CS 225

Data Structures

joinSpheres-returnByValue.cpp

```
11 /*
12  * Creates a new sphere that contains the exact volume
13  * of the volume of the two input spheres.
14  */
15 Sphere joinSpheres(const Sphere &s1, const Sphere &s2) {
16   double totalVolume = s1.getVolume() + s2.getVolume();
17
18   double newRadius = std::pow(
19     (3.0 * totalVolume) / (4.0 * 3.141592654),
20     1.0/3.0
21   );
22
23   return result(newRadius);
24 }
```

```
28 int main() {
29   Sphere *s1 = new Sphere(4);
30   Sphere *s2 = new Sphere(5);
31
32   Sphere s3 = joinSpheres(*s1, *s2);
33
34   delete s1; s1 = NULL;
35   delete s2; s2 = NULL;
36
37   return 0;
28 }
```

joinSpheres-returnByPointer.cpp

```
11 /*
12  * Creates a new sphere that contains the exact volume
13  * of the volume of the two input spheres.
14  */
15 Sphere *joinSpheres(const Sphere &s1, const Sphere &s2) {
16   double totalVolume = s1.getVolume() + s2.getVolume();
17
18   double newRadius = std::pow(
19     (3.0 * totalVolume) / (4.0 * 3.141592654),
20     1.0/3.0
21   );
22
23   return
       new Sphere(newRadius);
24 }
```

```
28 int main() {
29   Sphere *s1 = new Sphere(4);
30   Sphere *s2 = new Sphere(5);
31
32   Sphere *s3 = joinSpheres(*s1, *s2);
33
34   delete s1; s1 = NULL;
35   delete s2; s2 = NULL;
36
37   return 0;
28 }
```

joinSpheres-returnByReference.cpp

```
11 /*
12  * Creates a new sphere that contains the exact volume
13  * of the volume of the two input spheres.
14  */
15 Sphere & joinSpheres(const Sphere &s1, const Sphere &s2) {
16   double totalVolume = s1.getVolume() + s2.getVolume();
17
18   double newRadius = std::pow(
19     (3.0 * totalVolume) / (4.0 * 3.141592654),
20     1.0/3.0
21   );
22
23
24
25   Sphere *result =
        new Sphere(newRadius);
26   return *result;
27 }
```

```
28 int main() {
29   Sphere *s1 = new Sphere(4);
30   Sphere *s2 = new Sphere(5);
31
32   Sphere s3 = joinSpheres(*s1, *s2);
33
34   delete s1; s1 = NULL;
35   delete s2; s2 = NULL;
36
37   return 0;
28 }
```

joinSpheres-returnByReference2.cpp

```
11 /*
12  * Creates a new sphere that contains the exact volume
13  * of the volume of the two input spheres.
14  */
15 Sphere & joinSpheres(const Sphere &s1, const Sphere &s2) {
16   double totalVolume = s1.getVolume() + s2.getVolume();
17
18   double newRadius = std::pow(
19     (3.0 * totalVolume) / (4.0 * 3.141592654),
20     1.0/3.0
21   );
22
23
24
25   Sphere result(newRadius);
26   return result;
27 }
```

```
28 int main() {
29   Sphere *s1 = new Sphere(4);
30   Sphere *s2 = new Sphere(5);
31
32   Sphere s3 = joinSpheres(*s1, *s2);
33
34   delete s1; s1 = NULL;
35   delete s2; s2 = NULL;
36
37   return 0;
28 }
```

Exam #2

- **Will be a coding exam!**
  - **Similar format as the POTDs**
  - **You will compile locally, then get autograder feedback.**
  - **One or two problems.**
    - **MP 1**
    - **Labs**
    - **In-class code**

Honors Section

- **Starts Friday, September 22**
- **Trying to get it to be 5pm; the time in Banner needs to be changed**
- **Topics:**
  - **Functional programming**
  - **Data structures that are *immutable***
  - **Programming “in the large”**
  - **Clojure**

MP1 Deadline

Programming is hard!

MP1 Deadline

Programming is hard!

Every MP in CS 225 will have an automatic 24-hour grace period after the due date.

Due: Monday, 11:59pm

Grade Period until: Tuesday, 11:59pm

MP1 Deadline

Programming is hard!

Every MP in CS 225 will have an automatic 24-hour grace period after the due date.

Due: Monday, 11:59pm

Grade Period until: Tuesday, 11:59pm

Since the MP will past-due, **there are absolutely no office/lab hours on Tuesdays**.

Registration

The last chance to register for CS 225 is today.
We will not being doing any late adds.

If you've registered late, everything so far is due this **Tuesday, Sept. 12 @ 11:59pm**.

- lab_intro
- lab_debug
- mp1

addSpheres.cpp

```
1  #include "sphere.h"
2
3  int main() {
4    cs225::Sphere s1(3), s2(4);
5    cs225::Sphere s3 = s1 + s2;
6    return 0;
7  }
```

Operators that can be overloaded in C++	
Arithmetic	+ - * / % ++ --
Bitwise	& | ^ ~ << >>
Assignment	=
Comparison	== != > < >= <=
Logical	! && ||
Other	[] () ->

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
11
12
13
14
…         // ...
26      private:
27        double r_;
28
29    };
30  }
31
32  #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
…       // ...
    }
```

One Very Special Operator

Definition Syntax (.h):

```
Sphere& operator=(const Sphere& s)
```

Implementation Syntax (.cpp):

```
Sphere& Sphere::operator=(const Sphere& s)
```

Assignment Operator

Similar to Copy Constructor:

Different from Copy Constructor:

What constructors and operators are called?

```
1  Sphere s1, s2;
2
3  s2 = s1;
4
5  Sphere s3 = s1;
6
7  Sphere *s4 = &s3;
8
9  Sphere &s5 = s2;
```

Destructor

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
11
12
13
14
…         // ...
26      private:
27        double r_;
28
29    };
30  }
31
32  #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
…       // ...
    }
```

The “Rule of Three”

Towards a more advanced Sphere…

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
…       // ...
    }
```

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/

Finish MP1 – Due Tonight (11:59pm)

MP1 Grace period until Tuesday @ 11:59pm

MP2 Release: Tuesday, Sept 12th – Up to +7 Extra Credit for Early Submission

POTD

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