C Programming Short Notes

Part 1

1. Tell me about C Language?

- C programming is a **general – purpose, procedural, imperative** computer programming language developed in **1972** by **Dennis M. Ritchie** at the **Bell Telephone Laboratories** to develop the **Unix Operating System**
- The language has been given the name **C** because it succeeds another language called **B.**

2. Feature of C Language:-

- Simple
- Machine independent or portable
- Mid – level programming language
- Structured programming language
- Rich Library
- Memory management
- Fast speed
- Pointers
- Recursion
- Extensible

3. Advantage of C?

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low level activities
- It can be compiled on a variety of computer platforms.

4. Commends in C?

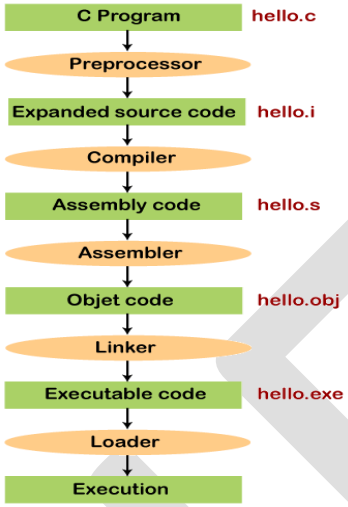
Comments are used to indicate something to the person reading the code comments are treated like a **blank by the compiler and do not change anything in the code's actual meaning.** There are two types of comments:

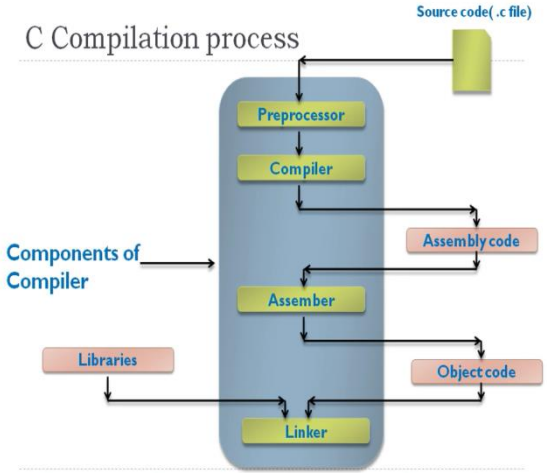
- Single line comments => //hii rockers…… ☺
- Multi line comments => /*Ready to
  Rock*/

5. C Compiler

- The Source code written in source file is the human readable source for your program. It need to be "compiled", into machine language so that your CPU can actually execute the program per the instructions given.
- The compiler compiles the source codes into final executable programs.

6. Compile and execute c Program

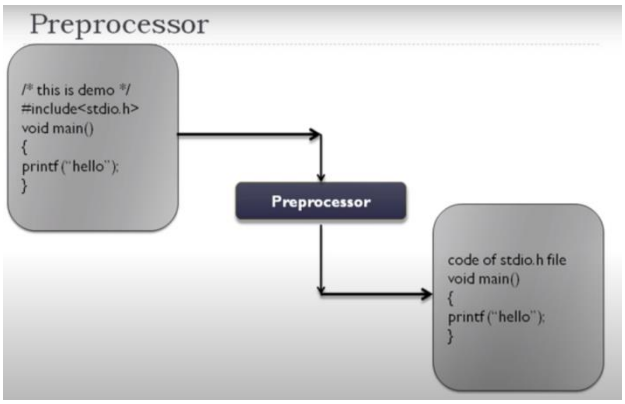




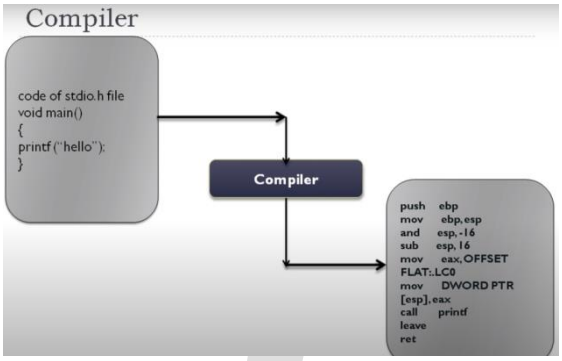
The compilation process in C involves several stages that translate **human-readable C source code into machine-executable code.**

Preprocessor: The preprocessor scans the source code and handles preprocessor directives, such as #include and #define, to prepare the code for compilation.

