

# CS 225

## Data Structures

*August 29 – Classes and Reference Variables*

*Wade Fagen-Ulmschneider*

## 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     double
9     Cube::getSurfaceArea() {
10         return 6 * length_ *
11             length_;
12
13     }
14 }
15
16
17
```

## 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 #include "Cube.h"
14 #include <iostream>
15
16 }
17
18 int main() {
19     cs225::Cube c;
20     std::cout << "Volume: " << c.getVolume() << std::endl;
21     return 0;
22 }
```

## 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_;
```

## 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 }
```

# CS 225 – Office Hours

**Lab Sections** – Meet with your TA and CAs every week!

**Open Office Hours** – Held in the basement of Siebel Center by TAs and CAs, ramping up over the next week. First open office hours start this Friday. *(Great place for both conceptual and programming questions!)*

**Faculty Office Hours** –

Wade's Office Hours: Fridays, 9:00am – 10:30am, TBA ECEB

Mattox's Office Hours: Wednesday, 12:15pm – 1:45pm, TBA ECEB

# CS 225 – Exam 0

**First exam is coming up next week!**

“Exam 0”

- Low-stress introduction to the CBTF exam environment.
- This exam is worth only 40 points
- Focuses primarily on foundational knowledge you have from your prerequisite classes.

**Full Details:**

<https://courses.engr.illinois.edu/cs225/fa2018/exams/>

# CBTF-based Exams

All CS 225 exams are held in the Computer Based Testing Facility (CBTF):

- You can choose which day to take your exam within the exam window for a given exam.

- Sign up for your exam here:

<https://cbtf.engr.illinois.edu/>



# 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     double Cube::getVolume() {
```

```
1 #include "Cube.h"
2 using cs225::Cube;
3 #include <iostream>
4 using std::cout;
5 using std::endl;
6
7 int main() {
8     Cube c;
9     cout << "Volume: " << c.getVolume() << endl;
10    return 0;
11 }
```

**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

```
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
```

## Cube.cpp

```
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

```
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
```

## Cube.cpp

```
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** \*



# Pointers and References

# Pointers and References

A variable containing an instance of an object:

```
1 Cube s1;
```

A reference variable of a Cube object:

```
1 Cube & s1;
```

A variable containing a pointer to a Cube object:

```
1 Cube * s1;
```

# Reference Variable

A reference variable is an alias to an existing variable.

*Key Idea: Modifying the reference variable modifies the variable being aliased.*

# Reference Variable

A reference variable is an alias to an existing variable.

```
1 #include <iostream>
2
3 int main() {
4     int i = 7;
5     int & j = i;    // j is an alias of i
6
7     j = 4;
8     std::cout << i << " " << j << std::endl;
9
10    i = 2;
11    std::cout << i << " " << j << std::endl;
12    return 0;
13 }
```

# Reference Variable

Three facts about reference variables:

1.

2.

3.



# CS 225 – Things To Be Doing

**Exam 0** starts on Thursday, Sept. 6<sup>th</sup>

Ensure you sign up for your CBTF timeslot for Exam 0!

**lab\_intro** is due this Sunday (Sept. 2<sup>nd</sup>)

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

**MP1** is released Friday!

Due: Monday, Sept. 10<sup>th</sup> (~10 days after release)

**Ensure you are on our Piazza**

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

**See you Friday!**