

Lec 2

program structure :

* main() Function

البرنامجه يكون فيه main function وادره بس لا

*PrintF() function

دی **function** دی **user** مها علشان اقدر اطبع واظهر على الشاشة لا
عشنان اقدر استخد مها لازم اكون عامل استدعاء للهكبة اللى فيها **Function**
#include < stdio.h >
standered input output

Note

.c → source file

• h → header file

`printf(" ");`

" " عشان اظهر نص كما هو بخطه بين " double quote

```

1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello, I am Mohamed Helmy! \n");
5     return 0;
6 }
7

```

```

D:\Embedded\test\test.exe + v
Hello, I am Mohamed Helmy!
Process returned 0 (0x0) execution time : 1.394 s
Press any key to continue.

```

*escape sequences:

is a sequence of characters that does not represent itself when used inside a character or string literal, but is translated into another character or sequence of characters that may be difficult or impossible to represent directly.

List of escape sequences

escape sequence	Name	Description
\a	Alarm or Beep	it is used to generate a bell sound in the C program
\b	Back space	it is used to move the cursor back word

escape sequence	Name	Description
\f	Form Feed	it is used to move the cursor to the start of the next logical page
\n	new line	it moves the cursor to the start of the next line
\r	Carriage Return	it moves the cursor to the start of the current line
\t	Tab (Horizontal)	it inserts some white space to the left of the cursor and moves the cursor accordingly
\v	Tab (vertical)	it is used to insert vertical space
\	Back slash	use to insert back slash character
\'	single quote	use to display a single quotation mark
\"	Double quote	use to display a double quotation mark
\?	Question Mark	use to display a question mark
\ooo	octal number	it is used to represent an octal number

escape sequence	Name	Description
\xhh	Hexa decimal number	use to represent the hexa decimal number
\0	null	it represents the Null number

Example ::

.printf "ahmed\" → printf ("\" ahmed\"");

.printf "ahmed" → printf ("\" ahmed\" ");

.printf \"ahmed\" \→ printf ("\\\" ahmed\\\"\\");

```

1 #include <stdio.h>
2 int main()
3 {
4     printf("\\\"ahmed\\\"");
5     printf("\n");
6     printf("\\\"ahmed\\\"");
7     printf("\n");
8     printf("\\\\\"ahmed\\\"\\");
9
10    return 0;
11 }

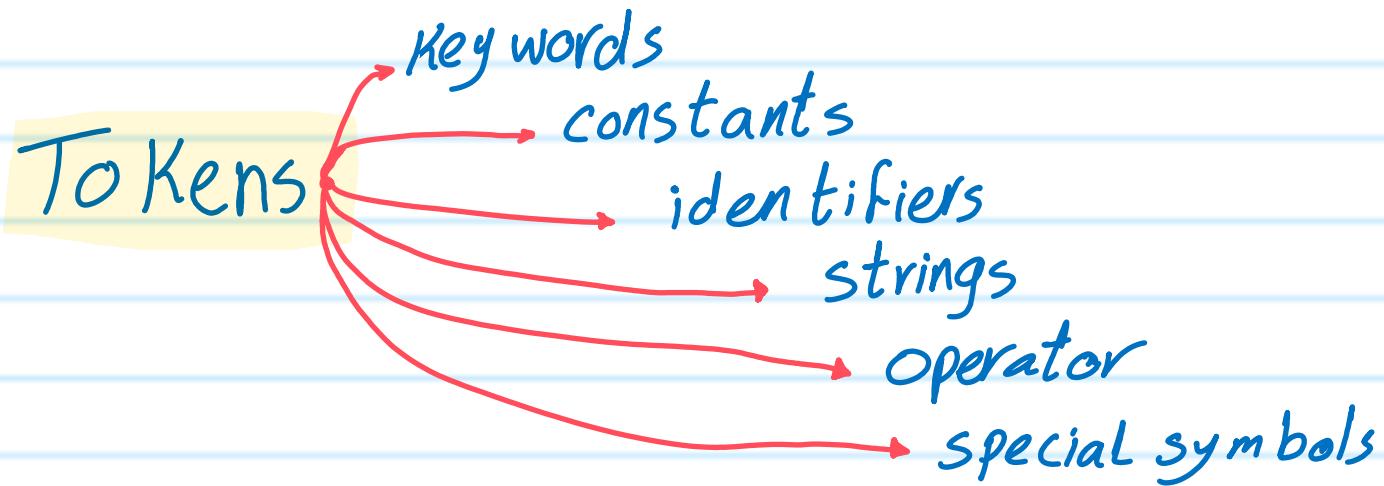
```

```

D:\Embedded\test\test.exe
\ahmed\
"ahmed"
\"ahmed\"
Process returned 0 (0x0)   execution time : 0.907 s
Press any key to continue.
|
```

