Chapter 11.2: Namespaces

Namespace: a region of name definitions, such as class definitions and variable declarations. Namespaces are used to organize code into logical groups and to prevent name collisions that can occur when your code base includes multiple libraries.

Most C++ library files (such as iostream) place name definitions (such as cout, cin) into the standard namespace. Hence, CS111 required you to place the two statements on the top of every .cpp file:

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

To place some code in a namespace, place it in a namespace grouping.

```
namespace <Name> {
        … // place name definitions here
}
```

For example:

```
namespace Space {
        void hello() {
                std::cout << “Hello!\n”;
        }
}
```

There are three ways to use the hello() function from the Space namespace:

using namespace Space;
 	
int main() {
    hello();
}
	using Space::hello;
	
	int main() {
	    hello();
	}
		int main() {
		    Space::hello();
		}

Names defined in the **unnamed namespace** are local to the file. It cannot be accessed outside of the file and will not cause a naming conflict.

```
namespace {
        …
}
```

The unnamed namespace is different from the **global namespace**, where no namespace is declared at all.