Learn Programming Basics (C Language)

LESSON #004
Introduction to C Programming

Topic for C programming

2. some Unix/Linux command
3. stages of Compiling
4. introduction, data types
5. printf, scanf, Precedence Table
6. if, else if, else, ternary, switch, break
7. for loop, while, do while, continue, break
8. function declaration, definition nation, read complex declaration
9. array, read complex declaration
10. pointer, pointer and array relation, double pointer
11. malloc(memory allocation)
12. struct, union
13. struct memory layout, size of struct/union

never move onto the next topic if you do not understand pointer (number 9.)

Learn Programming Basics (C Language)

#include, data types, string literal, statement, variables

#include

- Directories are library that lies in a C Program that begins with a “#” symbol
- 2 kinds:
  - #{command} <{arguments}>,
  - #command “{arguments}”
- #include <library.h>
  - is found in the library-include in the user directories somewhere in your computer
- #include “abc.h”
  - is found in the user-defined directories,
  - default path is in the same directory as the current C file

the ‘#’ symbol is a pre-processing symbol

- There is a few things that can be done using “#”; Example:
  - #define MARCO value
  - #ifdef, #else, #endif, #elif, … etc.
- **Predefined symbol is being compile during the pre-processing stage**

Pre-Processing Stage

- Read Libraries
  - example: #include <stdio.h>
- Remove Comment
  - /* c/c++ comments */
  - // for c++ comments only

Path directory

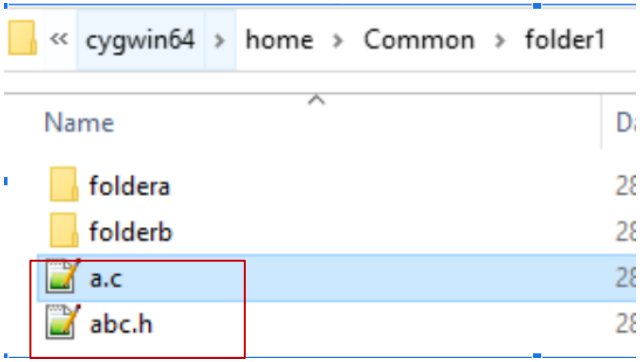
Example: Your file name is a.c in folder1

- #include “abc.h” (same path as your current file)
- #include “./abc.h” (same path as your current file)

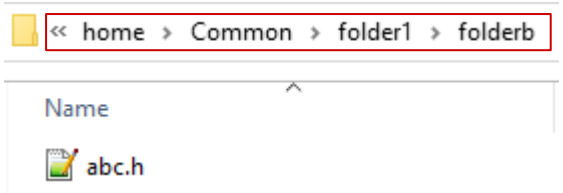
a.c

```c
#include <stdio.h>
#include "abc.h"
#include "./abc.h"

int main()
{
    printf("peko peko");
    return 0;
}
```



- #include “./folderb/abc.h” (next folder in the same path as your file)

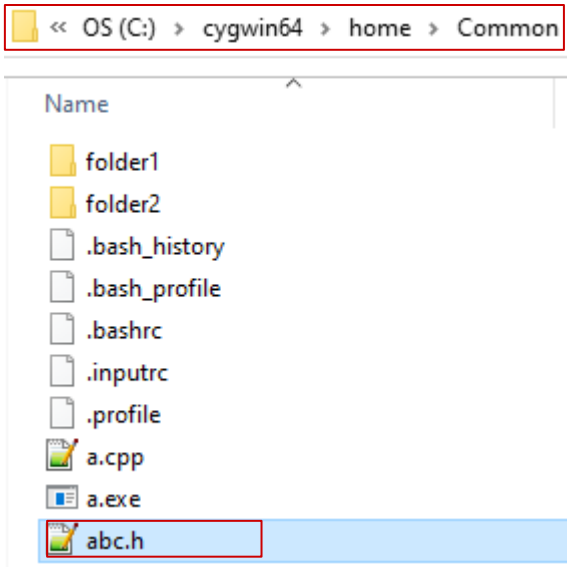


Example: Your file name is a.c in folder1

- #include “../abc.h” (the folder up, previous folder)

a.c

```
1  #include <stdio.h>
2  #include "abc.h"
3  #include "./abc.h"
4  #include "../abc.h"
5
6  int main()
7  {
8      printf("peko peko");
9      return 0;
10 }
```



