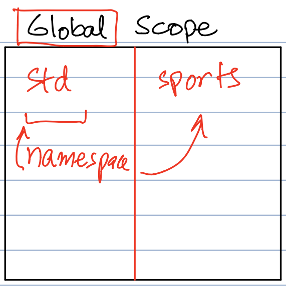
Global Scope



```
#include <iostream>
using namespace std;

int main () {
std:: cout << " Hello world " << endl;
    return 0;
}
```

std:: cout ← console output

sports:: cout ← captain of ——

```
 X ———————— X
    string               string
int  max  (int a,  int b) {
     // return the max of the two
}
```

" Usman " > " Ali "

placeholder

```
template <class T>
T    max (T a,  T a) {
        ═══
}

max<int>( 5,  6 )

max <string>( "Usman",  "Ali" )
```

function template

"generalized version of a function that can be "specialized" later ?

Student List

```
template <class T>
class List {
    struct node {
T  ←  int val;  → student               student
      ≡
T  ←  int pop();          void push(int in);
         → student
};
```

```
List<int>       l;
List<Student>   students;
```

"class template" ——

generics ——