



CS 225

Data Structures

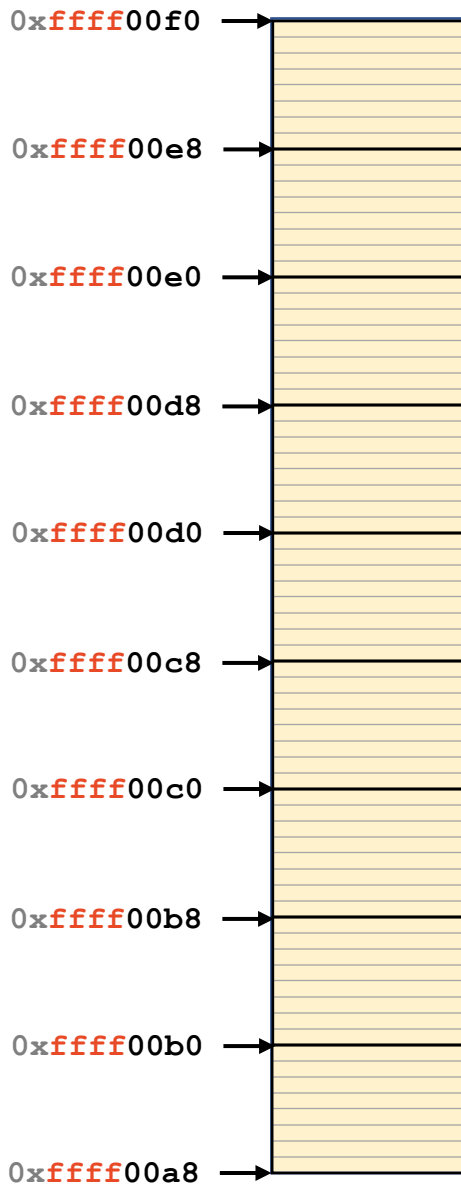
sphere.h

```
1 #ifndef SPHERE_H
2 #define SPHERE_H
3 namespace cs225 {
4     class Sphere {
5     public:
6         Sphere();
7         Sphere(double r);
8         double getRadius();
9         double getVolume();
10
11
12
13
14     private:
15         double r_;
16
17     };
18 }
19
20 #endif
```

sphere.cpp

```
1 #include "sphere.h"
2 namespace cs225 {
3     Sphere::Sphere() {
4
5     }
6
7     Sphere::Sphere(double r) {
8         r_ = r;
9         // ...lots of other logic...
10    }
11
12    double Sphere::getRadius() {
13        return r_;
14    }
15
16    double Sphere::getVolume() {
17        return (4 * r_ * r_ * r_ *
18                3.14159265) / 3.0;
19    }
20 }
```

Stack Memory



example1.cpp

| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0xffff00f0 | → | | |
| 0xffff00e8 | → | | |
| 0xffff00e0 | → | | |
| 0xffff00d8 | → | | |
| 0xffff00d0 | → | | |
| 0xffff00c8 | → | | |
| 0xffff00c0 | → | | |
| 0xffff00b8 | → | | |
| 0xffff00b0 | → | | |
| 0xffff00a8 | → | | |

```
1 int main() {  
2     int a;  
3     int b = -3;  
4     int c = 12345;  
5  
6     int *p = &b;  
7  
8     return 0;  
9 }
```

sizeof-int.cpp

```
1 #include <iostream>
2
3 int main() {
4     std::cout << sizeof(int) << std::endl;
5     return 0;
6 }
```


sizeof-intptr.cpp

```
1 #include <iostream>
2
3 int main() {
4     std::cout << sizeof(int *) << std::endl;
5     return 0;
6 }
```

example1.cpp

| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0x7ffe2ee87228 | | | |
| 0x7ffe2ee87220 | | | |
| 0x7ffe2ee87218 | | | |
| 0x7ffe2ee87210 | | | |
| 0x7ffe2ee87208 | | | |
| 0x7ffe2ee87200 | | | |
| 0x7ffe2ee871f8 | | | |
| 0x7ffe2ee871f0 | | | |
| 0x7ffe2ee871e8 | | | |
| 0x7ffe2ee871e0 | | | |

```
1 int main() {  
2     int a;  
3     int b = -3;  
4     int c = 12345;  
5  
6     int *p = &b;  
7  
8     return 0;  
9 }
```

