

CS 225

Data Structures

August 25 – Classes and Reference Variables


G Carl Evans



Encapsulation

Cube.h

Cube.cpp



```
1 #pragma once Cube.h
2
3 class Cube {
4     public:
5
6
7
8
9
10
11     private:
12
13
14 };
15
16
17
18
19
20
```

Cube.cpp

```
1 #include "Cube.h"
2
3 double Cube::getVolume() {
4
5
6 }
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Cube.h

```
4 class Cube {
5     public:
6         double getVolume();
```

Cube.h

```
1 #pragma once
2
3 class Cube {
4     public:
5         double getVolume();
6
7
8
9
10
11     private:
12
13
14 };
15
16
17
18
19
20
```

Cube.cpp

```
1 #include "Cube.h"
2
3 double Cube::getVolume() {
4
5
6 }
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```



Namespaces

Namespaces

cs225

Cube

PNG

HSLAPixel

std

cout

vector

queue

...

...

Cube.h

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6         double getVolume();
7         double getSurfaceArea();
8
9
10
11
12     private:
13         double length_;
14
15     };
16 }
17
18
19
20
```

Cube.cpp

```
1 #include "Cube.h"
2
3 namespace cs225 {
4     double Cube::getVolume() {
5         return length_ * length_ *
6             length_;
7     }
8
9     double
10    Cube::getSurfaceArea() {
11        return 6 * length_ *
12            length_;
13    }
14 }
15
16
17
```


Cube.h

Cube.cpp

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6         double getVolume();
7         double getSurfaceArea();
8
9
10
11
12 private:
```

```
1 #include "Cube.h"
2
3 namespace cs225 {
4     double Cube::getVolume() {
5         return length_ * length_ *
6             length_;
7     }
8
9     double
10    Cube::getSurfaceArea() {
11        return 6 * length_ *
12            length_;
```

```
13
14 #include "Cube.h"
15 #include <iostream>
16
17 int main() {
18     cs225::Cube c;
19     std::cout << "Volume: " << c.getVolume() << std::endl;
20     return 0;
21 }
```

main.cpp

```
1 #include "Cube.h"
2 #include <iostream>
3
4 int main() {
5     cs225::Cube c;
6     std::cout << "Volume: " << c.getVolume() << std::endl;
7     return 0;
8 }
```

main.cpp

```
1 #include "Cube.h"  
2 #include <iostream>  
3  
4 int main() {  
5     cs225::Cube c;  
6     std::cout << "Volume: " << c.getVolume() << std::endl;  
7     return 0;  
8 }
```

main.cpp

```
1 #include "Cube.h"
2 #include <iostream>
3
4 int main() {
5     cs225::Cube c;
6     std::cout << "Volume: " << c.getVolume() << std::endl;
7     return 0;
8 }
```



Constructor

Cube.h

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6         Cube();
7         double getVolume();
8         double getSurfaceArea();
9
10
11
12
13     private:
14         double length_;
15
16 };
17 }
18
19
20
```

Cube.cpp

```
1 #include "Cube.h"
2 namespace cs225 {
3     Cube::Cube() {
4
5
6     }
7
8     double Cube::getVolume() {
9         return length_ * length_ *
10            length_;
11     }
12
13     double
14     Cube::getSurfaceArea() {
15         return 6 * length_ *
16            length_;
17     }
18 }
19
20
```

Cube.h

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6         Cube(double length);
7         double getVolume();
8         double getSurfaceArea();
9
10
11
12
13     private:
14         double length_;
15
16 };
17 }
18
19
20
```

Cube.cpp

```
1 #include "Cube.h"
2 namespace cs225 {
3     Cube::Cube(double length) {
4
5
6     }
7
8     double Cube::getVolume() {
9         return length_ * length_ *
10            length_;
11     }
12
13     double
14     Cube::getSurfaceArea() {
15         return 6 * length_ *
16            length_;
17     }
18 }
19
20
```

Cube.h

Cube.cpp

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6         Cube(double length);
7         double getVolume();
8         double getSurfaceArea();
9
```

```
1 #include "Cube.h"
2 namespace cs225 {
3     Cube::Cube(double length) {
4         length_ = length;
5
6     }
7
8     double Cube::getVolume() {
```

```
10 #include "Cube.h"
11 using cs225::Cube;
12 #include <iostream>
13 using std::cout;
14 using std::endl;
15
16 int main() {
17     Cube c;
18     cout << "Volume: " << c.getVolume() << endl;
19     return 0;
20 }
```

puzzle.cpp *

Hate Typing cout:: and cs225::?

Useful Shortcut:

```
using std::cout;    // Imports cout into global scope
using std::endl;    // Imports endl into global scope
using cs225::Cube; // Imports Cube into global scope
```

Strongly Discouraged Shortcut

```
using namespace std; // Imports all of std:: into
                    // global scope!
                    // ...THOUSANDS of things!
```

Cube.h

Cube.cpp

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6
7         Cube(double length);
8         double getVolume();
9         double getSurfaceArea();
10
11
12
13
14
15
16
17
18
19
20
```

```
1 #include "Cube.h"
2 namespace cs225 {
3
4
5
6
7     Cube::Cube(double length) {
8         length_ = length;
9     }
10
11     double Cube::getVolume() {
```

```
7 int main() {
8     Cube c;
9     cout << "Volume: " << c.getVolume() << endl;
10    return 0;
11 }
12
13
14
15
16
17
18 }
```

puzzle.cpp *

Cube.h

Cube.cpp

```
1 #pragma once
2
3 namespace cs225 {
4     class Cube {
5     public:
6
7         Cube(double length);
8         double getVolume();
9         double getSurfaceArea();
10
11
12
13
14
15
16
17
18
19
20
```

```
1 #include "Cube.h"
2 namespace cs225 {
3
4
5
6
7     Cube::Cube(double length) {
8         length_ = length;
9     }
10
11     double Cube::getVolume() {
```

```
7 int main() {
8     Cube c;
9     cout << "Volume: " << c.getVolume() << endl;
10    return 0;
11 }
12
13
14
15
16
17
18 }
```

puzzle.cpp *



CS 225 – Things To Be Doing

lab_intro start in lab (Due Aug. 5th)

Make sure to attend your lab section every week
– they're worth points and EC starting the third week!

MP1 is released Tomorrow!

Due: Monday, Aug. 6th (~12 days after release)

Ensure you are on our Piazza

Details on the course website: <https://courses.engr.illinois.edu/cs225/>

See you Friday!