

Lecture-2

Basic structure of a C program:

1. Including header files
2. Macro definition
3. Global declaration
4. Main declaration
 - {
 - local declaration
 - statement sequence
 - other function call
 - }
5. User defined function

Statements are parts of your program that actually performs operation. Statements are contained within functions.

All C statements end with a semicolon. C does not recognize the end of the line as a terminator. This means there are no constraints on the position of statements within a line. Also you may place two or more statements on one line.

The standard library contains functions to perform I/O, string manipulation, mathematics and much more.

So, we can start our first C program that will print “This is our first C program” to the standard output device.

```
#include<stdio.h>
void main(void)
{
printf(“This is our first C program”);
}
```

1. **include is a pre processor directive** that tells the compiler to **include library functions contained in stdio.h** file with your program.
2. main function
3. { beginning of main function
4. library function printf, that prints string within “ “
5. } end of a function.

Character that can be used in C:

1. **Alphabets:** 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.
2. **Digits:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
3. **Special symbols:** ~ ` ! @ # \$ % ^ & * () _ - + = | \ { } [] ; : " ' < > , . ? /

White Space: blank, new line, tab.

5 types of tokens in C: keywords, identifiers, constants, operators & punctuators.

Keywords: Keywords are specially reserved words that have strict meaning as individuals token in C.

ANSI standard: 32 keywords.

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

Additional keywords like asm etc. are in turbo C.

Identifiers: Identifiers are user defined name of objects in a program. An identifier is composed of a sequence of letters, digits and underscores (_). An identifier must begin with a letter or underscore. Ex: sb_int, cse_aust, s_99, _x11.

Constants: C constants can be divided into 2 major categories.

1. **Primary** constants.
 - a. **Integer** constants: 123, +234, -876 etc.
 - b. **Real** constants: 23.4, -45.6, 0.56, 4.2e7, 9.3e-4, -8.1e4, -2e-4 etc.
 - c. **Character** constants: 'A', 'b', 'x', 'S', '&' etc.

IntRC

* Every keyboard key is a character constant.

2. **Secondary** constants.
 - a. **Array**
 - b. **Pointer** APSUE
 - c. **Structure**
 - d. **Union**
 - e. **Enum**