

## Lec 2

## \* Function main()

- ← هذه الدالة الرئيسة للبرنامج ، هي نفس برنامج مكتوب ال Function دى
- ← أولهم بضغط على (Build & Run) البروليسور يبدأ بتنفيذ البرنامج
- من عند ال main() وتسمى ← entry Point
- ← البرنامج يبدأ فيه فالكشن واحدة بس لا main

## \* Printf Function:-

- ← استخدمها عشوات اقدر اطبع على cmd لا user
- ← لازم عشوات اعرف استعمالها لازم اكونه مستخدم او عامل Library لا include
- الخاصة بيها ال header <stdio.h> ← standard input output
- printf (" ");

## \* escape sequences

- \* The escape sequence in c is the characters of the sequence of characters That can be used inside the string literal.

- \* The Purpose of the escape sequence is to represent the characters that cannot be used normally using the keyboard.

- \* some escape sequence characters are the Part of ASCII char set but some are not.

## escape sequence List:-

escape sequence	Name	Description
\a	Alarm or Beep	→ it is used to generate a bell sound in the program
\b	Back space	→ it is used to move the cursor backward
\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	Horizontal Tab	→ it inserts some white space to the left of the cursor and moves the cursor accordingly
\v	Vertical Tab	→ it is used to insert vertical space
\\	Backslash	→ Use to insert backslash character
'	single Quote	→ it is used to display a single quotation mark
"	Double Quote	→ it is used to display a double quotation mark
\?	question Mark	→ it is used to display a question mark



escape sequence	Name	Description
\ooo	octal number	it is used to represent an octal number
\xhh	Hexadecimal	it represents the hexadecimal n
\0	NULL	It represents the Null character

### Example:-

\* اوتاي اربع دي ← \ahmed\

printf("\ahmed\");

في الحالة دي كدا انا ليه ؟ لانه \ا كدا معناها alarm

وال \ كدا معناها \ا كذا هيطبع

فمعنا كدا اوتاي اربع \ بالاساس يستخدم \ا كذا بالاساس

printf("\\ahmed\\"); → \ahmed\

\* طب اوتاي اربع "دي كذا" ahmed" هخط قبل بالاساس

printf("\ahmed\");

\* طب اوتاي اربع \ahmed\

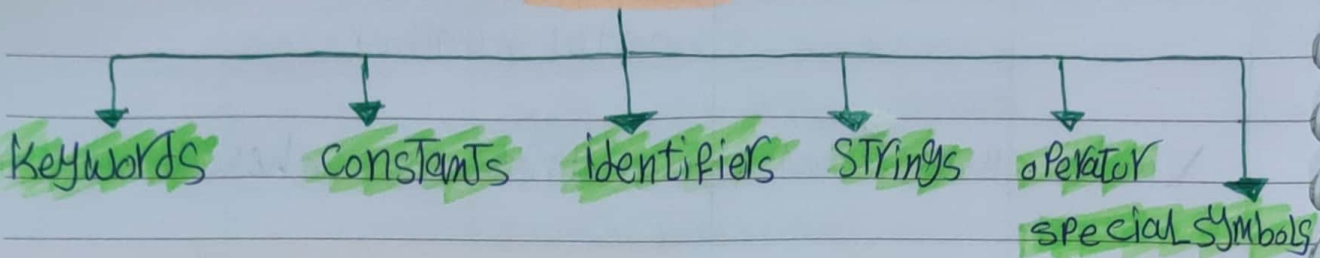
printf("\\ahmed\\"); ولو انا قفزت سطر جديد \n

