

```

1  #include <iostream>
2  //! to delete the prefix (std::) that use with cin & cout & endl use this statement after includes -->
3  using namespace std;
4  /** this statement will add std statement for all place in implecit way
5  /* !!nameSpace--->
6      ~ is a virtual group use to organize code
7      # in a way that get group of code and put it in one place
8      % and give it a name like namespace then use it to organize code
9  */
10 int main()
11 {
12     /** we use nameSpace to reduce the line of code
13     /**/ std::cout << "Hello World!\n"<< std::endl;
14
15     /** std:: is prefix i can delete it
16         |         /** by add using namespace std; after #include statement
17
18     /**~ so atfer use the statement i can write cout / cin / endl without use std::
19     cout << "hi this line without std:: ";
20
21     return 0;
22 }
23 /*
24 % to select part code in multiple line -->
25 select the part in one line then choose --> alt+ move the mouse */
26 /*
27
28 ! understand \n or endl ?? which is better ??
29     |         # we have something called Buffer flushing
30
31     !\n -->
32     ? before the output appear it use buffer and
33     ? put all statement inside it then put it on the screen
34     ? then do Buffer flushing
35     ~ note --> the Buffer should not contains many data if it contanis many data it will be very slow
36
37     ! endl -->
38     ? every statement put in Buffer then do flushing from the Buffer
39     ? so it do flushing after every statement
40
41
42     ! the best don't use one of them just ake mix between them
43     //ex if i have 1000 statement suplit it after 49 \n use one endl
44
45     % every programming language use Buffer
46     ~ it is like table to store data in queue way and use it
47     ~ It is used when executing the program and is very important for performance
48 */
49

```

```

50  /*
51
52  ! Note --> When I use the Comments less will be better, and less amount of should be used
53  #Comments in the program
54  ?comments :
55  ? comments can be used to explain c++ code , and to make it more readable
56  // type of comments -->
57      ~      1) single line (for short comments) --> //
58              it start with two forward slashes
59              any text between // and the end of the line is ignored
60              by the compiler (will not be executed )
61      ~      2) multi line (for long comments) /* * /
62              it staet with (/ *)    and end with (* / ) any text
63              between (/ * and * / ) will be ignored by the compiler
64  */

```