

Ans=> remove(), clear(),pop() etc are the inbuild methods for removing items from a list.

**12. Describe how list values and string values are identical.**

Ans=> Lists and Strings both are sequences.

**13. What’s the difference between tuples and lists?**

Ans=> Tuples – 1. Tuples are immutable.

1. Implications of iterations are much faster in tuples.
2. Elements can be accessed better.
3. Consumes less memory.
4. Does not have many built-in methods.
5. Unexpected errors and changes rarely occur in tuples.

Lists – 1. It is mutable

2. The implication of iterations is time-consuming in the list.

3. Operations like insertion and deletion are better performed.

4. Consumes more memory.

5. Many built-in methods are available.

6. Unexpected errors and changes can easily occur in lists.

**14. How do you type a tuple value that only contains the integer 42?**

Ans=> t=(42,)

**15. How do you get a list value’s tuple form? How do you get a tuple value’s list form?**

Ans=> List value’s tuple form 🡪 ([list1],[list2],[list3])

Tuple value’s list form 🡪 [(tuple1),(tuple2),(tuple3)]

**16. Variables that “contain” list values are not necessarily lists themselves. Instead, what do they contain?**

Ans=> Variables will contain **references to list values** rather than list values themselves. But for strings and integer values, variables simply contain the string or integer value.

**17. How do you distinguish between copy.copy() and copy.deepcopy()?**

Ans=> **copy()** create reference to original object. If you change copied object - you change the original object.

**deepcopy()** creates new object and does real copying of original object to new one. Changing new deepcopied object doesn't affect original object.