Lab3

7. Methods / Member functions that are applicable to lists but not tuples are following:

→ append() : Adds element at end of list
→ remove() : Removes given element from list (first occurrence)
→ pop() : Removes last element from list
→ insert() : Inserts element at index given
→ reverse() : Reverses list
→ sort() : Can sort list in ascending or descending order

The above methods are in place. They change the object. (Mutators)

Lists are mutable so they are allowed to have mutating methods

Tuples are immutable, so they do not have such methods.

11. A frozenset is similar to a set in python apart from the fact that the frozenset is immutable.

→ We create one as follows:

frozen_set = frozenset(iterable), any iterable like list, tuple, strings

→ Once you generate a frozenset they cannot be changed

s1 = frozenset("hello")
s2 = frozenset([1,2,2,3,4,3])

→ We can also create a set of frozensets

S = {frozenset{i, i+1, i+2} for i in range(10)}

Lab3

7. Methods / Member functions that are applicable to lists but not tuples are following:

→ append() : Adds element at end of list
→ remove(): Removes given element from list (First occurence)
→ pop() : Removes last element from list
→ insert() : Inserts element at index given
→ reverse(): Reverses list
→ sort() : Can sort list in ascending or descending order

The above methods are in place. They change the object. (Mutators).

Lists are mutable so they are allowed to have mutating methods.

Tuples are immutable, so they do not have such methods.

11. A frozenset is similar to a set in python apart from the fact that the frozenset is immutable.

→ We create one as follows:

frozen_Set = frozenset(iterable), any iterable like list, tuple, strings

→ Once you generate a frozenset they cannot be changed

s1 = frozenset("hello")
s2 = frozenset([1,2,2,3,4,3])

→ We can also create a set of frozensets

S = {frozenset{i, i+1, i+2} for i in range(10)}.