

# Lec 6

## Functions::

is a set of statements that together perform a specific task

مجموعة من الجمل مع بعضها يستخدم لتنفيذ أمر معين ←

\* Every C program consists of one or more functions

### The main( ) Function

is mandatory for the C program

because it is the entry point of your C code from where your program is executed.

الـ main() هو المدخل الأساسي الذي يبدأ من هنا الكود

## Advantages of Function::

### ① modularity

The Function increase the modularity of the program

- A large problem can be divided into sub problems and then solved by using functions.

تقدر تعمل جزء خاص لكل حاجة بمعنى ت عمل كل مودول Function دى يقوم بوظائف

( فلو عندك مثلاً 3 مودول Function لـ 3 مودول Function )

### ② reusability

The Function increases the reusability because Functions are reusable

- Once you have created a Function you can call it anywhere in the program without copying and pasting entire logic, So you don't need to write the same code again and again.

تقدر تكرر الكود بمختلف وسائطه كثيرة ب مجرد استدعاء Function

و تقدر تتفق الكود بين زمالة سهلة

### ③ maintainable

Because Function increase the modularity of your program, so the program becomes more maintainable

- If you want to modify the program sometimes later, you only need to update your functions without changing the base code

## Types of Function:

### a) Library Function

→ built-in library Functions to perform various <sup>operations</sup>

عنوان اقدر استخدمها لا زم استعمل ال (header file) اللي موجود فيه الـ function  
ويمكنا اقدر استخدمها في الكود براحتي

Ex:

**header File**

#include <stdio.h>

**output** → printf ("Enter The Number Count : ");

**input** → scanf ("%i", &NumberCount);

هـ Function جاهزه موجوده في الملف يستخدمها علـي تسهل تنفيذ بعض الأوامر

بس شرط استخدمها انك تستعمل او الخاص بها headerFile

هـ نوعين من الـ Library Function

① open source

مفتوحة المصدر اقدر اعرف بتكون من اى

② close source

مغلق مقدريش اعرف محتواها  
(بتكون محفوظة من الشركة او من المبرمج عموماً)

## Types of Function:

### a) user defined Function

→ we can also create a function according to our requirements.

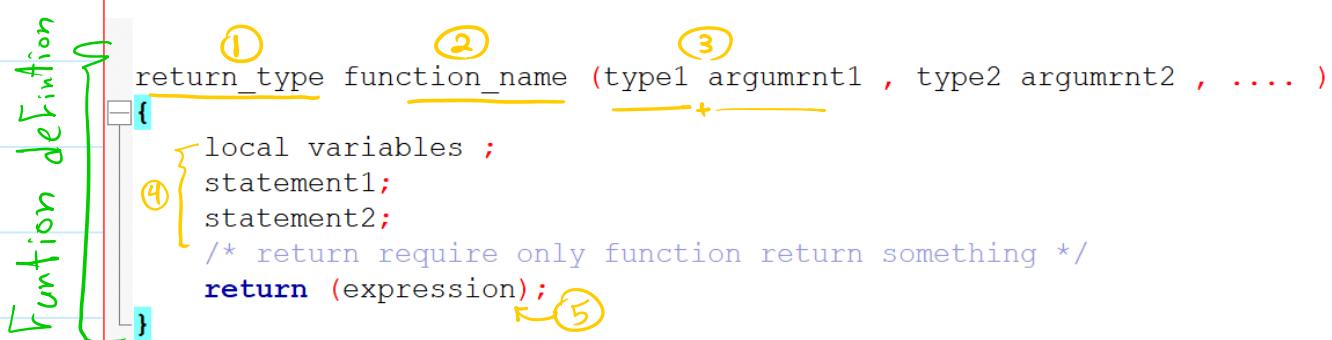
دی میں وظیفہ اور اپنے کام کا Function بنائیں۔

\* Any Function consist of:

- ① Function definition
- ② Function call
- ③ Function declaration

### 1- Function definition: (header + body)

ابتدئی Function کی actual presentation  
دا الکود الفعلی کو Function بتخواہی



main() يفخر يكون تحت دا Function definition

## 2- Function declaration::

return\_type function\_name (type1 argumrnt1 , type2 argumrnt2 , ... ) ;

السلسل دا كذا اسفل دا كذا Function prototype أو Function declaration دا

يفضل يكون فوق اد ( ) main

دا declaration يعني مجرد برقفه ان هر حاجة في البرنامج بتاعي اسمها كذا

و من بتتجزز مساحة من ال memory

(multiple definition) Function دا اكتر من دو نفس او declaration يمكن اعمل اكتر من دا function غير هر و واحد

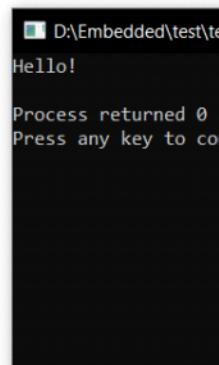
## 2- Function call::

