THAT’S ALL YOU GOT IN C++

1. C++ header files

Preprocessor directives - these get processed before the compilation of the source code.

In C++ it’s not necessary that every header file ends with .h extension but it’s a must in case of C.

The preprocessor directive #include directs the compiler to process these files before the execution of the source code.

Standard library header files - #include<stdio.h> or #include “iostream”

User defined header files - #define limit 5

1. Long double is most accurate
2. Rounding Off

a. floor(x) – rounds off the value of x to closest integer which is less than the given value. (1.71—1)

b. celi(x) – rounds off the value of x to the closest integer which is greater than the given value. (1.42 – 2)

c. trunc(x) – removes all the digits after decimal point

d. round(x) – rounds the number to closest integer.

1. Precision – sets the precision correct to decimals,  cout << fixed << setprecision(2)<<x
2. Operators
3. [String operations](strings.cpp)
4. [Functions](functions.cpp)

In case of call by value the value of the original caller may change if changes are made in the function but if we pass the variable by reference there is no change in the original variable even if some operation is done on it in the function.

1. [Return statement vs exit statement](https://www.geeksforgeeks.org/return-statement-vs-exit-in-main/)

In case of exit(0) destructor is not called and the program exits while in case of return(0) the destructor is allowed to work.

1. Floating point arithmetic is so silly –

double a=0.1, b= 0.2 , c=0.3;

*if*(a+b==c) cout<<"not executed!";

*else* cout<<"executed!";

1. Sizeof(arr)/sizeof(arr[0]) – is the limit of iterations
2. Static 🡪 If you declare a variable with static keyword then it is initialized to 0. (\*variable next to it is initialized to 1).
3. stoi(str) 🡪 it returns the integer value of a string
4. If
5. K