Namespaces

Victor Eijkhout, Susan Lindsey

Fall 2023

last formatted: November 3, 2023

1. You have already seen namespaces

Safest:

```
#include <vector>
int main() {
  std::vector<stuff> foo;
}
```

Drastic:

```
#include <vector>
using namespace std;
int main() {
  vector<stuff> foo;
}
```

Prudent:

```
#include <vector>
using std::vector;
int main() {
  vector<stuff> foo;
}
```

2. Defining a namespace

Introduce new namespace:

```
namespace a_namespace {
  // definitions
  class an_object {
  };
|
```

3. Namespace usage

Double-colon notation for namespace and type:

```
a_namespace::an_object myobject();
```

or

```
using a_namespace::an_object;
an_object myobject();
```

or even

```
using namespace a_namespace;
an_object myobject();
```

4. Including and using a namespace

There is a *vector* in the standard namespace and in the new *geometry* namespace:

```
// namespace/geo.cpp
#include <vector>
#include "geolib.hpp"
using namespace geometry;
int main() {
  // std vector of geom vectors:
  std::vector< vector > vectors;
  vectors.push_back( vector( point(1,1),point(4,5) ) );
```

5. Header definition

```
// namespace/geolib.hpp
namespace geometry {
  class point {
  private:
    double xcoord,ycoord;
  public:
    point() {};
    point( double x,double y );
    double x();
    double y();
  };
  class vector {
  private:
    point from,to;
  public:
    vector( point from,point to);
    double size();
  };
}
```

6. Implementations

```
// namespace/geolib.cpp
namespace geometry {
  point::point( double x,double y ) {
      xcoord = x; ycoord = y; };
  double point::x() { return xcoord; }; // `accessor'
  double point::y() { return ycoord; };
  vector::vector( point from,point to) {
    this->from = from; this->to = to;
  };
  double vector::size() {
    double
      dx = to.x()-from.x(), dy = to.y()-from.y();
    return sqrt( dx*dx + dy*dy );
  };
}
```

7. Why not 'using namespace std'?

This compiles, but should not:

```
// func/swapname.cpp
#include <iostream>
using namespace std;

def swop(int i,int j) {};

int main() {
  int i=1,j=2;
  swap(i,j);
  cout << i << '\n';
  return 0;
}
```

This gives an error:

```
// func/swapusing.cpp
#include <iostream>
using std::cout;

def swop(int i,int j) {};

int main() {
  int i=1,j=2;
  swap(i,j);
  cout << i << '\n';
  return 0;
}
```