

# GOOD PROGRAMMING STYLE

4<sup>TH</sup> WEEK LECTURE

엄현상(Eom, Hyeonsang)  
School of Computer Science and Engineering  
Seoul National University

# Outline

- Namespaces
  - Functionality
  - Namespace std
  - Format
  - Examples
- Q&A

# Namespaces

- To group entities under a name
  - Classes
  - Objects
  - Functions
- To divide the global scope in sub-scopes
  - Each one with its own name

# Namespace std

- All the files in the C++ standard library declare all of its entities within the std namespace

```
using namespace std;
```

# Format of Namespaces

```
namespace identifier  
{  
    entities  
}
```

- where identifier is any valid identifier and entities is the set of classes, objects and functions that are included within the namespace
- Example

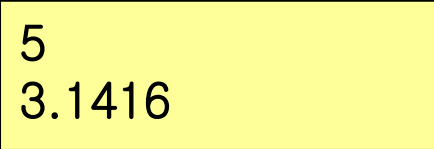
```
namespace myNamespace  
{  
    int a, b;  
}
```

```
myNamespace::a  
myNamespace::b
```

# Examples

- To avoid redefinition errors

```
// namespaces
#include <iostream>
using namespace std;
namespace first
{
    int var = 5;
}
namespace second
{
    double var = 3.1416;
}
int main () {
    cout << first::var << endl;
    cout << second::var << endl;
    return 0;
}
```



5  
3.1416

# Examples Cont'd

- To introduce a name from a namespace

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

```
namespace first
{
    int x = 5;
    int y = 10;
}
```

```
namespace second
{
    double x = 3.1416;
    double y = 2.7183;
}
```

```
int main () {
    using first::x;
    using second::y;
    cout << x << endl;
    cout << y << endl;
    cout << first::y << endl;
    cout << second::x
    << endl;
    return 0;
}
```

```
5
2.7183
10
3.1416
```

# Examples Cont'd

- To introduce a name from a namespace

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

```
namespace first
{
    int x = 5;
    int y = 10;
}
```

```
namespace second
{
    double x = 3.1416;
    double y = 2.7183;
}
```

```
int main () {
    using namespace first;
    cout << x << endl;
    cout << y << endl;
    cout << second::x
    << endl;
    cout << second::y
    << endl;
    return 0;
}
```

```
5
10
3.1416
2.7183
```



# Examples Cont'd

- To introduce a name from a namespace

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

namespace first
{
    int x = 5;
}

namespace second
{
    double x = 3.1416;
}
```

```
int main () {
    {
        using namespace first;
        cout << x << endl;
    }
    {
        using namespace
second;
        cout << x << endl;
    }
    return 0;
}
```

```
5
3.1416
```

# Namespace Alias

- To declare alternate names for existing namespaces

```
namespace new_name = current_name;
```