How to move from C to C++?

- Header Files
    - In C, you generally use the header files
        - stdio.h
        - string.h
        - math.h
    - In C++, you generally use the header files
        - iostream
        - cmath
        - cstring
        - <Other STL libraries>
    - In C++, unlike C, you don’t need to include a bunch of header files. Instead you can include only one header file which will include most of the required header files
        - bits/stdc++.h

      Caution: Using bits/stdc++.h may slow down compilation.

- Input Output functions
    - In C, the basic input output functions are scanf, printf, gets, puts.
    - In C++, you use cin, cout.

- For a new line, in C we use ‘\n’ in printf statement while we can use **endl** in cout statement in c++. **Caution**: endl slows down output, so better use ‘\n’.
- To input a line i.e. to input a string with spaces we use gets in c while getline in C++.
- All the functions like cin, cout etc. are defined in a standard namespace. So, instead of directly using it we usually add a namespace to avoid writing std every time. You do not need to understand much about namespace. Just remember to add a statement
    - **“using namespace std;” (without quotes)**

  At the top in all C++ codes.

See the table for better understanding.

C	Cpp	Explanation

#include<stdio.h>,
#include<math.h>
....
	#include <bits/stdc++.h>	You need to include only one header file

No namespace statement	using namespace std;	Namespace std is used in cpp

int a;
scanf(“%d”,&a);
printf(“%d,a);
	int a;
	cin>>a;
	cout<<a;
		Use of cin, cout in place of scanf and printf

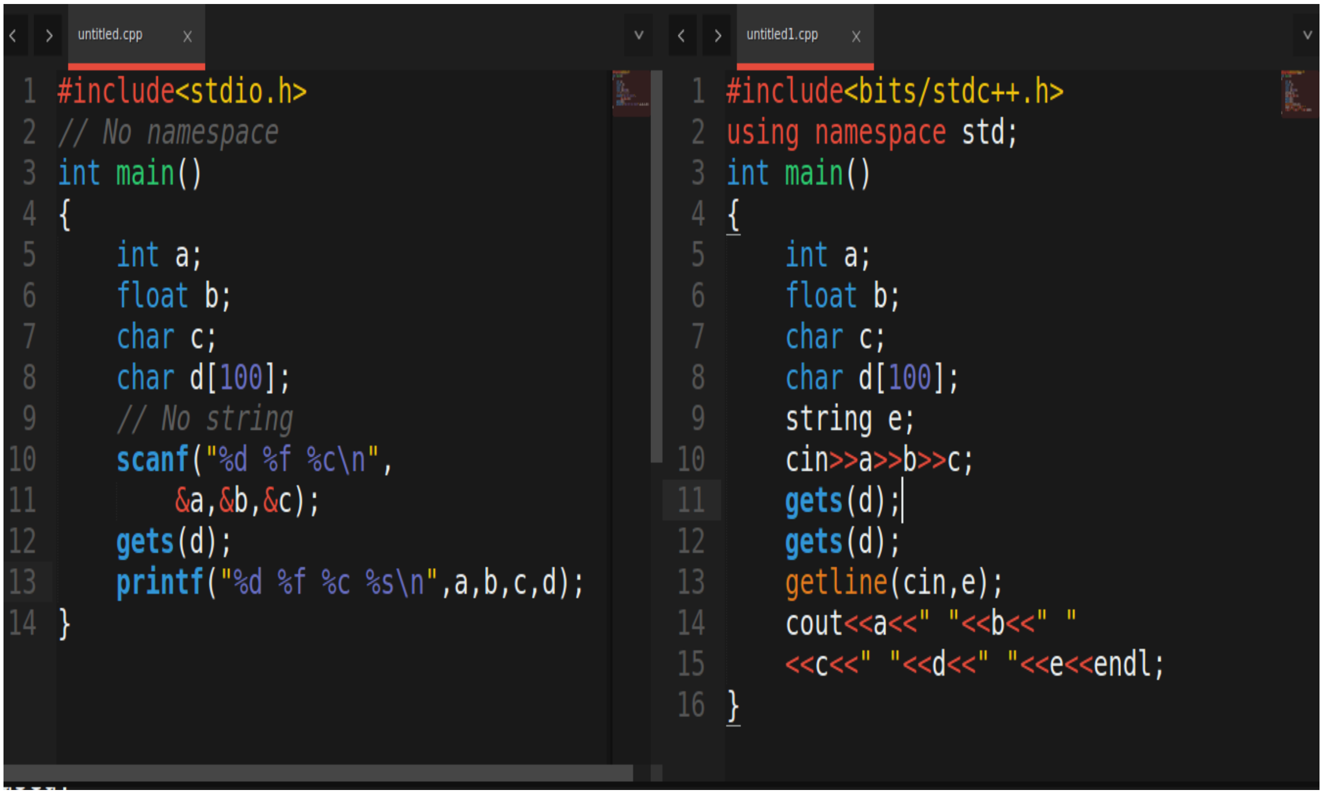
int a;float b;char c;
scanf(“%d%f%c”, &a,&b,&c);
printf(“%d%f%c”, a,b,c);
	int a;float b;char c;
	cin>>a>>b>>c;
	cout<<a<<b<<c;
		Multiple input output at the same time

printf(“%d\n”,a);	cout<<a<<endl;	To print some variable a followed by a newline character

char a[100];
scanf(“%s”,a);
	char a[100];
	cin>>a;
		To input a string

char a[100];
gets(a);
	stri ng a;
	getline(cin,a);
		To input a line
		Note - If you make a character array, you will have to use gets in cpp as well.

See the codes for better understanding.



```
1  #include<stdio.h>
2  // No namespace
3  int main()
4  {
5      int a;
6      float b;
7      char c;
8      char d[100];
9      // No string
10     scanf("%d %f %c\n",
11         &a,&b,&c);
12     gets(d);
13     printf("%d %f %c %s\n",a,b,c,d);
14 }
```

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main()
4  {
5      int a;
6      float b;
7      char c;
8      char d[100];
9      string e;
10     cin>>a>>b>>c;
11     gets(d);
12     gets(d);
13     getline(cin,e);
14     cout<<a<<" "<<b<<" "
15     <<c<<" "<<d<<" "<<e<<endl;
16 }
```

Do note the use of two gets in cpp code (line 11) and '\n' in scanf statement in C code (line 10). This is because when you press enter after taking a, b, c as input '\n' gets stored in d. So, we need to input d again. In C code, we already take \n as input in scanf statement. So we don't need to use gets. If we do not use \n in scanf, then we will have to use two gets. This might be a little bit confusing but you shall understand it as you practice.

Another very important thing that needs to be mentioned is the fact that cin cout are very slow as compared to scanf printf.

Therefore, you should never synchronize C++ streams to standard C streams.

```
ios_base::sync_with_stdio(false);
cin.tie(0);
cout.tie(0);
```

This makes cin, cout work comparatively faster. It is advised to add this statement at the top of your main function in all your codes. Although this is still slower as compared to scanf, printf. Therefore, sometimes you will be forced to use scanf, printf in place of cin, cout when input or output is quite large. Infact, at some occasions even scanf won't help. So you will have to use fast I/O. You can read about them here**.** Although in most problems unsynchronized cin cout will work fine.