

Qn. What is the symbol of # called and why do we write #include ?

Ans: In c and c++ lang , the symbol of # is called as POUND and any statement in our code which begins with # is called as PRE PROCESSOR DIRECTIVE.

These PRE PROCESSOR DIRECTIVES are special instructions which are not handled by the compiler rather they are handled by another special software built into compiler called as PRE-PROCESSOR

This pre-processor software reads our program before the compiler reads it and handles only those lines which begin with pound (#) . Following are the list of popular pre-processor directives handled by the pre-processor software.

1. #include : called as **file inclusion directive**
2. #define : called as **macro creation directive**
3. #undef : called as **macro removal directive**

4.

- a) #if
- b) #else
- c) #elif
- d) #endif
- e) #ifdef
- f) #ifndef

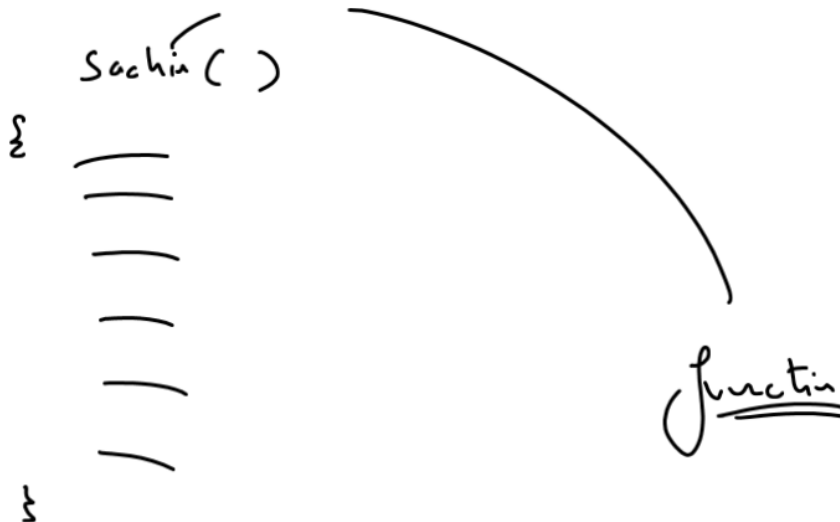
Common name: **conditional compilation directive**

Amongst all the above pre-processor directive the most popular is the **#include** directive which is also called as file inclusion directive and it is used by programmers for adding header files into their programs

Whenever the pre-processor reads the **#include** statement in our program it does the following

1. It reads the name of the header file mentioned in <> brackets
2. It copies the entire coding of the header file and paste's it in our program in place of the **#include** statement

Due to this no of lines in our program are increased and a seperate copy of our code gets generated called as **EXPANDED SOURCE CODE** . This expanded source code is then compiled by compiler and converted into **MACHINE CODE**



Q. What are functions and what is the importance of main()?