Data types

built-in type

- bool (at least 1 byte)
- char (1 byte)
- short (2 byte)
- unsigned short
- int (4 byte)
- unsigned int
- float(4 byte)
- double(8 byte)
- long(at least 4 byte)
- unsigned long

1 byte = 8 bits

user-defined type

e.g. in c++

- std::string
- std::vector

char

- char is represent using a single quote (‘ ‘)
- Example:
  - ‘a’
  - ‘a’, ‘b’, ‘c’, ‘!’
  - char a = ‘a’; (initializing a variable type char)
  - char a[5] = {‘a’, ‘b’, ‘c’, ‘d’, ‘e’}; (an array of char)

char (ASCII table)

- Idea to represent each character using 8 bit size
- Readable character from #0-127
- Non-readable character from #128-255

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

String Literal

string literal

- Is whatever in between the “ “ (double quote), with un-escaped characters
  - Example:
    - “This is a string literal”
    - “This is a string\n”
    - “ “
    - “”
- contains:
  - whitespaces (e.g. space, tab)
  - char
  - escape characters (e.g. \n, \\, \$ …)
  - “\n” – next line
  - “\t” – tab

Statement

A Statement

- A statement is a line that ends with semi-colon;

Nested Statement

- A nested or compound statement is represented within a scope { … }

Variables Declaration, Initialization, Assignment

Variables

- ~~Objects that represent real world things, or ideas, simulation, representation or alias~~
- ~~TRUTH: Computers only reads 1 and 0~~
- ~~Variables are assigned with a value to give meanings to it, to represent these value~~
- a variable is a representation of a value with its type and name
- An alias

- When a program is successfully compiled and linked,
- Each declared variable is assigned to a specific memory location during RUNTIME

Identifier

- a name
- Naming of variables in programming:
- Only these characters can be used:
  - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  - a b c d e f g h i j k l m n o p q r s t u v w x y z
  - 0 1 2 3 4 5 6 7 8 9
  - _ (underscore)

- an Identifier **CANNOT begins with a number** as the 1st character:
  - 1 B-ob (compile error)
  - 345221_Bob (compile error)

Declaration

syntax:

- (dataType) variableName;
  - int a;
  - int b, c, d, e; (multiple declaration of type int)

- declared a variable with it's name and type
- declared variables without initializing contains rubbish/unknown value

Initialization

syntax:

- (dataType) variableName = value;
  - int a = 1;
  - int b = 1, c = 2, d = 3, e = 4;
  - float f = 0.01f;
  - double g = 0.1;
  - char h[] = “Haachama”;
  - char i[3]= {‘a’, ‘b’, ‘c’};
- An initial value is initialized to a variable **in the beginning**

Declaration ≠ Initialization

- int a = 1;
- int b = 1, c = 2, d = 3, e = 4;
- float f = 0.01f;
- double g = 0.1;

Assignment

syntax:

- DeclaredVariableName = value;
  - a = 2;
  - b = 10, c = 20, d = 30, e = 40;
  - f = 0.0001f;
  - g = 0.01;

- value assigned to a variable **after its has been declared or initialized**, is called assignment/assigning a value

Keywords

Reserved words

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

Example - sizeof

- sizeof is an operator, not a function
- try it out:
  - printf("sizeof bool:%d", sizeof(bool));
  - printf("sizeof char:%d", sizeof(char));
  - printf("sizeof short:%d", sizeof(short));
  - printf("sizeof int:%d", sizeof(int));
  - printf("sizeof float:%d", sizeof(float));
  - printf("sizeof double:%d", sizeof(double));