



## Computer Technology Engineering Department

### Lecture: (2)

*If conditions & Loops*

**Subject: Object oriented programming II**

**Class: Second**

**Lecturer: Baqer Kareem**



## **If Statement**

Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions.

Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome is TRUE or FALSE otherwise.

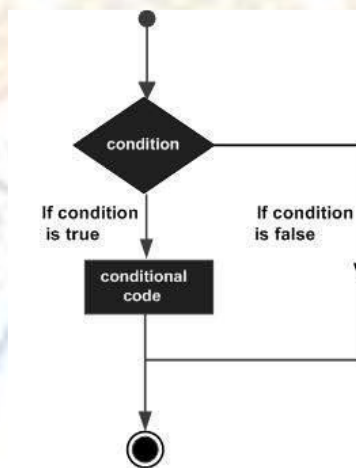


Figure 1.3. If statement

## **Type of If Statement**

- single line

Exp(1):

```
var = 100
if (var == 100):
    print("Value of expression is 100")
print("Good bye!")
```

- Nested Decisions

Exp(2):

```
x = 42
if x > 1:
    print('More than one')
    if x < 100:
```



```
        print('Less than 100')  
print('All done')
```

- Two-way using else

Exp(3)

```
x = 4  
if x > 2:  
    print('Bigger')  
else:  
    print('Smaller')  
print('All done')
```

- Multi-way (elif) stands for else if and we can have many elif statements

Exp(4)

```
x = 1  
if x < 2:  
    print('small')  
elif x < 10:  
    print('Medium')  
else:  
    print('LARGE')  
print('All done')
```

## While loops

- While loops are known as indefinite or conditional loops. They will keep iterating until certain conditions are met. There is no guarantee ahead of time regarding how many times the loop will iterate.
- The while loop, like the if statement, includes a boolean expression that evaluates to true or false. The code inside the loop will be repeatedly executed until the boolean expression is no longer true.

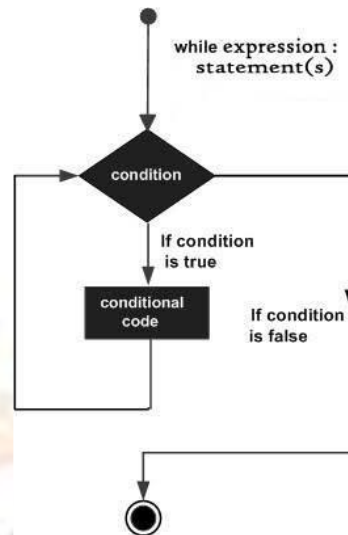


Figure 2.3. While Expression

Exp(5):

```
count = 0
while count < 5:
    print(count, " is less than 5")
    count = count + 1
else:
    print(count, " is not less than 5")
```

### For Loops

The for loop is a pretest loop that has built in expressions for initializing, testing and updating a counter. This makes it convenient to control the number of times the loop executes. For example, situations where you know the number of times the loop must execute.

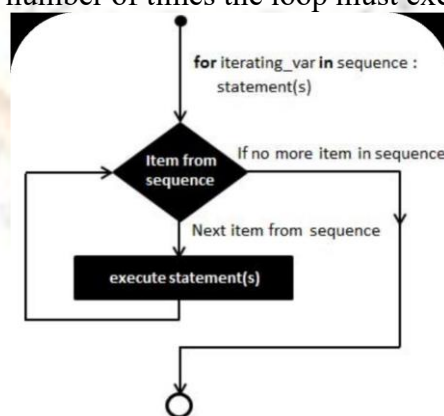


Figure 3.3. For Loops



Exp(6):

```
sum = 0
for i in range(10):
    sum = sum + i
print(sum)
```

### **Nested Loops**

- A nested loop is a loop inside the body of another loop
- Inner (inside), outer (outside) loops:

Exp(7):

```
for row in range(1, 3): # outer loop
    for col in range(1, 3): # inner loop
        sum = row * col
        print('sum =', sum)
```