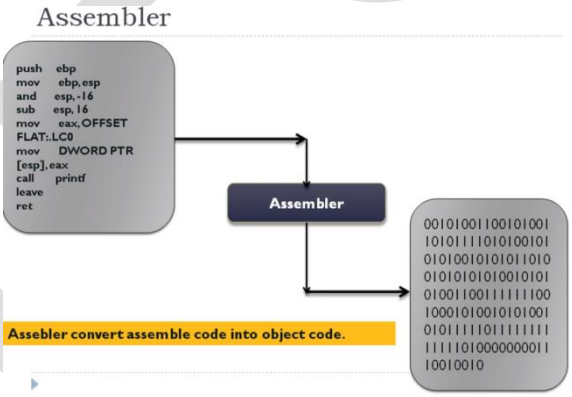
- Removal of comments
- Expansion of macros
- Expansion of the include files
- Conditional compilation



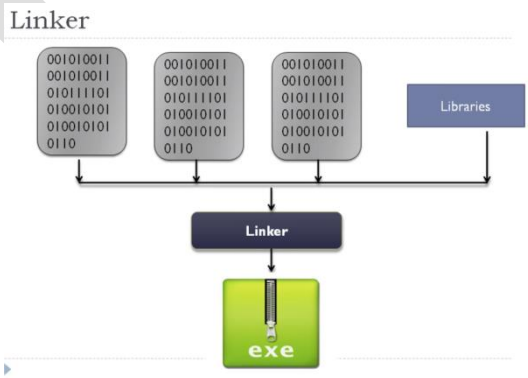
Compiler: The compiler translates the preprocessed source code into assembly language or an intermediate representation. It performs syntax and semantic analysis to detect errors and generates intermediate code.



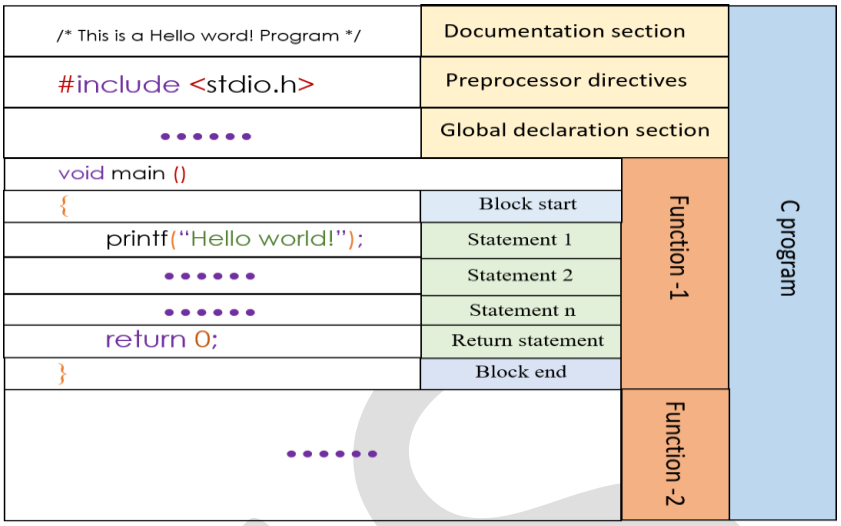
Assembler: The assembler translates the assembly code into machine language instructions (object code). It resolves addresses and generates relocation information. For linux .o and windows.obj.



Linker: The linker combines multiple object files and resolves external references to create an executable file. It also links in any necessary libraries and resolves symbols. For linux a.out and windows .exe.

