Building Blocks of C Programming Language

By Anand George

6 mainly

- All C program mostly contains following 6 constituents.
  - Keyword
  - Identifier
  - Literal
  - Comments
  - Operators.
  - Other keyword like tokens and aliases

Keywords

- Words which complier know the meaning.
- Like a word in dictionary.
- Very few in C.
- More keywords which are specific to compilers

Mostly used keywords in C

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

Demo Keywords

- Some Keywords
- int, void

Keyword

- It is required to know and understand the use of each keyword.
- Not that easy to get all in hand in the beginning.
- We will see one by one as we go.

Identifier

- Name of something in the program.
- Compiler wont understand them before introducing them with keywords.
- Like a name of a person which normal not in the dictionary.

Demo Identifier

- int a;

Literals

- Text and numbers normally.
- Text will be in double quotes - “”;
- Number are like 10, 0xAB (hex) etc.

Operator

- + - * / % etc
- Some are bit confusing for beginners as mean of the operator kind of different from math.
- Example = is not "equal" but assignment.

Operator Demo

- + plus operator sample

Comment

- Not compiled by the compiler
- Mostly for the readability of code
- /* anything – even more than one line */
- // single line file should be .cpp

Comments demo

- Some comment in C

Others

- Mostly understood by the compiler mostly behave exactly like keywords from programmer standpoint.
- Example
  - Preprocessor statement like #define, #include #pragma
  - { } braces,
  - ; semi colon
  - Complier specific reserved keyword like __declspec is a reserved keyword ONLY in Microsoft C complier.

Other (Aliases)

- One of the items we already discussed but the name will be different.
- Just kind of old wine in new bottle.
- Nothing new but causes a lot of confusion to beginners looking at production code.
- Too many in production code.
- #define (pre processor definitions)and typedef is mostly for implementing this.

Alias or place holders

- #include <stdio.h> place holder for contents in the file stdio.h

Example 1

```
#include <stdio.h>
void main ()
{
 printf (“hello world”);
}
```

Example 2

```
// helloworld.cpp : Defines the entry.
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
 return 0;
}
```

Conclusion

- Building blocks of the C.
- Over all discussion with out much details like preprocess directives, typedef etc.
- Don’t worry if you are not able to understand completely at this point in time.

Thank you