 Real results when running on linus.ews.illinois.edu

```
&a : 0x7ffe2ee87218  
&b : 0x7ffe2ee87214  
&c : 0x7ffe2ee87210  
&p : 0x7ffe2ee87208
```

example2.cpp

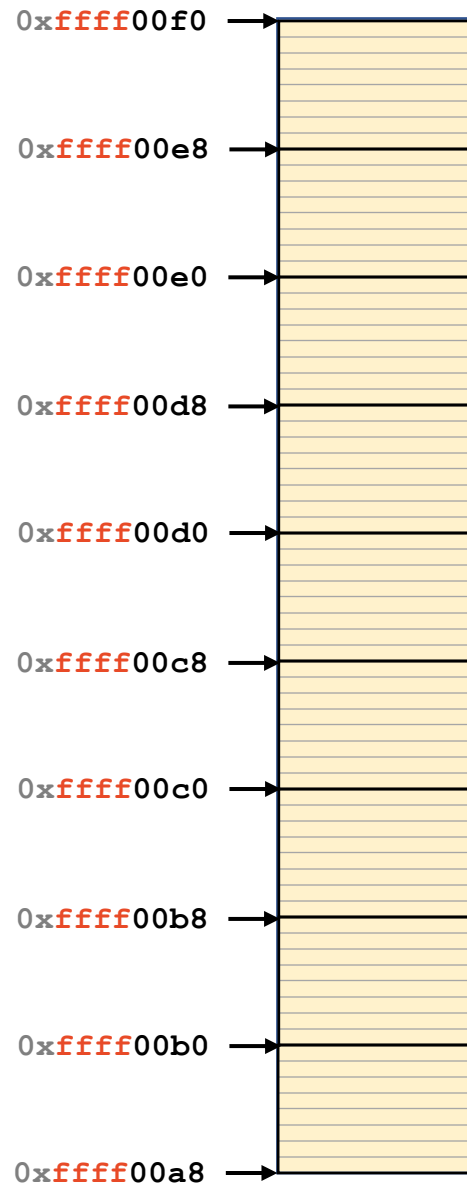
| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0xffff00f0 | → | | |
| 0xffff00e8 | → | | |
| 0xffff00e0 | → | | |
| 0xffff00d8 | → | | |
| 0xffff00d0 | → | | |
| 0xffff00c8 | → | | |
| 0xffff00c0 | → | | |
| 0xffff00b8 | → | | |
| 0xffff00b0 | → | | |
| 0xffff00a8 | → | | |

```
1 #include "sphere.h"
2
3 int main() {
4     cs225::Sphere s;
5     cs225::Sphere *p = &s;
6
7     return 0;
8 }
9
```


sizeof-sphere.cpp

```
1 #include <iostream>
2 #include "sphere.h"
3
4 int main() {
5     std::cout << sizeof(cs225::Sphere) << std::endl;
6     std::cout << sizeof(cs225::Sphere *) << std::endl;
7     return 0;
8 }
```

Stack Frames



```
1 int hello() {  
2     int a = 100;  
3     return a;  
4 }  
5  
6 int main() {  
7     int a;  
8     int b = -3;  
9     int c = hello();  
10    int d = 42;  
11  
12    return 0;  
13 }
```



