C Programming:

Course 7:

ITERATIVE Statements

Part two

Dr. Chénche

for Loop

1.The update

Update:
Incrementation or
decrementation

```
int i ;
for ( i = 0 ; i < 10 ; i ++){
printf("The value of counter is:%d \n",i );}
```

```
Examples :
i++ // or i=i+1
i--
i+2 Not correct we write i=i+2
```

Note: you can update **multiple variables** in the *third expression* by separating them with **commas**.
Example: for (i = 0, j = 10; i < j; **i++, j--**)

for Loop

2. The condition

Condition

```
int i ;
for ( i = 0 ; i < 10 ; i ++){
printf("The value of counter is:%d \n",i );
}
```

If you want to use multiple **conditions** to control the for loop, use **logical operators**:

Example: for (i = 0, j = 10; i < 5 && j > 0; i++,)

`for` Loop

1.The initialization

```
int i ;
for ( i = 0 ; i < 10 ; i ++){
printf("The value of counter is:%d \n",i );}
```

initialization

You can use multiple initializations, separated by commas.

These are all executed **once**, at the very beginning of the loop.

From for loop to while / do..while loops:

```
for ( initialization; condition; update)
```

```
initialization;
While(condition)
{Update;}
```

```
initialization;
do{
Update;}while (condition);
```

Exercises :

Exercise 1 : (Course)

1) Translate the following “for" loop in “while“ and “do ...while” loop, and give the display.

```
for (i=50, x=0; i>=x; i-=10)
        printf("%d\n", i);
```

2) What can you say if x was initialized by 60?

Exercises :

3) Translate the following “for" loops in “while“ loop, and give the display.

```
for (i=0; i<3; i++)
    for (j=0; j<2; j++)
        printf(“i=%d  j=%d\n“, i, j);
```

Exercises :

Exercise 2 : (Course)

Write a C treatment allowing to :

1- Display the sum of integer numbers that are between 1 and 100.

2- Display the even integers that are between 1 and 80.

3- Display the odd integers that are between 1 and 80.

4- Display all natural numbers multiple of 5 that are between 1 and 100.

5- Display the sum of 10 integer numbers as well as their average.

6- Display the sum of n integer numbers (n>0) as well as their average.

7- Read an integer x>0, then double it as many times until it exceeds 60.