Session-6

Condition statements and loops:

Condition statements :- (>, <, ==, >=, <=, !=) o/p ⇒ True / false

- Variables should not be a keyword
- We use "if" keyword.

```
if (condition):
    ---- Operation to perform
indentation
else:
    ---- operation to perform
```

Ex:
```
num1 = 300
num2 = 200
if (num1 > num2):
    print("num1 is greater")
else:
    print("num2 is greater")
```
o/p ⇒ num1 is greater.

- if-elif-else

```
num1 = 200
num2 = 200
if (num1 > num2):
    print("num1 is greater")
elif (num2 > num1):
    print("num2 is greater")
else:
    print("Both are equal")
```
o/p ⇒ Both are equal.

nested if-elif-else

```
if (- - -):
  if(- - - -):
  elif(- - -):
  else():
elif(- - -):
  if(- - -):
else:
```

Rule based approach ⇓ Nearest Neighbour algorithm

How to know if an element is present in a list?

Method-1

```
a = [10,20,30,40,50]   element to search = 60
b = a.count(60)
if (b != 0):
    print("element exists")
else:
    print("element doesn't exists")
```
o/p ⇒ element doesn't exists.

```
a = [20,40,60,80,100]
search_element = 60
b = a.count(search_element)
if (b != 0):
    print("Element exists")
else:
    print("element doesn't exists")
```
o/p ⇒ Element exists.

Method-2 — in → Operator.

```
a = [10,20,30,40,50]
b = 60
print(b in a)
```
o/p ⇒ false

```
a = [10,20,30,40,50]
search_element = 60
if (search_element in a):
    print("Element exists")
else:
    print("Element doesn't exists")
```
o/p ⇒ Element doesn't exists.

if condition :

operation

Condition will be true when
⇒ non-zero element True
⇒ non-empty list True

Condition will be false if
[ ] false
None false
0 false

```
a = [10,20,30,40,50]
if (a.count(60)):
    print("Element exists")
else:
    print("Element doesn't exists")
```

o/p ⇒ Element doesn't exists

Given total number of marks and marks scored by a student. Calculate whether the student has passed or failed.

→ If the % of the marks scored by the student is less than 40 then the student needs to reappear in the exam.

Loops in Python :

- for loop
- while loop.

for loop :

for [user_defined_variable] in [iterable datatype]:

---- block of code

↓ indentation

Iterable datatype : Any datatype that supports indexing.

Ex :

```
fruits = ["apple","banana","cherry",
 "dragonfruit"]
for fruit in fruits :
    print(fruit)
```

user defined variable → fruit ← iterable list

o/p ⇒ apple
banana
cherry
dragonfruit

Length of a list -

```
fruits = ["apple", "banana", "cherry",
          "dragonfruit"]
for idx in range(0, len(fruits)):
    print(fruits[idx])
```

range(0, +, 1)

o/p ⇒ apple
banana
cherry
dragonfruit

Range based for loop -

range(, ,)

range is an iterable datatype

by default = 0 ← start number | end number | step size → by default = 1

- To access elements from an iterable, we should iterate on it.

```
for element in range(0,10,2):
    print(element)
```

o/p ⇒ 0 2 4 6 8 (even number)

- Print the element along with its index in a list.

```
fruits = ["apple", "banana", "cherry", "dragonfruit"]
for idx in range(len(fruit)):
    print(idx, fruit[idx])
```

range(0,4,1)

o/p ⇒
0 apple
1 banana
2 cherry
3 dragonfruit

- Given list of student marks, the total masks in an exam and pass percentage as well, filter the students who passed and failed.

```
student_marks = [200, 350, 160, 400, 450]
total_marks = 500
pass_percentage = 40
```