Return Values

sphere-badCreate.cpp

| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0xffff00f0 | → | | |
| 0xffff00e8 | → | | |
| 0xffff00e0 | → | | |
| 0xffff00d8 | → | | |
| 0xffff00d0 | → | | |
| 0xffff00c8 | → | | |
| 0xffff00c0 | → | | |
| 0xffff00b8 | → | | |
| 0xffff00b0 | → | | |
| 0xffff00a8 | → | | |

```
1 #include "sphere.h"
2 using namespace cs225;
3
4 Sphere *CreateUnitSphere() {
5     Sphere s(1);
6     return &s;
7 }
8
9 int main() {
10     Sphere *s = CreateUnitSphere();
11     double r = s->getRadius();
12     double v = s->getVolume();
13     return 0;
14 }
```

sphere-badCreate.cpp

| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0xffff00f0 | → | | |
| 0xffff00e8 | → | | |
| 0xffff00e0 | → | | |
| 0xffff00d8 | → | | |
| 0xffff00d0 | → | | |
| 0xffff00c8 | → | | |
| 0xffff00c0 | → | | |
| 0xffff00b8 | → | | |
| 0xffff00b0 | → | | |
| 0xffff00a8 | → | | |

```
1 #include "sphere.h"
2 using namespace cs225;
3
4 Sphere *CreateUnitSphere() {
5     Sphere s(1);
6     return &s;
7 }
8
9 int main() {
10     Sphere *s = CreateUnitSphere();
11     double r = s->getRadius();
12     double v = s->getVolume();
13     return 0;
14 }
```

sphere-badCreate.cpp

| <u>Location</u> | <u>Value</u> | <u>Type</u> | <u>Name</u> |
|-----------------|--------------|-------------|-------------|
| 0xffff00f0 | → | | |
| 0xffff00e8 | → | | |
| 0xffff00e0 | → | | |
| 0xffff00d8 | → | | |
| 0xffff00d0 | → | | |
| 0xffee00f0 | → | | |
| 0xffee00e8 | → | | |
| 0xffee00e0 | → | | |
| 0xffee00d8 | → | | |
| 0xffee00d0 | → | | |

```
1 #include "sphere.h"
2 using namespace cs225;
3
4 Sphere *CreateUnitSphere() {
5     Sphere s(1);
6     return &s;
7 }
8
9 int main() {
10     Sphere *s = CreateUnitSphere();
11     double r = s->getRadius();
12     double v = s->getVolume();
13     return 0;
14 }
```



What happens on a real system?

```

13 int main() {
14     Sphere *s = CreateUnitSphere();
15     cout << s->getRadius() << endl;
16     cout << "s->getRadius(): "
17         << s->getRadius() << endl;
18     cout << "&s (main): " << &s << endl;
19     cout << " s (main): " << s << endl;
20     double r = s->getRadius();
21     cout << "&r (main): " << &r << endl;
22     cout << " r (main): " << r << endl;
23     double v = s->getVolume();
24     cout << "&v (main): " << &v << endl;
25     cout << " v (main): " << v << endl;
26 }

```

Real results when running on linus.ews.illinois.edu

```

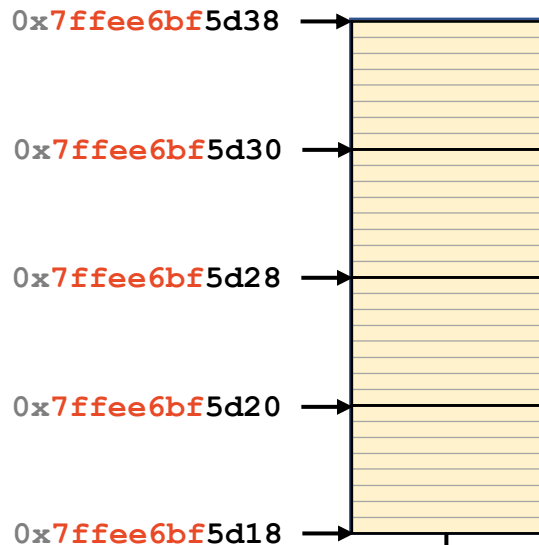
&s (CreateUnitSphere): 0x7ffee6bf5ca8
1
s->getRadius(): 2.07941e-317

&s (main): 0x7ffee6bf5d30
 s (main): 0x7ffee6bf5ca8

&r (main): 0x7ffee6bf5d28
 r (main): 6.95312e-310

&v (main): 0x7ffee6bf5d20
 v (main): 0