← لو جالو سؤال امشي من الشمال للمين وامشي خطوة خطوة

## C-Basic Syntax

a c program consists of various tokens Like-

### Tokens



ال compiler هي التي تقوم بتحويل البرنامج المكتوب بلغة C إلى لغة الآلة (machine code) التي يفهمها الحاسوب. هذه العملية تسمى Tokenization. في هذه العملية، يقوم الحاسوب بتقسيم البرنامج إلى أجزاء صغيرة تسمى Tokens. كل Token يمثل شيئاً معيناً مثل كلمة مفتاحية (keyword) أو متغير (variable) أو علامة (operator) أو سلسلة نصية (string) أو ثابت (constant) أو رمز خاص (special symbol). إذا كان البرنامج مكتوباً بشكل صحيح، فإن كل Token سيكون له معنى. أما إذا كان مكتوباً بشكل خاطئ، فإن الحاسوب سيعطي رسالة خطأ (error message) وتسمى هذه العملية بالتحليل (parsing).

### \* Semicolons

- in a c program, the semicolon is a statement terminator.
- That is, each individual statement must be ended with a semicolon

### \* 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, an underscore '\_' followed by zero or more letters, underscores, and digit (0 to 9)
- c doesn't allow punctuation characters @, \$, %



- C is case-sensitive Program Language.

Some Example of identifiers are recommended :-

int NumberOne  
int Number\_One  
int number\_one

من الأفضل أن يبدأ الـ identifier بأرقام وخط تحتها ليس  
ويفضل أن يكون descriptive  
مسافة

**Keywords:-**

هذه هي الكلمات ممنوعة استخدامها كـ identifier يعني ممنوعة  
استخدامها بأسماء متغير أو وظائف

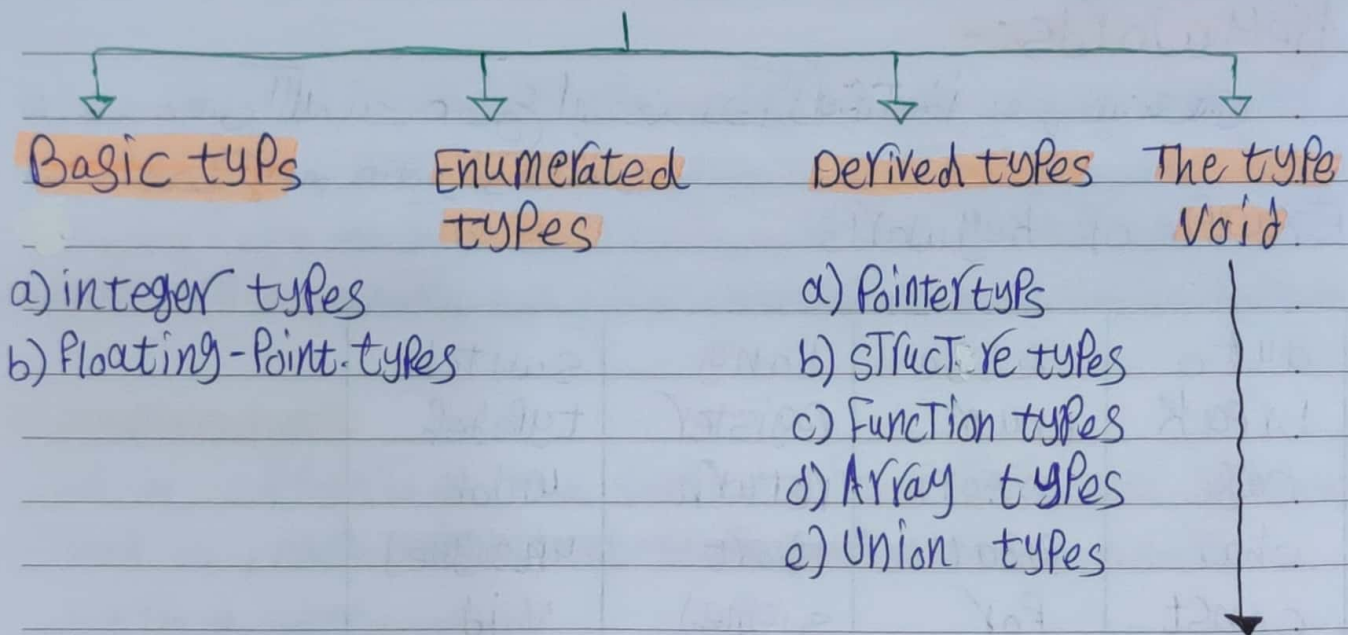
Example of keywords

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			

## C-Data types

- Data types in C used for declaring variables or functions of different types.
- The type variable determines how much space it occupies in storage.

