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

```cpp
... puzzle.cpp w/ above custom constructor
8  cs225::Cube c;
9  cout << "Volume: " << c.getVolume() << endl;
...

...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!

Pointers and References

Often, we will have direct access to our object:

```cpp
Cube s1; // A variable of type Cube
```

Occasionally, we have a reference or pointer to our data:

```cpp
Cube & r1 = s1; // A reference variable of type Cube
Cube * p1; // A pointer that points to a Cube
```

Pointers

Unlike reference variables, which alias another variable's memory, pointers are variables with their own memory. Pointers store the memory address of the contents they're "pointing to".

Three things to remember on pointers:
1.
2.
3.
int main() {
    cs255::Cube c;
    cs255::Cube *p = &c;
    return 0;
}

Puzzle: What happens here?
Cube *CreateCube() {
    Cube c(20);
    return &c;
}

CS 225: TTBD → lab_intro and mp_intro ongoing