Lesson 74 Practice 17

In this lesson, We are going to see the exact role of return in a function.

The return statement is used to terminate the execution of a function and transfer program control back to the calling function. In addition, it can specify a value to be returned by the function.

When we call a function in any place, the value of the function equals the value returned by the function.

```
#include <stdio.h>
int fun() {
return 5;
}
int main() {
int x= fun();
//when we call fun() it returns  5
printf("%d", x);
}
```

Output :

```
5
```

Try the code : Click Here!

int function : must have a returned integer.

void function : returns nothing, no need to write **‘return *value*’** statement or write **return;**

In a function we can define variables in parentheses :

```
#include <stdio.h>
int fun(int a, int b) {
//we defined them between parentheses
return a + b;
}
int main() {
int x=3, y=5;
int z = fun(x, y);
/*   we must put the values for the variables
that we have defined in the function
Remark ; when setting variables when the function is called, it must be the same type as the data defined inside the function brackets
*/
printf("%d", z);
}
```

output:

```
8
```

Try the code : Click Here!

If the names of variables in main and function are similar, this does not mean that they are the same variable, but this is a variable that has a place to work in, whether it is inside function or whether it is inside main. Each variable has its own scope and is not related to the other, but if they are defined in global above main and the function, they are the same variables in all the functions.

Example :

```
#include <stdio.h>
int fun(int y) {
// Step 3 change y to x value
if (y > 0)
return 10;
return-10;
}
int main() {
int x = 3;
int z = fun(x);
//1-//z equals fun(x)
//2-// we execute function using the variable x
printf("%d", z);
}
```

output:

```
10
```

Try the code : Click Here!

```
#include <stdio.h>
int fun(int y) {
if (y > 0)
return 10;
return-10;}
int main() {
int x = 0;
int z = fun(x);
printf("%d", z);
}
```

output:

```
-10
```

Try the code : Click Here!

That **function returns** only one number and when you do **return** it will put the number next to **return** just where the function is called