**Preprocessor: What is pre-processor, Types of Pre-processor, Macros, File Inclusion, Conditional Compilation.**

### What is pre-processor

The C preprocessor is a **macro processor** that is used automatically by the C compiler to transform your program before actual compilation.

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. All preprocessor commands begin with a hash symbol (#).

### Types of Preprocessor Directives:

1. Macros
2. File Inclusion
3. Conditional Compilation

### Macros

A macro is a piece of code in a program that is replaced by the value of the macro. Macro is defined by **#define** directive. Whenever a macro name is encountered by the compiler, it replaces the name with the definition of the macro. Macro definitions need not be terminated by a semi-colon(**;**).

Syntax:

```
#define name substitute text
```

### Example

```
#define N 50
void main()
{
    int j;
    for (j=1;i<=N; j++)
    {
        printf("\n%d", j);
    }
}
```

### Example: Macros with Arguments

```
#define AREA(a) (5.18 * a * a)
void main()
{
    float r = 3.5, x;
    x = AREA (r);
    printf ("\n Area of circle = %f", x);
}
```

```
#include <stdio.h>
// Macro definition
#define AREA(l, b) (l * b)
int main()
{
   int l1 = 10, l2 = 5, area;
   area = AREA(l1, l2);
   printf("Area of rectangle is: %d", area);
   return 0;
}
```

**File Inclusion**

This type of preprocessor directive tells the compiler to include a file in the source code program. The type of files that can be included by the user in the program is
**Header files or Standard files**: These files contain definitions of pre-defined functions like **printf(), scanf(),** etc. Example stdio.h, string.h..

**Syntax**:
#include< *file_name* >
where *file_name* is the name of the file to be included. The ‘<‘ and ‘>’ brackets tell the compiler to look for the file in the standard directory.

**Conditional Compilation.**

Conditional Compilation directives are a type of directive that helps to compile a specific portion of the program or to skip the compilation of some specific part of the program based on some conditions. This can be done with the help of the two preprocessing commands ‘**ifdef**‘ and ‘**endif**‘.
**Syntax**:

```
#ifdef macro_name
   statement1;
   statement2;
   .
   statementN;
#endif
```

If the macro with the name ‘*macro_name*‘ is defined, then the block of statements will execute normally

```
include <stdio.h>
#define NUMBER -2
int main() {
#if (NUMBER>0)
printf("Value of Number greater than 0 is: %d",NUMBER);
#else
printf("Value of Number less than 0 is: %d",NUMBER);
// #endif is used to determine the end the if condition.
#endif
return 0;
}
```