```



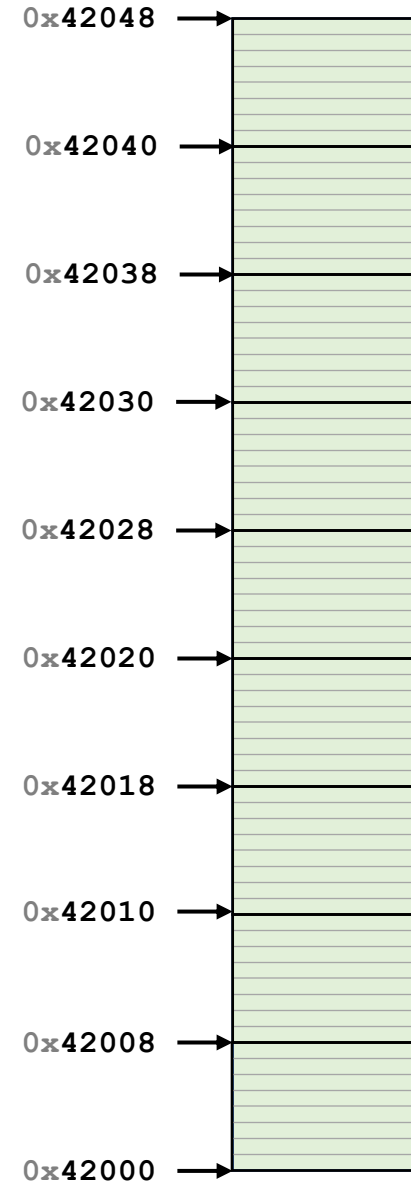
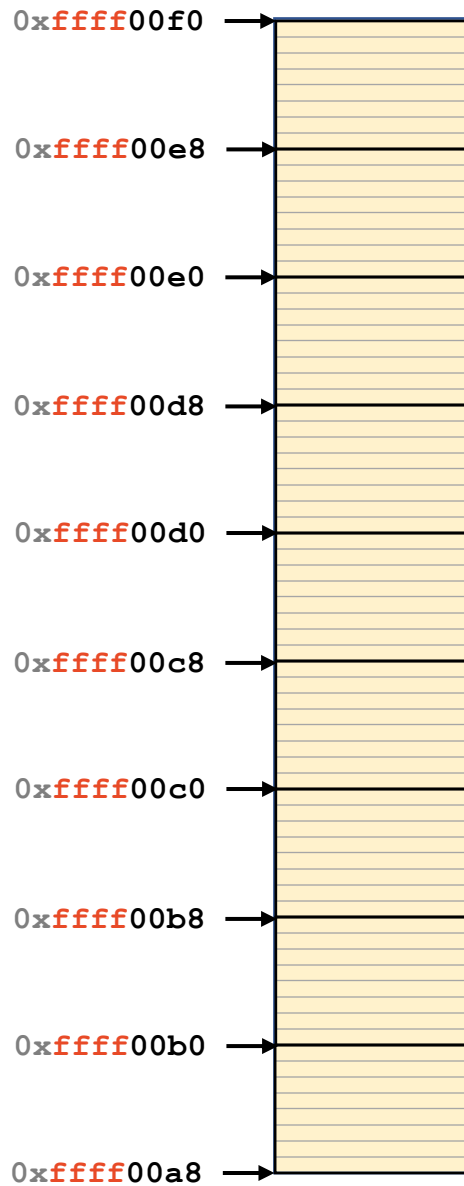
(0x60 bytes not shown)



