CS 225
Data Structures

Sept. 15 - Templates
<table>
<thead>
<tr>
<th>Function</th>
<th>Type of <code>obj</code></th>
<th>Explanation</th>
<th>Type of <code>obj</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>obj.print_1();</code></td>
<td>“Sphere”</td>
<td><code>No print_1()</code> is defined in RedBall, so we use the base class (Sphere)’s <code>print_1()</code>: “Sphere”</td>
<td>“Sphere”</td>
</tr>
<tr>
<td><code>obj.print_2();</code></td>
<td>“Sphere”</td>
<td>The type of <code>obj</code> is RedBall, so we’ll use RedBall’s implementation: “Ball”</td>
<td>The type of <code>obj</code> is Sphere, so we’ll use Sphere’s impl since <code>Sphere::print_2()</code> is not virtual: “Sphere”</td>
</tr>
<tr>
<td><code>obj.print_3();</code></td>
<td>“Sphere”</td>
<td><code>No print_3()</code> is defined in RedBall, so we use the base class (Sphere)’s <code>print_3()</code>: “Sphere”</td>
<td>“Sphere”</td>
</tr>
<tr>
<td><code>obj.print_4();</code></td>
<td>“Sphere”</td>
<td>The type of <code>obj</code> is RedBall, so we’ll use RedBall’s implementation: “Ball”</td>
<td>The type of <code>obj</code> is Sphere, but <code>Sphere::print_4()</code> is virtual. Therefore, we will used the derived class’ impl: “Ball”</td>
</tr>
<tr>
<td><code>obj.print_5();</code></td>
<td>Will not compile since Sphere is an abstract class when <code>print_5()</code> is defined as a pure virtual function.</td>
<td>The type of <code>obj</code> is RedBall, so we’ll use RedBall’s implementation: “Ball”</td>
<td>The type of <code>obj</code> is Sphere, but <code>Sphere::print_4()</code> is virtual. Therefore, we will used the derived class’ impl: “Ball”</td>
</tr>
</tbody>
</table>
class Sphere {
  public:
  Sphere(double d) { /* ... */ }
}

class Ball : public Sphere {
}

int main() {
  Ball b;
  return 0;
}
Abstract Class:

[Requirement]:

[Syntax]:

[As a result]:
class Sphere {
public:
    virtual Sphere();
}

class Ball : public Sphere {
public:
    ____________________________;
};
```cpp
virtual-dtor.cpp

class Sphere {
    public:
    virtual ~Sphere();
}

class Ball : public Sphere {
    public:
    __________________________;
}
```
Call Order – How are derived classes created?
Call Order – How are derived classes destroyed?
MP: Extra Credit

The most successful MP is an MP done early!
Unless otherwise specified in the MP, we will award +1 extra credit point per day for completing Part 1 before the due date (up to +7 points):

Example for MP2:

+7 points: Complete by Monday, Sept. 18 (11:59pm)
+6 points: Complete by Tuesday, Sept. 19 (11:59pm)
+5 points: Complete by Wednesday, Sept. 20 (11:59pm)
+4 points: Complete by Thursday, Sept. 21 (11:59pm)
+3 points: Complete by Friday, Sept. 22 (11:59pm)
+2 points: Complete by Saturday, Sept. 23 (11:59pm)
+1 points: Complete by Sunday, Sept. 24 (11:59pm)

MP2 Due Date: Monday, Sept 25
MP: Extra Credit

The most successful MP is an MP done early!
We will give partial credit and maximize the value of your extra credit:

You made a submission and missed a few edge cases in Part 1:
  Monday: $+7 \times 80\% = +5.6$ earned

You fixed your code and got a perfect score on Part 1:
  Tuesday: $+6 \times 100\% = +6$ earned \textit{(maximum benefit)}

You began working on Part 2, but added a seg fault to Part 1:
  Wednesday: $+5 \times 0\% = +0$ earned

...
Overloaded Operator LHS/RHS

```cpp
bool Sphere::operator<( ________________ ) {
    // ...
}
```

```cpp
Sphere& Sphere::operator=( ________________ ) {
    // ...
}
```
void Sphere::_destroy() {  delete[] props_;  }
void Sphere::_copy(const Sphere &other) {
    r_ = other.r;
    props_max_ = other.props_max_;   
    props_ct_ = other.props_ct_;   
    props_ = new std::string[10];   
    for (unsigned i = 0; i < props_ct_; i++) {
        props_[i] = other.props_[i];
    }
}
Sphere& Sphere::operator=(const Sphere &other) {
    _destroy();
    _copy(other);
    return *this;
}

#include "Sphere.h"
int main() {
    cs255::Sphere s(10);
    s = s;
    return 0;
}
Abstract Data Type
List ADT
What types of “stuff” do we want in our list?
Templates
```cpp
template1.cpp

T maximum(T a, T b) {
    T result;
    result = (a > b) ? a : b;
    return result;
}

template2.cpp

T maximum(T a, U b) {
    T result;
    result = (a > b) ? a : b;
    return result;
}
```
ifndef LIST_H
#define LIST_H

class List {
    public:

    private:

};
#endif
CS 225 – Things To Be Doing

Exam 2 starts on Monday!
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

lab_inheritance
Due: Sunday, Sept. 17 (11:59pm)

MP2 is out – Early Deadline Monday, Sept. 18
Up to +7 Extra Credit for Early Submission

POTD
Every Monday-Friday – Worth +1 Extra Credit /problem (up to +40 total)