

Lec 6

Functions:

- A function is a set of statements that together perform a specific task

مجموعة من الجمل مع بعضها تستخدم لتنفيذ أمر معين

- Every C program consists of one or more functions.

أي برنامج C يتكون من فائتين أو أكثر

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.

ال main() هي نقطة الدخول لبرنامج C

Advantages of function:

1. increases the modularity of the program

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

أو عندك برنامج كبير ممكن تقسمه لمجموعات من الفائتين

2. increases the reusability because function 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.

Because function increases the popularity of your program So the Program becomes more **Maintainable**

* لو انا محتاج اعدل البرنامج بعد كذا صقلت اعدله الفانكشن على عكس
لو انا كتبت الكود فخطئ ان اعدل كل المكتوب

Types of function:- أنواع الفانكشن؟

a) library function.

built in function to perform various operation

Example:-

Input-output → printf(), scanf() → <stdio>

String → strcpy, strcmp → <string.h>

* هي عبارة عن فانكشن جاهزة موجودة في اللغة يستخدمها عشاق تسهل
على تنفيذ بعض التوامر

* عشاق اقدر استخدم الفانكشن دي لازم اعمل include لل header
file الخاص بيها الا بيها موجود فيه definition الفانكشن و
وظيفتها اياه وتعمل اياه بعض الفانكشن بتسا open source
مفتوحة المصدر اقدر اعرف تكوينها والبعض بيها close
مغلق من الشركة او من المصدر ومغلقاً

b) User defined function

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

بتسا عبارة عن فانكشن انا اعملها كبرمج عشاق انفذ وظيفة معينة

3-function declaration

- بتکون من جزئیں
function header ←
function body ←

3

return (expression) و

3

main function لکھو

2 Function declaration

و (`return type function-name (arguments)`)

* دأكرأ اسم Function decl أو Prototype الخاض بالفانكشن
ويفضل يكون فوق ال main

* وزي ما احنا عارفين إنه أي حاجة فيها declaration
هتبقى ليها مساحة في الميموري

* ممكن اعل أكثر من declaration لنفس الفانكشن عادي
لكن ال ميثفكش اعل أكثر من مرة هو ال Function definition

3-Function call

* بشان اقدر استخدم الفانكشن لازم اعملها Call زي
واستدعاء للفانكشن

* ممكن اعل واستدعاء لفانكشن وانا في فانكشن تانيه وممكن
اعلها واستدعاء في ال main من ال ميثفكش اعلها واستدعاء في الفراغ

Example:-

#include <stdio.h>

Step 1:

Function declaration

void print_hello (void);

int main() {

Step 2:

function definition

print_hello();

}

void print_hello (void)

{

Step 3:

Function call

printf("Hello-Ahmed\n");

}

Types of function:-

- 1- No input No output
- 2- No input Yes output
- 3- Yes input No output
- 4- Yes input Yes output

* المقصود بال input هو أن الالة لها Parameter أو argument ولا

* المقصود بال output هو أن الالة قد لها Return value ولا لا يعني يرجع قيمة ولا

1- No input No output

```
void function_name (void);
```

```
int main()
{
```

```
    function_name();
```

```
}
```

```
void function_name (void)
```

```
{
```

```
    statement 1;
```

```
    statement 2;
```

```
}
```

↑
+ تسمى الدالة void
ليس الأفضل وأتركك كليهما

2- No input Yes output

```
return type function_name (void);
```

```
{ int main() {
```

```
    // initialization
```

```
    → return type = function_name();
```

```
}
```

```
→ return type function_name (void) {
```

```
    statement 1;
```

```
    return (expression);
```

```
}
```


3- Yes input No output

```
void function_name (type1 arg , type2 arg, ---);
```

```
int main() {
```

```
function_name (argument1, argument2, -----);
```

```
}
```

```
void function_name (type1 arg , type2 arg, ---)
```

```
{
```

```
statements;
```

```
}
```

4- Yes input Yes output

```
return_type function_name (type1 arg , type2 arg, ---);
```

```
int main() {
```

متغير رئيس نوع

```
return_type = function_name (arguments);
```

```
}
```

```
> return_type function_name (type1 arg , type2 arg, ---)
```

```
{
```

```
statements;
```

```
return (expression);
```

```
}
```

Functions (Modules and Multi Files Approach)

Assume we have a large Project (Robot)
and this robot has many things to do

Robot

Motor

Sensor

Motor Module Functionality

- Motor Move Forward
- Motor Move backward
- Motor Move Left
- Motor Move Right

Sensor Module Functionality

- Read Data from sensor.
- Perform calculations.

هناك هنا مثالان أحدهما عندنا بروجيكت أو مشروع روبروت
يتكون من سنسور وماتور مثلاً
فأحنا بنقول إن الماتور دا هوديول والسنسور يا هوديول
وكل هوديول من دول ليته مجموعة من الوظائف أو التاسطات

كل هوديول من دول بيبقا ليته على الأقل ملفين
source file ←
header file ←

Motor.c

Motor.h

ملف بقا عندي أكثر من source file واحد أو أكثر من
header file واحد على حسب ال Project

* عشان افكر الفاييلين مع بعض لازم اعمل include الفاييل
header ع source

- The "source file" includes "header file"

* ازاى اعمل include ؟
بكتب ع فايل ال source كذا

#include "motor.h"

* كذا اعملت رابطين ال source file وال header file

* هنا اعملت " " عشان يعمل ال file وانما لو library هعمل < >

* ع source file

definition بكتب ال

function الخاضع بار

وإني بعد اعملت include
لل header

* ع header file

بكتب ال prototype الخاص

بال Functions بس

تويعمل declaration يعني
لل function

* لو أنا جيت استخدم فانشن وانما مش عامله definition
مينقول Linker error