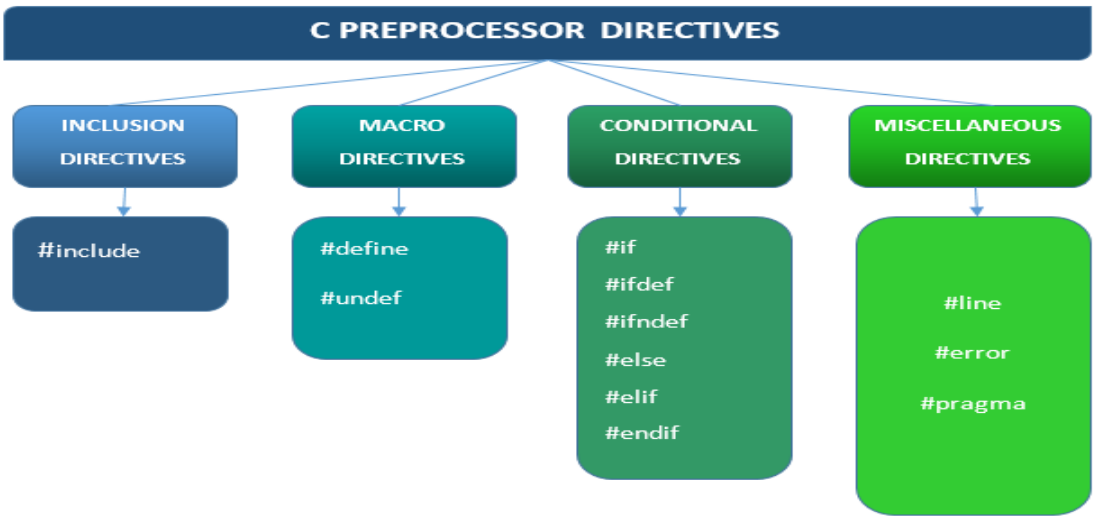


7. Structure of c program?



Preprocessor directives

- Preprocessor directives **tell the compiler to preprocess the source code before compiling**.
- It is a step prior to the compilation.
- It begins with **hash ('#')** symbol.



8. Variables in c?

- A variable is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times.

Exg:- int a;
char b;
float c;

9. Variable Declaration:

- Variable declaration is the process of specifying the data type and name of a variable without allocating memory to it.
- It informs the compiler about the existence of a variable and its data type, allowing the compiler to reserve space for it during compilation.

Syntax: data_type variable_name;

Eg:- int a, b, c;

10. Variable Definition:

- Variable definition is the process of declaring a variable and allocating memory space to it.
- It reserves storage space in memory for the variable, allowing it to store values.

Syntax: data_type variable_name = value;

Eg:- int a=100;

11. Rules of Variables Name Declaration

- Combination of alpha, number, and ('_') underscore.
- Cannot start with number.
- Case sensitive
- Whitespace is not allowed inbetween variables

eg:-

```
ab_c     //valid
ab_12    //valid
_abc     //valid
Auto     //valid
abc12    //valid

1abc     //invalid
abc#@    //invalid
a b      //invalid
auto     //invalid
```

12. Lvalue and Rvalue

- Lvalue ->Left hand side -> **L means location** -> **Variables(memory location)**
- Rvalue ->Right hand side -> **R means reading** -> **Data values**

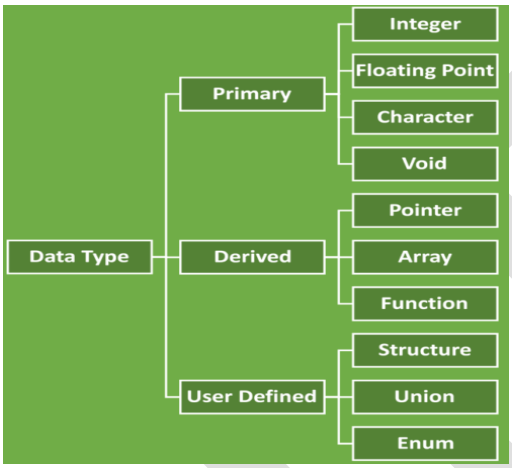
eg:-

```
a = 10    //valid
a = b     //valid
a = a+b   //valid

10 = a     //invalid
a+b = 10  //invalid
```

13. Data type in C?

Data types in C are used to specify the size and type of data that can be stored in a variable



Format Specifier	Type
%c	Character
%d	Signed integer
%f	Float values
%i	Unsigned integer
%l or %ld or %li	Long
%lf	Double
%Lf	Long double
%lu	Unsigned int or unsigned long
%lli or %lld	Long long
%llu	Unsigned long long

Datatypes	Memory Size	Range
char	1 byte	-128 to 127
unsigned char	1byte	0 to 255
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
long	8 bytes	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615
float	4 bytes	1.2E-38 to 3.4E+38 6 decimal places
long	8 bytes	2.3E-308 to 1.7E+308 15 decimal places
Double	10 bytes	3.4E-4932 to 1.1E+4932 19 decimal places

The size of an integer data type is compiler-dependent, and you can use the **sizeof operator** to check the actual size of any data type.