### Data types



Void indicates that no value is available



## integer Data Types

type	storage size	value range
char	1 byte	-128 to 127 or 0 to 255 (Depends on the IDE)
unsigned char	1 byte = 8 bit	0 to 255
signed char	1 byte	-128 to 127
int (default signed)	2 or 4 bytes	-32768 to 32767 or -2147483648 to 2147483647 → Depend on the compiler
unsigned int	2 or 4 bytes	0 to 65535 or 0 to 4294967295
short int (default sign)	2 bytes	-32768 to 32767
↑ long int (DS)	4 bytes	-2147483648 to 2147483647
short (DS)	2 bytes	-32768 to 32767
unsigned short	2 bytes	0 to 65535
long (DS)	4 bytes	-2147483648 to 2147483647
unsigned long	4 bytes	0 to 4294967295
long long (DS)	8 bytes	-9223372036854775808 to 9223372036854775807
unsigned long long	8 bytes	0 to 18446744073709551615

Range

• إذا كان مقدار حسب الرتبة لا داتا تايب ؟

١- لنشوف أول حاجة ال size الخاص بيها دايه

٢- ونشوف كاضربت يعني أحولها لبت

$$\text{byte} = 8 \text{ bit}$$

$$2^n$$

حيث  $n$  عدد البتات

٤- لنشوف هو signed و unsigned

٥- لو signed  $\leftarrow$  هجيب نتائج  $2^n$  واقسمه على 2 ويبقا الرقم السالب اعداد من الموجب بمقدار واحد (نفس نتائج القسمة وتلك الموجب اقلها)

$$\text{مثال لو هو 2 byte} \leftarrow 16 \text{ bit} \leftarrow 2^{16} \leftarrow 65536 \leftarrow \div 2$$

$$\leftarrow 32768 \text{ داتا نتج القسمة اما الذي هيبقا كذا}$$

$$-32768 \text{ to } 32767$$

$$\leftarrow \text{مثال تاني 4 bytes} \leftarrow 32 \text{ bit} \leftarrow 2^{32} \leftarrow 4294967296 \leftarrow \div 2$$

$$-2147483648 \text{ to } 2147483647$$

٦- هيب لو unsigned كذا معناه ان الرقم موجب والذي يتاوه من 0 ل  $2^n$

$$\text{مثال لو عندك 2 byte} \leftarrow 16 \text{ bit} \leftarrow 2^{16} \leftarrow 65536$$

$$\text{كدا الذي من صفر ل } 65536$$



\* To get the exact size of a type or variable on Particular Platform "compiler", you can use the sizeof operator like this:-

```
printf("%d", sizeof(char)); printf("%d", sizeof(متغير));
printf("%d", sizeof(int));      Number one
printf("%d", sizeof(float));   size of الـ size بعد الـ
```

```
unsigned int NumberOne = 3;
```

كما انا تجزئت مكانة في الذاكرة اسمها NumberOne size 3  
هنا نوع unsigned int و خزن فيه قيمة (رقم) تساوي 3  
طب لو عاين الـ value قيمة NumberOne

```
printf("%i\n", NumberOne); → syntax code
```

Specifier

طب لو انا عاين اربع حرف مثلا %C هسخدم

Format	type of output	Format	type of output
%d or %i	A decimal integer & signed integer	%s	string, sequence character
		%lf	double
		%Lf	long double
		%o	octal number
%c	signed character	%u	short unsigned integer
%f	signed float	%ld	long decimal integer
%e	A floating-point number	%x	Hexadecimal integer

