Ch1. Namespace

Jong-Hyeok Park
akindo19@gmail.com

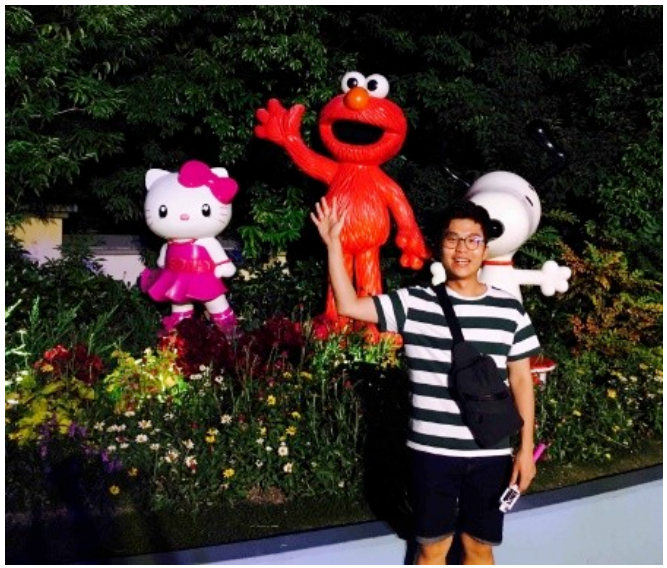
SUNGKYUNKWAN UNIVERSITY 1398 성균관대학교

VLDB Lab. @ SKKU
http://vldb.skku.ac.kr

Who am I ?

Jong-Hyeok Park

- Ph.D student @ SKKU (VLDB Lab.)
- Research Interests
  - Flash-based/Pmem-aware DBMS Tech.
  - Cloud computing
- Semi-vaccinated (Pfizer)
- Hobbies
  - Netflix, Camping, Drinking



https://github.com/JonghyeokPark

https://www.linkedin.com/in/jonghyeokpark/

Namespace (1)

- Scope resolution operator **::**

ch1 — scope.cc + (~/Dropbox/VLDB/공부/...l-cpp-study-jhpark/slides/ch1) - VIM — vim scope.cc...

vim

```
1 #include <iostream>
2 int count = 0;
3
4 int main() {
5   int count = 0;
6   ::count = 1; // set global variable
7   count = 2;   // set local variable
8 }
9
~
~
~
~
~
~
~
~
~
scope.cc [+]
```

Namespace (2)

- **Question**: How to handle duplicate function in namespace

```
#include <cmath>

#define CEILING_POS(X) ((X-(int)(X)) > 0 ? (int)(X+1) : (int)(X))
#define CEILING_NEG(X) ((X-(int)(X)) < 0 ? (int)(X-1) : (int)(X))
namespace skku {

int ceil(double num) {
  std::cout << "our own version!\n";
  return ( ((num) > 0) ? CEILING_POS(num) : CEILING_NEG(num) );
}

void guess() {
  std::cout << "test ceil: " << std::ceil(6.5) << std::endl;
  std::cout << "test ceil: " << ::ceil(6.5) << std::endl;
  std::cout << "test ceil: " << ceil(6.5) << std::endl;
}

namespace vldb {
  void guess2() {
    std::cout << "test ceil(4.5): " << std::ceil(4.5) << std::endl;
    std::cout << "test ceil(4.5): " << ::ceil(4.5) << std::endl;
    std::cout << "test ceil(4.5): " << skku::ceil(4.5) << std::endl;
  };
}; // namespace vldb
}; // namespace skku

int main(void) {
  skku::guess();
  skku::vldb::guess2();
  return 0;
}
```

https://gist.github.com/JonghyeokPark/dd3f5c0fa17faea3aac8310bc7a24e6f

Namespace (3)

- Overloading

ch1 — tmux — tmux — 100×22

tmux

```
1 #include <iostream>
2
3 // original X.h
4 int foo(int);
5 // original Y.h
6 int foo(char);
7
8 #include "X.h"
9 #include "Y.h"
10 int main() {
11   foo('a'); // call from Y.h (foo)
12 }
~
~
~
~
~
~
~
namespace2.cc
"namespace2.cc" 12L, 162C
[0] 0:zsh*                    "JonghyeokParkui-MacBo" 23:28 30- 8-21-
```

ch1 — tmux — tmux — 100×22

tmux

```
1 #include <iostream>
2
3 // new X.h
4 namespace X {
5   int foo(int);
6 }
7 // new Y.h
8 namespace Y {
9   int foo(char);
10 }
11
12 #include "X.h"
13 #include "Y.h"
14 using namespace X;
15 using namespace Y;
16
17 int main() {
18   foo('a'); // call from Y.h (foo)
19 }
namespace3.cc
[0] 0:vim*                    "JonghyeokParkui-MacBo" 23:28 30- 8-21-
```

Namespace (3)

- Extension
  - Possible to extension to another file

ch1 — namespace4.cc + (~/Dropbox/VLDB/고...l-cpp-study-jhpark/slides/ch1) - VIM — vim name...

vim

```
1 #include <iostream>
2
3 namespace X {
4   int x;
5   int y;
6 };
7
8 namespace X {
9   int z;
10  int k;
11 };
12
13 int main () {
14   X::z = 3;
15   std::cout << X::z << std::endl;
16 }
17
~
namespace4.cc [+]
```

Namespace (4)

- Namespace and friends
  - If a friend declaration in a non-local class first declares a class(function) → member of the innermost enclosing namespace

ch1 — namespace5.cc + (~/Dropbox/VLDB/고...l-cpp-study-jhpark/slides/ch1) - VIM — vim names...

vim

```
void z(int);
namespace A {
  class X {
    friend void f(X);  // A::f is a friend
  };
  // A::f is not visible here
  X x;
  void f(X) { /* ... */}  // f() is defined and known to be a friend
}

using A::x;

void z()
{
  A::f(x);    // OK
  A::X::f(x); // error: f is not a member of A::X
}
```

namespace5.cc [+]

-- INSERT (paste) --

References

[1] https://www.ibm.com/docs/en/i/7.2

[2] https://www.ibm.com/docs/en/zos/2.4.0?topic=only-namespaces-overloading-c