

C PROGRAMING / LINUX [DASL-100]

WEEK 1 [Section 1]

INSTRUCTOR: JEAN CHAGAS VAZ



➤ **If you have Windows machine**

- Set up the virtual box.
- Set up Ubuntu

<https://www.youtube.com/watch?v=WzbbwBEjRZY>

➤ **If want to use a compiler on Windows**

- Code::Blocks

<http://www.codeblocks.org/downloads/26>

➤ Ubuntu basic comments

.Privileges.

sudo command – run command as root
sudo -s – open a root shell
sudo -s -u user – open a shell as user
sudo -k – forget sudo passwords
gksudo command – visual sudo dialog (GNOME)
kdesudo command – visual sudo dialog (KDE)
sudo visudo – edit /etc/sudoers
gksudo nautilus – root file manager (GNOME)
kdesudo konqueror – root file manager (KDE)
passwd – change your password

.Dir Navigation.

pwd - Show directory
ls - List files in directory
cd/ - Go to root directory
*cd - Go to \$HOME directory
cd .. - Go up one directory
cd - - Go to previous directory
(cd dir && command) - Go to dir, execute command and return to current dir
cp - Copy
cp -r - Copy directory
mv - Move
rm - Remove
rm -r - Remove directory
mkdir - Create directory
man - Manual

➤ Ubuntu basic comments

```
ubuntu:~/Desktop/Imacify/Program/C in Ubuntu$ gcc -Wall varinter.c -o varint  
varinter.c:5:1: warning: return type defaults to 'int' [-Wreturn-type]  
varinter.c: In function 'main':  
varinter.c:24:1: warning: control reaches end of non-void function [-Wreturn-type]  
ubuntu:~/Desktop/Imacify/Program/C in Ubuntu$
```

File Name of
the C program

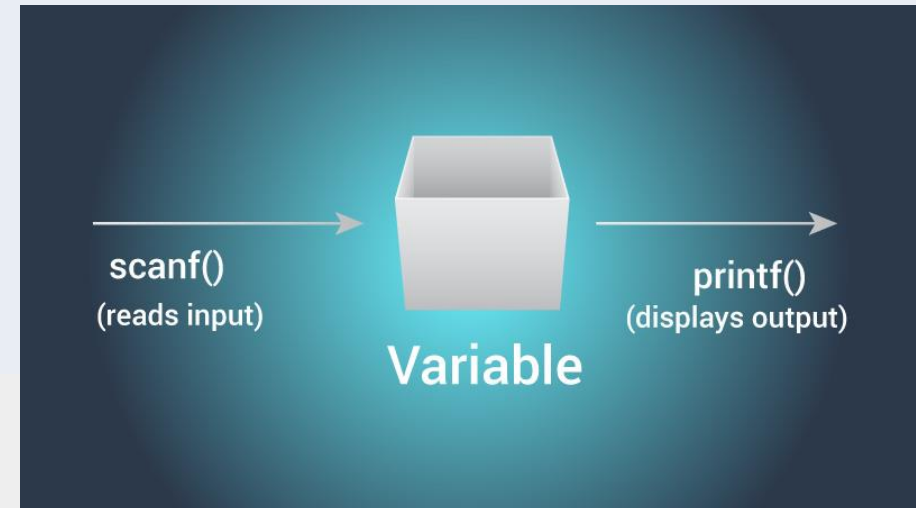
File Name for
the output file

➤ First Program “Hello World”



```
#include <stdio.h>
int main()
{
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

➤ Program to Print an Integer



```
#include <stdio.h>
int main()
{
    int number;

    // printf() displays the formatted output
    printf("Enter an integer: ");

    // scanf() reads the formatted input and stores them
    scanf("%d", &number);

    // printf() displays the formatted output
    printf("You entered: %d", number);
    return 0;
}
```

➤ Program to Add Two Integers

```
#include <stdio.h>
int main()
{
    int firstNumber, secondNumber, sumOfTwoNumbers;

    printf("Enter two integers: ");

    // Two integers entered by user is stored using scanf() function
    scanf("%d %d", &firstNumber, &secondNumber);

    // sum of two numbers in stored in variable sumOfTwoNumbers
    sumOfTwoNumbers = firstNumber + secondNumber;