%P Print memory address in the hexadecimal form

char ← أقرأ خزن فيها قيم معينة أو حرف واحد  
العلية دي من خلال specifier

### Floating-Point types

لستخدوها في حالة الأرقام العشرية والكسور فارق معايها يعني مهم  
كام رقم بعد العلامة العشرية

Type	Storage size	Value Range	Precision
Float	4 byte	1.2E-38 to 3.4E+38	6
double	8 byte	2.3E-308 to 1.7E+308	15
long double	16 byte	3.4E+932 to 1.1E+4932	19

\* أي الفرق بين Variable declaration و Variable definition

يعمل الـ Variable definition ← تعريف ← يعمل الـ declaration يعني يعرفك ان  
مساحة الـ ميموري وفي نفس الوقت أنا عني variable في البرنامج  
يعمل الـ declare  
و متجرب لو شمساحة في الميموري  
لستخدم مع الـ كلمة extern  
قبل نوع الداتا كاي  
ممكن بقا الـ definition في فايل  
تاني وبالتالي بقا الـ مساحة  
في الميموري



extern unsigned int var = 99;

\* هنا تركايب variable ← declaration ← بس مديله قيمه كذا  
 بقا variable ← definition ← الجيز له مساحة في الذاكرة  
 يعني ال extern هنا هلاش لدرجه

\* طب لو أنا عامل variable declaration وهش مديله قيمه زي كذا مثلا

extern unsigned int var;

وحيت استخدمت ال var دا جوا ال main وعملت كمان

var = 3;

وحيت اعمل Build & Run ← هيجب Linker error لاننا كذا  
 نحاول اخزن في variable ملهوش مساحة في الذاكرة.

extern char test1;  
 extern char test2;  
 extern char test3;

\* ممكن اعمل declaration اكثر من مرة ل variable كذا

unsigned Number one = 3;

\* لو أنا محدتش نوع ال data تايب ل variable زي كذا default by compiler  
 هيعتبره int وال size بتاعه بيعتد على ال compiler

unsigned char test1, test2, test3;

العملية دي اسمها variable list بتعرف كذا variable من نفس النوع  
 في نفس السطر بس دي مش recommended ال recommended انك تعمل كل  
 variable في سطر لوحده.

ممكن اعمل initialize ل variable وانا بعل declaration يعني اديله قيمة

unsigned char test1, test2 = 3, test3 = 6;

## \* الفرق بين Global Variable &lt; Local Variable

## Global Variable:-

أي متغير موجود برا ال main و مش ففائلشن أو جوف في اسمه  
Global Variable ولو أنا مدلووش قيمة تبقى قيمة تساوي صفر

## Local Variable

أي متغير موجود بين block ← في اسمه Local ولو أنا مدلووش  
Initialize تبقى قيمة Garbage Value يعني قيمة غير معروفة

## Constants:-

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

التوابت :- هه عبارة عن قيم ثابتة ممكن تبقى هه نوع char أو int  
أو float أو double أو string literal

## example :-

float const PI = 3.14 ← مش هتفرق  
const float PI\_1 = 3.14 ← سواء float const أو const float

## constants

Local

Global

## Global:-

← مينفعش اغير في قيمة أبدأ هه اقدر اقرأها بس  
← مينفعش اعدل Global ← const ← من غير Initialize كدا ملوش ادره  
لونه constant قيمة بصرف دايه او مغيرش اعدل فيها



**Local:-**

- لنضع اغير في قيمته بس بطريقة غير مباشرة باستخدام ال **Pointers**  
 - نوع الـ **local** ← **const** ← **Initialize** ← قيمته **Garbage Value**

```
int main () {
```

```
    const int S = 5;
```

```
    int *ptr;
```

```
    ptr = &S;
```

```
    *ptr = 7;
```

```
}
```

\* لو انا غير قيمته

هل بوينتر والغير

قيمة البوينتر

\* كما اننا غيرت قيمه

الـ **const**