عندي استخدم اد Function لازم اعملها call جوا اد main() بنادي عليها عندي استخدم

مكن استدعى Function جوا تانية دا وطبعاً جوا اد main() دا يمكن استدعىها في الفرع

Function دا يكون الاستدعاء هو انه تكتب اسم دا input وللو ليها بتدري لها

```
1 #include <stdio.h>
2 void print_Hello (void) ;
3 int main ()
4 {
5     print_Hello();
6     return 0 ;
7 }
8
9 void print_Hello (void)
10 {
11     printf("Hello! \n");
12 }
13
```



1) Function declaration

2) Function definition

3) Function call

input  
output

کل حاجہ ناپڑے فن کل اور الا Function

input  
output

اللی ممکن یہ عدد خواہیں اور Function

## Types of Function definition ::

- 0 No Return (void) - 0 No parameters (void)
- 0 No Return (void) - 1 parameter list (int x, int y, ...)
- 1 Data Type (int/float/...) - 0 No parameters (void)
- 1 Data Type (int/float/...) - 1 parameter list (int x, int y, ...)

return value لئے Function کی جو

و یا

parameter لئے Function کی جو

و یا

- ① 0 No Return (void) - 0 No parameters (void)

```
1 void function_name (void);  
int main ()  
{  
    3 function_name();  
    return 0;  
}  
  
2 void function_name (void)  
{  
    /* code */  
}
```

ممكن اكتب لها  
ويمكنها  
بس يفعل اكتبها

## (2) 0 No Return (void) - 1 parameter list (int x, int y, ...)

```
#include <stdio.h>
void function_name (unsigned int x , ...);
int main ()
{
    function_name(1 , ...);
    return 0 ;
}

void function_name (unsigned int x , ...)
{
    /* code */
}
```

وامر لا يرجع

أما إنها

لرجوع لها

## (3) 1 Data Type (int/Float/...) - 0 No parameters (void)

```
#include <stdio.h>
return data_type function_name (void) ;
int main ()
{
    return_data_type variable ;
    variable = function_name();
    return 0 ;
}

return_data_type function_name (void)
{
    /* code */
    return (expression);
}
```

القيمة التي ترجع

data type  
funcion return  
return من نفس النوع  
يكتب العبرية الى البرنامـج  
فـاـنـهـيـرـيـزـ

#### (4) 1 Data Type (int/Float/...) - 1 parameter list (int x, int y, ...)

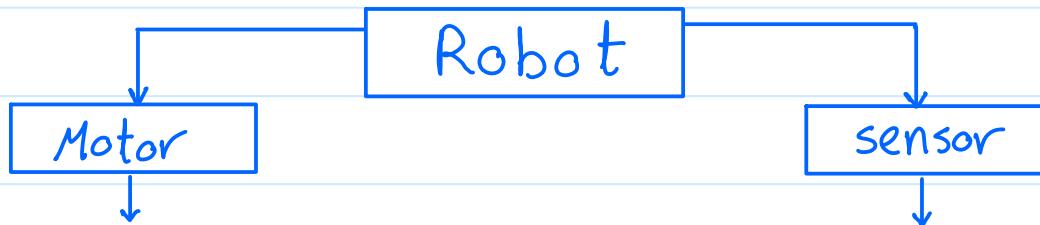
```
#include <stdio.h>
return_data_type function_name (type1 arg, type2 arg, ....);
int main ()
{
    return_data_type variable ;
    variable = function_name ();
    return 0 ;
}

return_data_type function_name (type1 arg, type2 arg, ....)
{
    /* code */
    return (expression);
}
```

Value *ویژگی* *کال* *کار*

### Functions (Modules and Multi Files Approach)

Assume we have a larg project (Robot) and this robot has may things to do :



Motor Module Functionality

- Move Motor Forward
- Move Motor Backward
- Move Motor left
- Move Motor Right

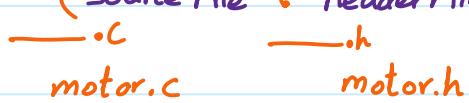
sensor Module Functionality

- Read Data From sensor
- Perform Calculations

\* فی المثال دا عندنا Robot عن project و هو بيكون من Motor & sensor

① Motor moduel , ② sensor moduel ← 2 moduel  
فنقول بقان عندنا

كل moduel ليه مجموعة من الـ Functions اللي هيقوم بيها

\* كل moduel يكون ليه على الأقل بيكون في ملفين ( source File & headerFile )  


```

    .
    +-- motor.c
    +-- motor.h
  
```

\* ممكن يبقا عندك أكثر من header file على حسب الـ project و لكن المفهوم اللي فيه

علاقان اربط اد Files مع بعض بروح اجمل include

? include ازاي اجمل \*

① بروح في الـ Source File اللي هو .c ولكن

#include "motor.h" ② أكتب هناك كما

كدا ربطت بينهم ومن شرط يكون الاشترين نفس الاسم لاء بروح من الـ .h و أكتب اسم .h اللي عاوز اربط بين

\* اما يكون اد header file دا انا اللي عامله خطوة بين " .h " اما

#include < .h > كذا لكن لو هو اصلا library موجودة فخطوه

## Source File

يكتب الـ definition الخاص بالـ Functions

① يحمل الأول " .h "

الـ declaration اللي موجود فيه الـ header

definition بس اجمل " .h " دى Function

( Linker error ) ← هرض لحدوث

\* استخدام Function من معمول لها definition

8  
٢٠٢٣