Variables and Classes in C++
Every variable is defined by four properties:

    1.

    2.

    3.

    4.

And every variable is one of two types:

`int myFavoriteInt;`
`char grade = 'A';`
`double gamma = 0.653;`
	`Sphere myFavoriteSphere;`
	`Cube rubix;`
	`Grade courseGrade;`

Creating New Types
In data structures, we will be learning and creating new types of structures to store data. We will start simply – by the end, we will have types we built being the building blocks for new types!

Big Idea: Encapsulation

Encapsulation principles:

Cube.h		Cube.cpp	
1		1	
2		2	
3		3	
4		4	
5		5	

Our First Class – Cube:

	Cube.h		Cube.cpp
1	`#pragma once`	1	`#include "Cube.h"`
2		2	
3	`class Cube {`	3	`double Cube::getVolume() {`
4	`  public:`	4	
5	`    double getVolume();`	5	
6		6	`}`
7		7	
8		8	
9		9	
10		10	
11	`  private:`	11	
12		12	
13		13	
14		14	
15		15	
16	`};`	16	

Public vs. Private:

Situation	Protection Level
`Cube` functionality provided to **client code**	
Variable containing data about the `Cube`	
Helper function used in `Cube`	

Hierarchy in C++:
There `Cube` class we're building might not be the only `Cube` class. Large libraries in C++ are organized into _____.

	Cube.h		Cube.cpp
1	`#pragma once`	1	`#include "Cube.h"`
2		2	
3	`namespace cs225 {`	3	`namespace cs225 {`
4	`  class Cube {`	4	`  double`
5	`  public:`		`    Cube::getVolume() {`
6	`    double getVolume();`	5	`    return length_ *`
7			`           length_ * length_;`
…		6	`  }`
		…	

Our First Program:

<table>
<tr><td colspan="2" align="center">main.cpp</td></tr>
<tr><td>1</td><td><code>#include "Cube.h"</code></td></tr>
<tr><td>2</td><td><code>#include <iostream></code></td></tr>
<tr><td>3</td><td></td></tr>
<tr><td>4</td><td><code>int main() {</code></td></tr>
<tr><td>5</td><td><code> cs225::Cube c;</code></td></tr>
<tr><td>6</td><td><code> std::cout << "Volume: " << c.getVolume() << std::endl;</code></td></tr>
<tr><td>7</td><td><code> return 0;</code></td></tr>
<tr><td>8</td><td><code>}</code></td></tr>
</table>

...run this yourself: run `make` and `./main` in the lecture source code.

Several things about C++ are revealed by our first program:

1. _____
   main.cpp:4

2. _____
   main.cpp:5, main.cpp:1

3. _____
   main.cpp:6, main:cpp:2

4. However, our program is unreliable. **Why?**

Default Constructor:
Every class in C++ has a constructor – even if you didn't define one!

- Automatic/Implicit Default Constructor:

- Custom Default Constructor:

<table>
<tr><td colspan="2" align="center">Cube.h</td><td colspan="2" align="center">Cube.cpp</td></tr>
<tr><td>…</td><td></td><td>…</td><td></td></tr>
<tr><td>4</td><td><code>class Cube {</code></td><td>3</td><td><code>Cube::Cube() {</code></td></tr>
<tr><td>5</td><td><code> public:</code></td><td>4</td><td></td></tr>
<tr><td>6</td><td><code> Cube();</code></td><td>5</td><td></td></tr>
<tr><td>…</td><td><code> /* ... */</code></td><td>6</td><td><code>}</code></td></tr>
<tr><td></td><td></td><td>…</td><td></td></tr>
</table>

Custom, Non-Default Constructors:
We can provide also create constructors that require parameters when initializing the variable:

<table>
<tr><td colspan="2" align="center">Cube.h</td><td colspan="2" align="center">Cube.cpp</td></tr>
<tr><td>…</td><td></td><td>…</td><td></td></tr>
<tr><td>4</td><td><code>class Cube {</code></td><td>3</td><td><code>Cube::Cube(double length) {</code></td></tr>
<tr><td>5</td><td><code> public:</code></td><td>4</td><td></td></tr>
<tr><td>6</td><td><code> Cube(double length);</code></td><td>5</td><td></td></tr>
<tr><td>…</td><td><code> /* ... */</code></td><td>6</td><td><code>}</code></td></tr>
<tr><td></td><td></td><td>…</td><td></td></tr>
</table>

Puzzle #1: How do we fix our first program?

<table>
<tr><td colspan="2" align="center">puzzle.cpp w/ above custom constructor</td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>8</td><td><code> cs225::Cube c;</code></td></tr>
<tr><td>9</td><td><code> cout << "Volume: " << c.getVolume() << endl;</code></td></tr>
<tr><td>…</td><td></td></tr>
</table>

...run this yourself: run `make puzzle` and `./puzzle` in the lecture source code.

Solution #1:

Solution #2:

The beauty of programming is both solutions work! There's no one right answer, both have advantages and disadvantages!

<table>
<tr><td align="center">CS 225 – Things To Be Doing:</td></tr>
<tr><td>

1. Attend lab and complete lab_intro; due Sept. 9th
2. MP1 released Today; due Wednesday, Sept. 9th
3. Visit Piazza and the course website often!

</td></tr>
</table>