C - Basic syntax :

a C program consists of various tokens



ال Compiler J عشان يفهمون printf ("Hello,world")

ويشوف أنت كاتب مع ولا 8

لوكتب خطأ بطلعلك error

لوكتب مع هيتحولها machine code .
ويباً يفهمه

Compiler J يقوم بها ال Tokenization

* العمليات اسها

printf ←
(←

"Hello,world" ←
) ←
; ←

* SEMI COLONS

- In a C program, the semicolon is a statement terminator.
- That is, each individual statement must be ended with a semicolon.
- It indicates the end of one logical entity.

```
printf("hallo");  
return 0;
```

Comments o Comments are like helping text in your C program and they are ignored by the compiler.
o They start with /* and terminate with the characters */ as shown below :

* COMMENTS

- Comments are like helping text in your C program and they are ignored by the compiler.
- They start with /* and the terminate with the characters */

```
1 /*  
2 * Sample preview code  
3 * This is a block comment  
4 */  
5  
6 #include <iostream> // this is a line comment  
7 #include <cstdio>  
8  
9 /**  
10 * This is a documentation comment block  
11 * @param xxx does this (this is the documentation keyword)  
12 * @authr some user (this is the documentation keyword error)  
13 */  
14
```

* Identifiers

- A C identifier is a "name" used to identify a variable, function or any other user-defined item.
- An identifier starts with a letter A to Z, a to z or underscore '_' followed by zero or more letters, underscores, and digits (0 to 9).
- C does not allow punctuation characters such as @ \$ %.
- C is a case-sensitive programming language.

ایڈنٹیفایر یعنی اسم یا رقم کا نام ہے Identifier یعنی اسم یا رقم کا نام ہے Note
descriptive دسکرپٹو یعنی مذکون ہے ↪

* Key words

Identifier یعنی اس اسم یا رقم کا نام ہے جو الگ الگ کلمات میں ہے اور کوئی دوسرے کام کے لیے استعمال نہ کیا جائے۔
variable یعنی اس کام کا نام ہے جو کوئی فریکشن کے لیے استعمال کیا جائے۔

auto break case char const continue default do double else Enum extern float for

* White spaces

C compiler totally ignores it.

* Data Types :

- C use data types for declaring variable or functions of different types.
- The type of variables determines how much space in storage and how the bit pattern stored is interpreted

Basic types

ⓐ Integer types

They are arithmetic types

ⓑ Floating types

أجنبية ملحوظة \times ملحوظة ملحوظة
موجبة أو سالبة

Enumerated types

arithmetic types

Derived types

- pointer types
- Function types
- structure types
- Array types
- union types

The type void

Integer data types

Data Type	Memory (bytes)	Range
short int	2	-32,768 to 32,767
unsigned short int	2	0 to 65,535
unsigned int	4	0 to 4,294,967,295
int	4	-2,147,483,648 to 2,147,483,647
long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295
long long int	8	-(2 ⁶³) to (2 ⁶³)-1
unsigned long long int	8	0 to 18,446,744,073,709,551,615
signed char	1	-128 to 127
unsigned char	1	0 to 255
float	4	
double	8	
long double	12	

Floating-point types

ازای تجربہ Range Data type کی

اوشوف ال size بشاعه کام

Bit و Byte در جویار

Bits لا عدد هي 2^n .

unsigned و signed کو شوف ہو

٥. لو signed ← ناتج \parallel^2 هتحتسه على ٢ و الرقام السالب يبقا أعلى من الموجب
بمقدار و ا Mehr

Ex: $\text{int} \rightarrow 2\text{byte} \rightarrow 2 \times 8 = 16 \text{ bit} \rightarrow 2^{16} = 65536$

$$32768 = \frac{65536}{2} \leftarrow \text{Signed ١٦ bit المُرْفَع$$

بیت المدی هن 32768 - عال 32767

٦٠ طب لو unsigned كا الرقم موجب والمدى يكون من Zero ل 2^n
نفس المثال اللي فوق لكن الرقم هنا unsigned

