WEEK-3

1. Write a python program to explain about Membership operators.

Source code:

```
x = 'Hello world'
y = {1:'a', 2:'b'}
# check if 'H' is present in x string
print('H' in x)  # prints True
# check if 'hello' is present in x string
print('hello' not in x)  # prints True
# check if '1' key is present in y
print(1 in y)  # prints True
# check if 'a' key is present in y
print('a' in y)  # prints False
```

Output:

```
True
True
True
False
```

Description:

In Python, `in` and `not in` are the membership operators. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary).

Here, `'H'` is in `x` but `'hello'` is not present in `x` (remember, Python is case sensitive).

Similarly, `1` is key and `'a'` is the value in dictionary `y`. Hence, `'a' in y` returns `False`.

2. Write a python program to explain about Identity operators.

Source code:

```
x1 = 5
y1 = 5
x2 = 'Hello'
y2 = 'Hello'
x3 = [1,2,3]
y3 = [1,2,3]
print(x1 is not y1)  # prints False
print(x2 is y2)  # prints True
print(x3 is y3)  # prints False
```

Output:

False
True
False

Description:

In Python, `is` and `is not` are used to check if two values are located on the same part of the memory. Here, we see that `x1` and `y1` are integers of the same values, so they are equal as well as identical. Same is the case with `x2` and `y2` (strings).

But `x3` and `y3` are lists. They are equal but not identical. It is because the interpreter locates them separately in memory although they are equal.

For Loop

3. write a program to create, append and remove lists in python

Source code:

create a list:-

```
create a list in python.
list1 = ['computer', 'programming', 1957, 2070, 3242];
list2 = [1, 2, 3, 4, 5];
list3 = ["a", "b", "c", "d", "e"];
ex:
# Python Lists Example - Creating a list
programmy_list = ["zero", "one", "two",
"three"]; print("Elements of the list, my_list
are:");
for ml in my_list:
print(ml);
```

Output:

```
Elements of the list, my_list are:
zero
one
two
three
```

ii) Concatenating two lists in python

```
# Lists concatenation in python example
my_list = ["zero",
"one","two", "three", "four"]
my_new_list = ["five","six"]
my_list += my_new_list
print("List's items
after
concatenating:");
for l in my_list:
        print(l)
```

Output:

```
List's items after concatenating:
zero
one
two
three
four
five
six
```

---

iii)To delete any element from a list in python

```
# Deleting element from list in python
example my_list = ["zero", "one", "two",
"three", "four"];print("Elements of the list,
my_list are:");
for ml in my_list:
  print(ml);
index = input("\nEnter index
no:");index = int(index);
print("Deleting the element present at index
number",index);del my_list[index];
print("\nNow elements of the list, my_list
are:");for ml in my_list:
 print(ml);
```

Output:

```
Elements of the list, my_list are:
zero
one
two
three
four

Enter index no:3
Deleting the element present at index number 3

Now elements of the list, my_list are:
zero
one
two
four
```

Description:

Creating a list in Python involves defining a sequence of elements enclosed in square brackets

Appending to a list in Python involves adding one or more elements to the end of an existing list. This can be done using the append() method.

Removing elements from a list in Python can be done using several methods. Here are some common ones:

Remove(): removes the first occurrence of a given value from the list.

Pop(): removes and returns the last element of the list.