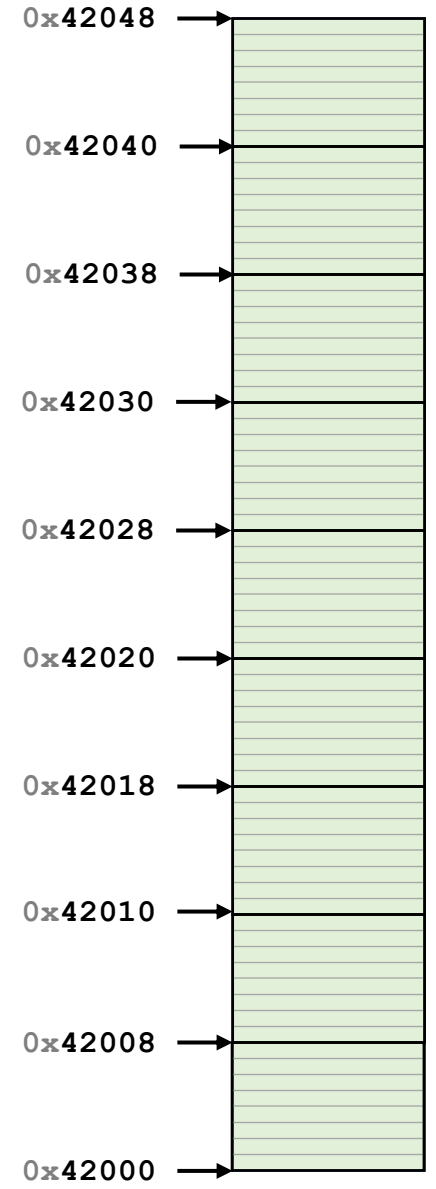
Stack Memory

vs.

Heap Memory

