SFU CMPT-212 2008-1 Topic: Namespaces

Ján Maňuch

E-mail: jmanuch@sfu.ca

Wednesday 6th February, 2008

Namespaces

Example:

```
1 std::cout <<i;
2 using namespace std;
3 using std::cout;
```

Q. How to create and add functions, variables, etc into a namespace?

```
1 namespace name_of_namespace {
2   declarations and definitions
3 }
```

Comments:

- namespace cannot appear inside of a block: can be located at the global level or inside other namespaces
- namespace can be reopened many times, each time adding new declarations and definitions

Example:

```
1  namespace CMPT {
2    const int nstudents=58;
3    float grades[nstudents];
4    float add(int,float);
5    struct Student { ... };
6  }
7
8  namespace CMPT {
9    float add(int sn,float incr) {
10     grades[sn] += incr;
11     return grades[sn];
12   }
13 }
```

Accessing names in a namespace

- using the *scope resolution operator* `::`

  Examples:

```
1 CMPT::Student Fox;
2 CMPT::add(CMPT::nstudents-1,7.5);
```

  Note. `::` with omitted namespace name will access the global scope of the file:

```
1 int n=5;
2 int main() {
3   int n=6;
4   cout << n;   // print 6
5   cout << ::n; // print 5
6 }
```

- using `using`-declarations — declares a name in the local scope

  Example:

```
1  using CMPT::Student;
2  Student Fox;
3  {
4    using CMPT::nstudents;
5    cout << nstudents; // ok
6    int nstudents;     // error:
7                       //   nstudents already defined
8  }
9  cout << nstudents;   // unknown variable
```

- using `using`-directive — adds all names to the local scope as global names (no name conflicts — local names just hide global names!)
  Example:

```
1 {
2   using namespace CMPT;
3   cout << nstudents; // ok, prints 58
4   int nstudents=20;  // new local variable is created,
5                      // hides CMPT::nstudents
6   cout << nstudents; // prints 20
7 }
8 cout << nstudents;   // unknown variable
```

- nested namespaces

```
1 namespace A {
2   namespace B { int c; }
3 }
4 A::B::c=2;
```

Sample questions

- Declare a namespace called `CMPT` containing global variable `students` of type `int`, a structure `Student` (containing at least 3 data members according to your choice) and a function with a header `void Print(const Student &s)`.

  Show *three* different ways how to access a variable `students` from the namespace `CMPT` in a program.

- Which of the following lines will generate compiler error?

```
1  namespace Test {
2    int count;
3  }
4  int main() {
5    {
6      using Test::count;
7      int count;
8    }
9    {
10     using namespace Test;
11     int count;
12   }
13 }
```