CS 225

Data Structures

Towards a more advanced Sphere…

redSphere-main.cpp

```
 1 // Create a Sphere with various properties, like color,
 2 // texture, and others.
 3 #include "Sphere.h"
 4
 5 int main() {
 6   cs225::Sphere s(10);
 7   s.addProperty("Red");
 8   s.addProperty(“Rubber");
 9   return 0;
10 }
```

sphere.h

```
 1  #ifndef SPHERE_H
 2  #define SPHERE_H
 3
 4  namespace cs225 {
 5    class Sphere {
 6      public:
 7        Sphere();
 8        Sphere(double r);
 9        Sphere(const Sphere &s);
10
…
26        // ...
27      private:
28        double r_;
29
30
31
32
33    };
34  }
35
36  #endif
```

sphere.cpp

```
 1  #include "sphere.h"
 2
 3  namespace cs225 {
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
…       // ...
    }
```

sphere.h

```
 1  #ifndef SPHERE_H
 2  #define SPHERE_H
 3
 4  namespace cs225 {
 5    class Sphere {
 6      public:
 7        Sphere();
 8        Sphere(double r);
 9        Sphere(const Sphere &s);
10
…
26        // ...
27      private:
28        double r_;
29
30
31
32
33    };
34  }
35
36  #endif
```

sphere.cpp

```
 1  #include "sphere.h"
 2
 3  namespace cs225 {
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
…       // ...
    }
```

sphere.h

```
 1  #ifndef SPHERE_H
 2  #define SPHERE_H
 3
 4  namespace cs225 {
 5    class Sphere {
 6      public:
 7        Sphere();
 8        Sphere(double r);
 9        Sphere(const Sphere &s);
10
…
26        // ...
27      private:
28        double r_;
29
30
31
32
33    };
34  }
35
36  #endif
```

sphere.cpp

```
 1  #include "sphere.h"
 2
 3  namespace cs225 {
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
…       // ...
    }
```

sphere.h

```
 1  #ifndef SPHERE_H
 2  #define SPHERE_H
 3
 4  namespace cs225 {
 5    class Sphere {
 6      public:
 7        Sphere();
 8        Sphere(double r);
 9        Sphere(const Sphere &s);
10
…
26        // ...
27      private:
28        double r_;
29
30
31
32
33    };
34  }
35
36  #endif
```

sphere.cpp

```
 1  #include "sphere.h"
 2
 3  namespace cs225 {
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
…       // ...
    }
```

Labs

Every lab is worth 10 points.

Attendance: 4 points (40%)

- Work on your lab for the full lab section, **and/or**
- *Anytime after the intro*, show your TA a completely passing test suite when you've finished the lab!
- Only will get credit in your official lab section

Correctness: 6 points (60%)

- Complete and submit your lab by Sunday @ 11:59pm

Labs

Lab points contribute 100 points to the base 1000 points in this course.

lab_intro + lab_debug → 20 points
13 labs remain * 6 points → 78 points
= 98 points for correctness

13 labs remain * 4 points → 52 attendance points
(50 of those are EC!)

Inheritance

RedBall.h

```
1  #ifndef REDBALL_H
2  #define REDBALL_H
3
4  #include "Sphere.h"
5
6  public RedBall
7    public:
8
9
10
11
12
13
14   private:
15
16
17
18
19
20 }
21
22 #endif
```

RedBall.cpp

```
1  #include "RedBall.h"
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

A _____:

[Constructor]:

[Inherits]:

[Inherits]:

Sphere.cpp

```
1  Sphere::print_1() {
2    cout << "Sphere" << endl;
3  }
4
5  Sphere::print_2() {
6    cout << "Sphere" << endl;
7  }
8
9  virtual Sphere::print_3() {
10   cout << "Sphere" << endl;
11 }
12
13 virtual Sphere::print_4() {
14   cout << "Sphere" << endl;
15 }
16
17 // In .h file:
18 virtual Sphere::print_5() = 0;
19
20
21
22
```

RedBall.cpp

```
1  // No print_1() in RedBall.cpp
2
3
4
5  RedBall::print_2() {
6    cout << "Ball" << endl;
7  }
8
9  // No print_3() in RedBall.cpp
10
11
12
13 RedBall::print_4() {
14   cout << "Ball" << endl;
15 }
16
17 RedBall::print_5() {
18   cout << "Ball" << endl;
19 }
20
21
22
```

Runtime of Virtual Functions

	`Sphere obj;`	`RedBall obj;`	`RedBall r;` `Sphere &s = r;`
`obj.print_1();`			
`obj.print_2();`			
`obj.print_3();`			
`obj.print_4();`			
`obj.print_5();`			

Abstract Class:

[Requirement]:

[Syntax]:

[Therefore]:

virtual-ctor.cpp

```
15 class Sphere {
16   public:
17     virtual Sphere();
18 }
19
20 class Ball : public Sphere {
21   public:
22     ____________________________________________;
23 }
24
```

virtual-dtor.cpp

```
15 class Sphere {
16   public:
17     virtual ~Sphere();
18 }
19
20 class Ball : public Sphere {
21   public:
22     ____________________________________________;
23 }
24
```

Call Order – How are derived classes created?

Call Order – How are derived classes destroyed?

CS 225 – Things To Be Doing

Exam 1 is happening now
Exam 2 registration is available (programming exam)

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)*