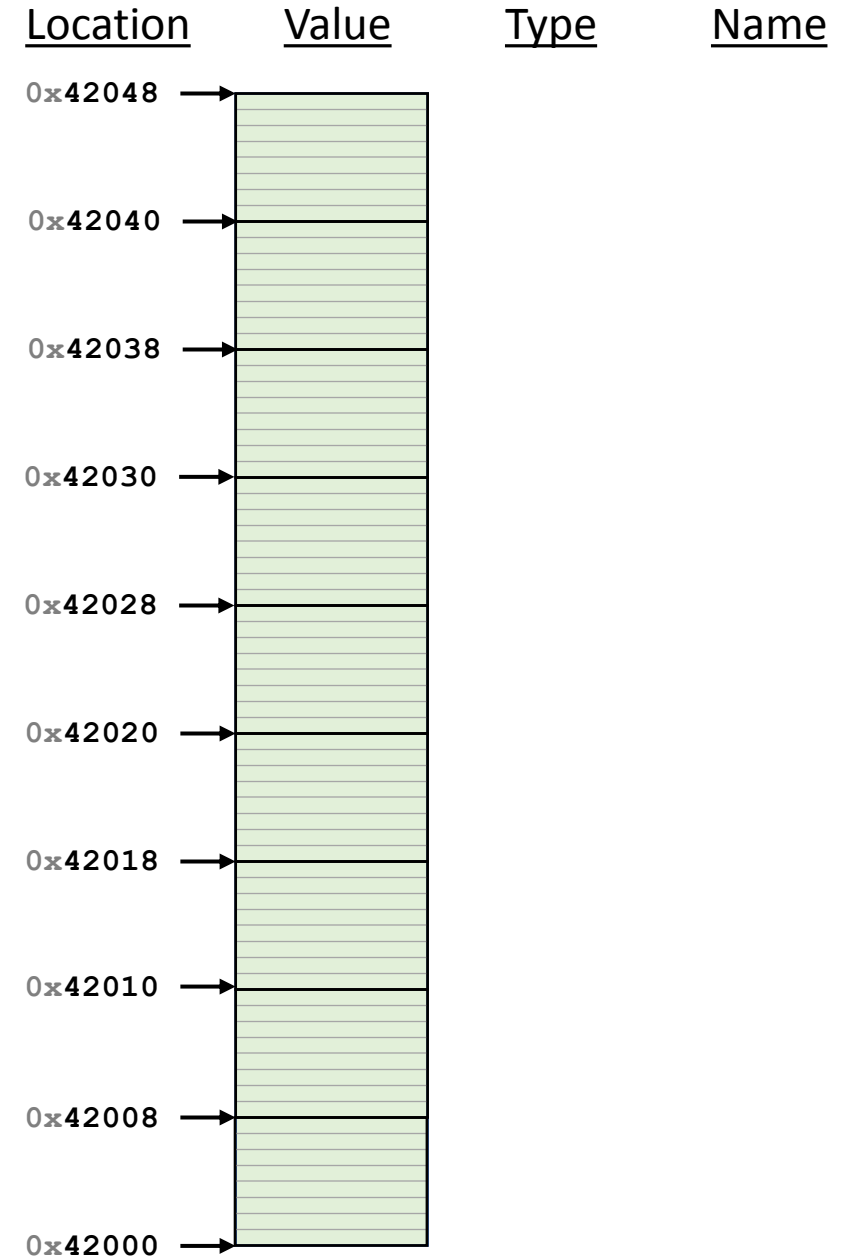
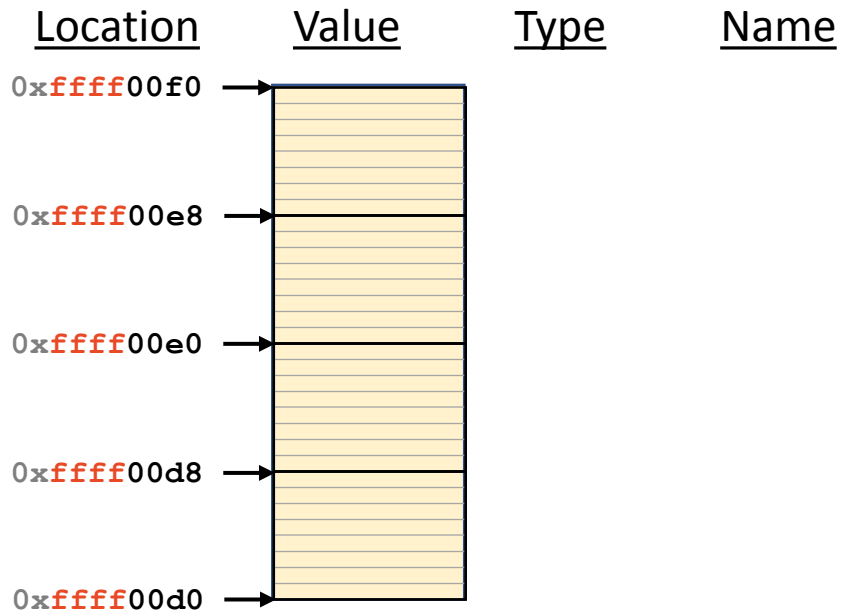