الحادي من حضر الـ 65535

• في حاله استخدام datatype غير مناسب
• هيتم استهلاك الذاكرة وكثرة حجم البرنامج

Note

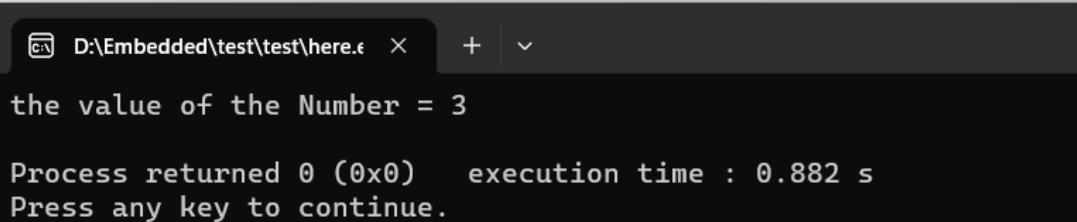
• من حال تخزين قيمة أكبر من حجم datatype إلى مفروض يشتمل
هذا يحصل Compiler error due to syntax مبنوطة خطأ
crash value لكن الذي يحصل أنه المتغير ليس له قيمة فيه تكون فيه

* data type (int): Compiler depend
2bit و غيره يعبره compiler

use short int → 2 byte } with signed
Long int → 4 byte } or unsigned
Long long → 8 byte

ركلز في المثال دا:

```
1 #include <stdio.h>
2
3 int main () {
4     unsigned int Number = 3 ;    ①
5     printf("the value of the Number = %i \n", Number);
6
7     return 0;
8 }
9
```



① كذا أنا حجزت مكان فاراد size أو Number raw memory و خزنت فيه قيمة ساوي 3 من نوع unsigned int

لو أنا عاوز أطبع قيمة المتغير Number بعمل حاجة اسمها data specifier

printf ("%i", Number); variable → المعاوز يعني قيمة المتغير

نوع variable اداري حفظ قيمة نار ؟ يتغير على حسب ادار datatype

To get the exact size of a type or a variable on a particular platform "Compiler"

We use Operator called Size of operator
Size of (datatype or variable)

- printf("%d", sizeof(char));
- printf("%d", sizeof(int));
- printf("%d", sizeof(float));

printf ("%_d", sizeof (متغير))
byte ← ترجع لنا القيمة بال size of

رکزخ المثال كذا:

```
unsigned char Test1 ='a'; →①
unsigned char Test2 =97; →②
int main ()
{
    القيمة التي عاوزها تطلع
    printf("Test1 = %i \n", Test1);
    printf("Test1 = %c \n", Test1);
    printf("Test2 = %i \n", Test2);
    printf("Test2 = %c \n", Test2);
```

```
D:\Embedded\test\test\here.c
Test1 = 97
Test1 = a
Test2 = 97
Test2 = a
```

① حجزت مكان خالٍ في الـ memory من نوع Test1 و من نوع 'a' ← و حطيت فيه قيمة (حرف)

② نفس الكلام و لكن اسمه Test2 و حطيت قيمة (رقم) ←

انا لما جيت المبرمج Test1 قوله ثغر عاوز القيمة اللي تطلع

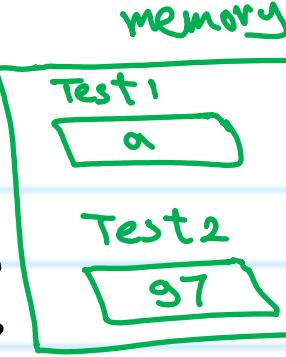
تكون من نوع int ← integer رقم يعن

أنا مخزن حرف (a) أصلـا

عن ازاي هيطبع رقم ← Ascii code

يروح يشوف a تساوى كام ويطبعها فطبع 97

ولكن لما طلبت يطبع كـ c ← character طبع a عادي



∴ data specifier →

يلو نوع اداتا اللي عاوزها تطلع

من اللي هي عليه

عن Test2 موجود 97 اذا عاوز المبرمج حرف بقا هيعمل اى

هياخد الا 97 ويشوف بتساوي كام خار Ascii ويطبع فطبع ← a

Format	Type of output	Format	Type of output
%d - %i	decimal integer - signed integer	% s % lf % Lf	string, sequence character double long double
% c	singed character	% o	octal number
% f	Signed Float	% u % ld	short unsigned integer long decimal Integer
% e	A float-point number	% X % P	Hexadecimal Integer print memory address in the hexadecimal form

تحدد تخزن فيها قيمة معرفة او حرف char

تستخدم خواص الأرقام المكسرية و الأسرار ملخص عاوز أفهم

Floating-point types

عدد الأرقام بعد العلامة

precision →

Float	→ 4 byte	range	6
double	→ 8 byte	2^n	15
long double	→ 16 byte	وأحسب بقى	19

```
1 #include <stdio.h>
2 float Test1 = 3.145359;
3 int main ()
4 {
5     printf ("Test1 = %f \n", Test1);
6     printf ("Test1 = %0.1f \n", Test1);
7     printf ("Test1 = %0.3f \n", Test1);
8     return 0;
9 }
```

D:\Embedded\test\test\here.c + | -
Test1 = 3.145359
Test1 = 3.1
Test1 = 3.145

Process returned 0 (0x0) execution
Press any key to continue.

