Introduction to C programming

C is a procedural programming language initially developed by Dennis Ritchie in the year 1972 at Bell Laboratories of AT&T Labs. It was mainly developed as a system programming language to write the UNIX operating system.

The main features of the C language include:

- General Purpose and Portable
- Low-level Memory Access
- Fast Speed
- Clean Syntax

- Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on the C language. C++ is nearly a superset of C language.
- Learning C will help to understand a lot of the underlying architecture of the operating system. Working with memory locations

Structure of C program

By structure, it is meant that any program can be written in this structure only. Writing a C program in any other structure will hence lead to a Compilation Error.

First C Program

```
#include <stdio.h>

int main() {

char a = “R” ;

printf("%c", a);

return 0;

}
```

Components of a C program

1. Header File Inclusions - Preprocess
2. Main Method Declaration - Entry Point
3. Body Of Main Method - define the function
4. Statements - instructions to compiler
5. Return Statement - return value of return type

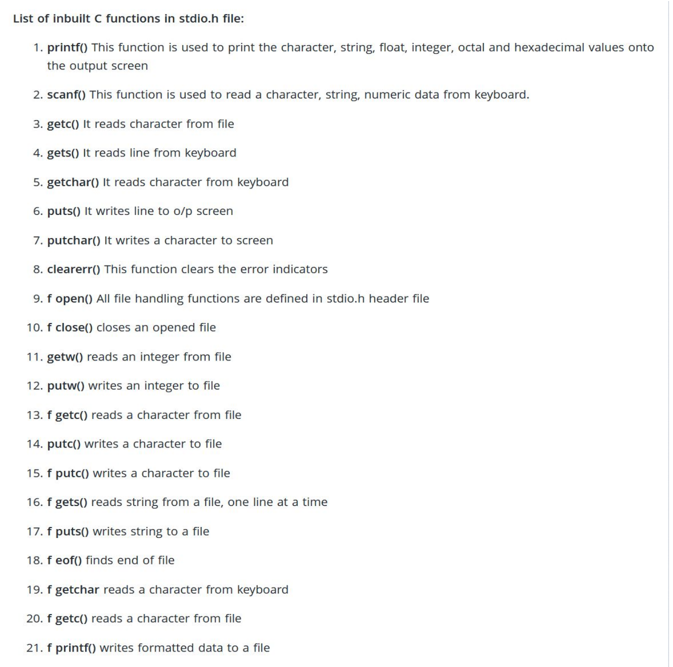
What is a Compiler?

- **A special program that translates a programming language's source code into machine code, bytecode or another programming language**.
- The source code is typically written in a high-level, human-readable language such as C or C++.

What is a preprocessor?

The **C Preprocessor** is not a part of the compiler, but is a separate step in the compilation process.

In simple terms, a C Preprocessor is just a text substitution tool and it instructs the compiler to do required pre-processing before the actual compilation.



List of inbuilt C functions in stdio.h file:

1. **printf()** This function is used to print the character, string, float, integer, octal and hexadecimal values onto the output screen
2. **scanf()** This function is used to read a character, string, numeric data from keyboard.
3. **getc()** It reads character from file
4. **gets()** It reads line from keyboard
5. **getchar()** It reads character from keyboard
6. **puts()** It writes line to o/p screen
7. **putchar()** It writes a character to screen
8. **clearerr()** This function clears the error indicators
9. **f open()** All file handling functions are defined in stdio.h header file
10. **f close()** closes an opened file
11. **getw()** reads an integer from file
12. **putw()** writes an integer to file
13. **f getc()** reads a character from file
14. **putc()** writes a character to file
15. **f putc()** writes a character to file
16. **f gets()** reads string from a file, one line at a time
17. **f puts()** writes string to a file
18. **f eof()** finds end of file
19. **f getchar** reads a character from keyboard
20. **f getc()** reads a character from file
21. **f printf()** writes formatted data to a file

What are data types?

- Data types are the type of data stored in a C program.
- Data types are used while defining a variable or functions in C. It’s important for the compiler to understand the type of predefined data it is going to encounter in the program.
- In general terms, a data type is an attribute that tells a computer how to interpret the value.

Types of Data Types

- Primary data types: Integer, Character, Floating point , Double floating point, Void
- Derived data types: Function, Array, Pointer, Reference
- User Defined data types: Class, Structure, Union, Enum, TypeDef

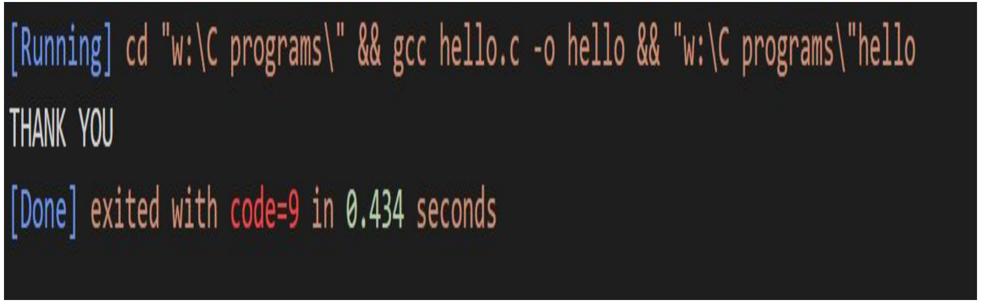
DATA TYPE	MEMORY (BYTES)	RANGE	FORMAT SPECIFIER
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	Same as int	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
Signed char	1	-128 to 127	%c
Unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3,4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf

Stack memory and Heap memory

- Allocation of memory
- Run-time
- Compile-time

Run different codes to get ->

Unique and different outputs —>



```
[Running] cd "w:\C programs\" && gcc hello.c -o hello && "w:\C programs\"hello
THANK YOU
[Done] exited with code=9 in 0.434 seconds
```