Heap Memory



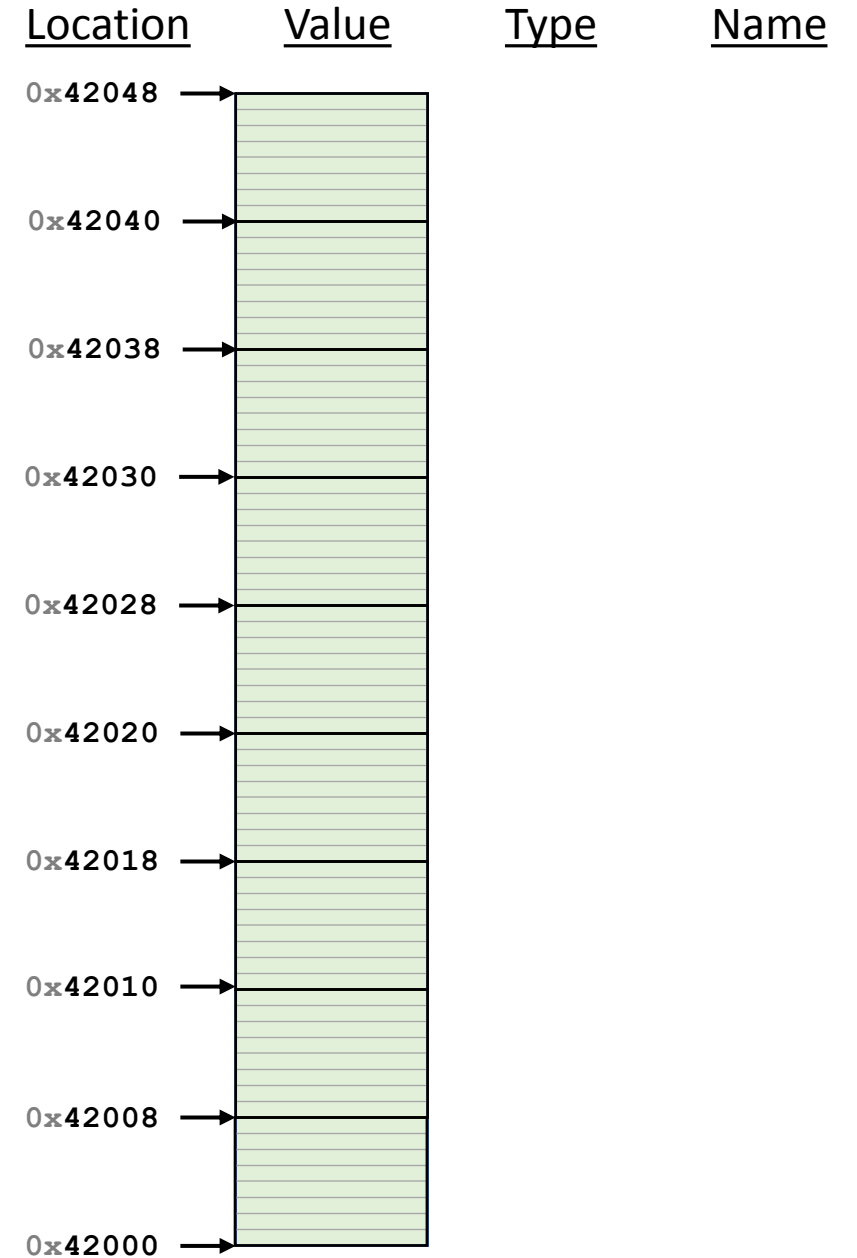
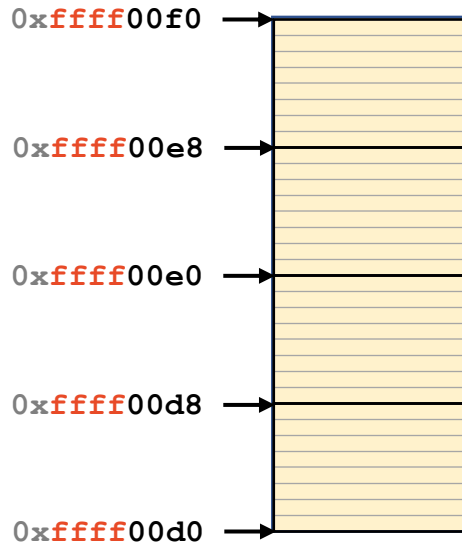
heap1.cpp

```
1 #include "sphere.h"
2 using namespace cs225;
3
4 int main() {
5     int *p = new int;
6     int *s = new Sphere(10);
7
8     return 0;
9 }
```



heap2.cpp

```
1 #include "sphere.h"
2 using namespace cs225;
3
4 int main() {
5     Sphere *s1 = new Sphere();
6     Sphere *s2 = s1;
7
8     s2->setRadius( 10 );
9
10    return 0;
11 }
```



extra-puzzle1.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int *p;
6     int x;
7
8     p = &x;
9     x = 6;
10
11     cout << x << endl;
12     cout << p << endl;
13
14     return 0;
15 }
```

extra-puzzle2.cpp

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int *p, *q;
6     p = new int;
7     q = p;
8     *q = 8;
9     cout << *p << endl;
10
11     q = new int;
12     *q = 9;
13     cout << *p << endl;
14     cout << *q << endl;
15
16     return 0;
17 }
```

CS 225 – Things To Be Doing

lab_intro Due Sunday by 11:59pm

Make sure to turn in lab_intro by Sunday at 11:59pm

MP1 is out tonight! (*after labs have finished*)

Due: Monday, Sept. 11th (weekend + week + weekend from now)

Office Hours are Starting Up...

~20 hours of office hours this weekend (*Check “Calendar” on CS 225 website*)

- *Having trouble with C++?*
- *Having trouble with lab_intro?*
- *Don't understand something from lecture?*

Have an awesome holiday weekend!

...see you next Wednesday!