نفس الكلمة حجزت مكانها درجة وحذفت قيمة ← 3,145359

① عاوز الطبع الرقم وهو عشري فطبع له %f

② حددت عدد الأرقام بعد العلامة %f اربع %

③ حدد 3 أرقام بعد العلامة %.3f

لو حددت 2 هي مثل تقرير (3,15) ولكن لو فتقرب

unsigned و Signed هي نفس الشيء استخدم

Floating type مع او

هيدرك syntax error

ملحوظة

* اى الفرق بين variable declaration & variable definition

variable declaration

عمل declare يعني بعرفك ان انا عندي variable في البرنامج و متحجز لها مساحة في الميموري

يستخدم معاه نوع قبل نوع data type او

variable definition

definition ← variable ← بدل لـ declaration
يتعجز لها مساحة في الميموري ←
declare ← بنفس الوقت بعده

unsigned int x = 99;

مكان يبقا له defention ←
وبالتالي يبقا فيه مساحة في الميموري

Ex: extern unsigned int x=99;

هنا في تركيبة declaration ← Variable ← بس هديله initial value ←
كابقا definition ← التجز لها مساحة في الميموري
يعنى كذا او ملهاش لا زمان extern

طلب لو انا عامل variable declaration ←
و هسته هديله قيمة زي كذا
extern unsigned int x;

وجيت استخدمت او x را جوا او main و عملت ز 3
وجيت اعمل اعمل Linker error ← Build & Run لأن انا كذا
بحاول اخزن في ملحوظ مساحة في الميموري

ممكن اعمل declaration مثل `int a = 10;` و `float b = 10.5;` و `char c = 'A';` و `String d = "Hello";` و `boolean e = true;`

extern char test!;

```
extern char test1;
```

```
extern char test();
```

```
unsigned Numberone = 3;
```

لو ڈا مدد دش نوع اور data type لا زی کا compiler بتا جائے یعنی علی اور size int ہی گتھرہ Compiler by default ← ←

```
unsigned char test1, test2, test3;
```

* العمليه دی اسمها variable list بعرف کذا من نفس النوع

في نفس المطر

بے دا ہن Recomended ایک تعامل کل Recomended فر سطر لو جدے۔

definition class وان variable ن initialize ممكن است

Ex: `unsigned char test1 = 3;`

?

Global variable

8

local variable

الفرق بين

Global variable

local variable

ای هتغیر موجود بر اد و مش جو main Function

ای متنیکر موجود نی local [] block

Global vari.-qawl { } gl

لومنڈا میں وہ قيمت و بتقا جو

﴿ لوہن مریاہ پیغمہ فتبقا ساوی Zero ﴾

يُعنَى قِيمَهُ غَيْر مَعْرُوفَه

«يَخْفِي لَوْمَتْنَا لَهُ الْمُتَعَذِّرْ قِيمَهُ أَخْلَقَهُ يَسَاوِي هَذِهِ»

Constants ::

refer to fixed value that the program may not alter during its execution.

النوابت . هي عبارة عن قيمة ثابتة لا يمكن تغيير بالطرق العادي
string literal < double < float < int < char ممكن تكون من نوع

Const float Pi = 3.14;
or float const Pi = 3.14;

أى حاجة على

* Constant Global

• هنفعت اغير قيمته ابا

• اقدر اقرأها بس

• لو عملته ومتدوش قيمته كرا مالوين لازمه

لأنه ه يكون constant قيمته بضرر دايم

و هقدر شرط اعدل عليه

* Constant Local

* يفع اغير قيمته ولكن هن بطريقة مبتكرة

* هغير قيمته بلا استخدام ال Pointers

* لو عملته ومتدوش قيمة
garbage value هتبقا

Ex:

int main() {

 const int x = 5; → عده

 constant

 int *ptr;

 ptr = & x;

② عاوز اغير قيمته

 *ptr = 7;

• هعمل بوينتر
و اغير قيمته

 return 0;

• كدا خترت

}

• قيمه او