

## SCA ASSIGNMENT 1

### QUESTION :

**WHAT IS A LOOP, OPERATORS IN PYTHON, CONDITIONAL STATEMENTS IN PYTHON, TERNARY STATEMENTS?**

### ANSWER :

a) A loop is a statement that allows for execution of a statement or a group of statements multiple times under specific conditions. There are two main types of loops in Python which can also be nested.

i) **While loop** is used to iterate over a block of code repeatedly until a given condition returns false (the while loop can execute a set of statements as long as a condition is true.)

First the given condition is checked, if the condition returns false, the loop is terminated and the control jumps to the next statement in the program after the loop. If the condition returns true, the set of statements inside loop are executed and then the control jumps to the beginning of the loop for next iteration.

The syntax of a **while** loop in Python programming language is :

```
while expression:  
    statement(s)
```

### EXAMPLE :

```
x = 1  
while x < 6:  
    print(x)  
    x += 1  
else:  
    print("x is no longer less than 6")
```

### ii) For loop

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

The syntax of a **for** loop in Python programming language is :

```
for iterating_var in sequence:  
    statements(s)
```

```
for letter in 'Python':  
    print( 'Current Letter :', letter)
```

b) Operators are the constructs which can manipulate the value of operands. There are multiple types of Operators in Python which are:

- **Arithmetic operators :**  
Arithmetic operators are used with numeric values to perform common mathematical operation (+ *Addition*, - *Subtraction*, \* *Multiplication*, / *Division*, % *Modulus*, \*\* *Exponentiation*, // *Floor division*)
- **Assignment operators**  
Assignment operators are used to assign values to variables(=, +=, -=, \*=, /=, %=, //=, \*\*=, &=, |=, >>=, <<=)
- **Comparison operators**  
Comparison operators are used to compare two values(==Equal, != Not equal, > Greater than, < Less than, >= Greater than or equal to, <= Less than or equal to)
- **Logical operators**  
Logical operators are used to combine conditional statements(***and*** Returns True if both statements are true, ***or*** Returns True if one of the statements is true, ***not*** Reverse the result, returns False if the result is true)
- **Identity operators**  
Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location  
(***is*** Returns True if both variables are the same object, ***is not*** Returns True if both variables are not the same object)
- **Membership operators**  
Membership operators are used to test if a sequence is presented in an object(***in*** Returns True if a sequence with the specified value is present in the object, ***not in*** Returns True if a sequence with the specified value is not present in the object)
- **Bitwise operators**  
Bitwise operators are used to compare (binary) numbers  
(& AND, |OR, ^XOR, ~ NOT, <<Zero fill left shift, >>Signed right shift)

- c) Conditional statements are a set of rules performed if a certain condition is met or not.  
These constitute of *if*, *elif*, *if – else*

**If statement:**

```
x, y = 5, 6  
if x>y:  
    print("x")
```

**Elif statement:**

The elif keyword is executed if the previous conditions were not true

```
x, y = 5, 6  
if x>y:  
    print("x")  
  
elif x == y:  
    print("x and y are equal")
```

**Else:**

The else keyword catches anything which isn't caught by the preceding conditions.

```
x, y = 5, 6  
if x>y:  
    print("x")  
  
elif x == y:  
    print("x and y are equal")  
  
else:  
    print("y")
```

d) Ternary operators also known as conditional expressions are operators that evaluate something based on a condition being true or false. It simply allows to test a condition in a single line replacing the multiline if-else making the code compact.

#### **EXAMPLE :**

a. Using python if-else statement :

```
x, y = 5, 6
if x>y:
    print("x")
else:
    print("y")
```

b. Using ternary operator

```
x, y = 5, 6
print("x" if x> y else "y")
```