    // Displays sum
    printf("%d + %d = %d", firstNumber, secondNumber, sumOfTwoNumbers);

    return 0;
}
```



Wednesday, May 24, 2017, 11:42

➤ Program to Add Two Integers



```
#include <stdio.h>
int main()
{
    double firstNumber, secondNumber, productOfTwoNumbers;
    printf("Enter two numbers: ");

    // Stores two floating point numbers in variable firstNumber and secondNumber respectively
    scanf("%lf %lf", &firstNumber, &secondNumber);

    // Performs multiplication and stores the result in variable productOfTwoNumbers
    productOfTwoNumbers = firstNumber * secondNumber;

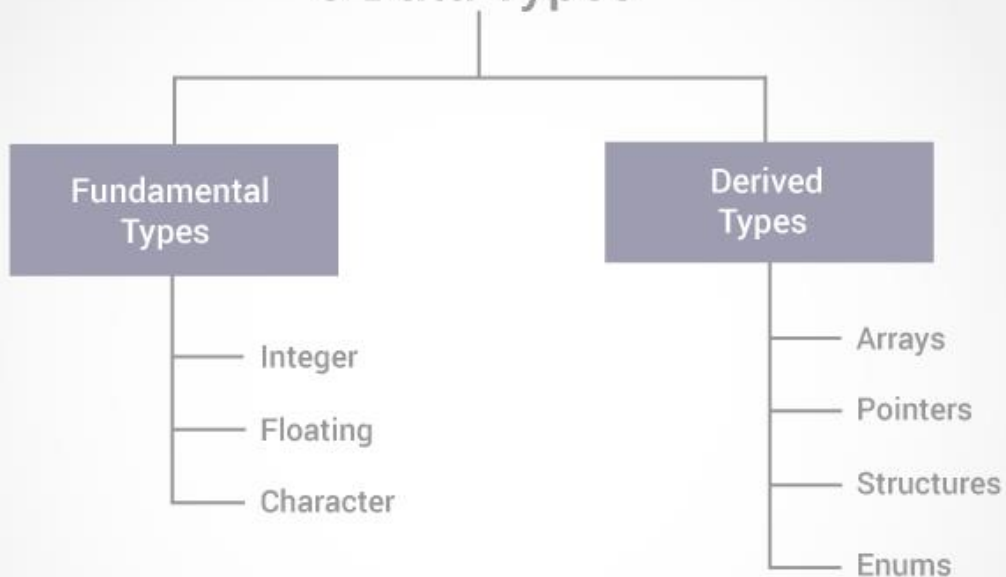
    // Result up to 2 decimal point is displayed using %.2lf
    printf("Product = %.2lf", productOfTwoNumbers);

    return 0;
}
```


➤ Program to Print ASCII Value



C Data Types





➤ Integer constants

➤ An integer constant is a numeric constant (associated with number) without any fractional or exponential part. There are three types of integer constants in C programming:

- decimal constant(base 10)
- octal constant(base 8)
- hexadecimal constant(base 16)

```
Decimal constants: 0, -9, 22 etc
```

```
Octal constants: 021, 077, 033 etc
```

```
Hexadecimal constants: 0x7f, 0x2a, 0x521 etc
```



➤ **Floating-point constants**

➤ A floating point constant is a numeric constant that has either a fractional form or an exponent form.

```
-2.0  
0.0000234  
-0.22E-5
```

➤ A character variable holds ASCII value (an integer number between 0 and 127) rather than that character itself in C programming. That value is known as ASCII value.

For example, ASCII value of 'A' is 65.

What this means is that, if you assign 'A' to a character variable, 65 is stored in that variable rather than 'A' itself.



➤ A character variable holds ASCII value (an integer number between 0 and 127) rather than that character itself in C programming. That value is known as ASCII value.

For example, ASCII value of 'A' is 65.

What this means is that, if you assign 'A' to a character variable, 65 is stored in that variable rather than 'A' itself.



```
#include <stdio.h>
int main()
{
    char c;
    printf("Enter a character: ");

    // Reads character input from the user
    scanf("%c", &c);

    // %d displays the integer value of a character
    // %c displays the actual character
    printf("ASCII value of %c = %d", c, c);
    return 0;
}
```

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

➤ To do List

- Start Homework 1
- Create a hand made layout explaining the Differences between the “C Data Types”[scan it in a pdf format and send it to me (U can use the Lab’s printer to scan direct in pdf format)] use the figure bellow as reference. (**DUE NEXT SECTION**)

