

## Practical 08

Aim : Write a python function that takes a list and return a new list with unique elements of the first list. For example, Sample List = [1,1,1,2,2,3,3,4]  
Unique List = [1,2,3,4]

### Theory:

In Python, both sets and lists are used to store collections of items, but they have some differences in terms of their characteristics and functionality.

A list is an ordered collection of items, which can be of any data type. You can add, remove, and access items in a list using index numbers. Lists are created using square brackets `[]` and the elements are separated by commas. For example:

```
my_list = [1, "hello", True, 3.14]
```

A set, on the other hand, is an unordered collection of unique items. Sets do not allow duplicates, and their elements can be of any data type, like lists. You can add and remove items from a set, but you cannot access individual elements using index numbers. Sets are created using curly braces `{}` or by using the `set()` function. For example:

```
my_set = {1, "hello", True, 3.14}
```

Some differences between sets and lists are:

1. Order: Lists are ordered, meaning that the elements in a list have a specific position or index. Sets, on the other hand, are unordered, so their elements have no specific order.
2. Duplicates: Lists allow duplicates, meaning you can have multiple instances of the same value in a list. Sets do not allow duplicates, meaning that you can only have one instance of each value in a set.
3. Access: You can access individual elements in a list using index numbers. Sets do not support indexing, so you cannot access individual elements directly. However, you can use iteration or membership testing to access or check if an element exists in a set.
4. Mutability: Lists are mutable, meaning you can change their values by adding, removing or modifying elements. Sets are also mutable, so you can add or remove elements from them.

In summary, lists are useful for storing ordered collections of data that allow duplicates, while sets are useful for storing unordered collections of unique data.