CS 225

Data Structures

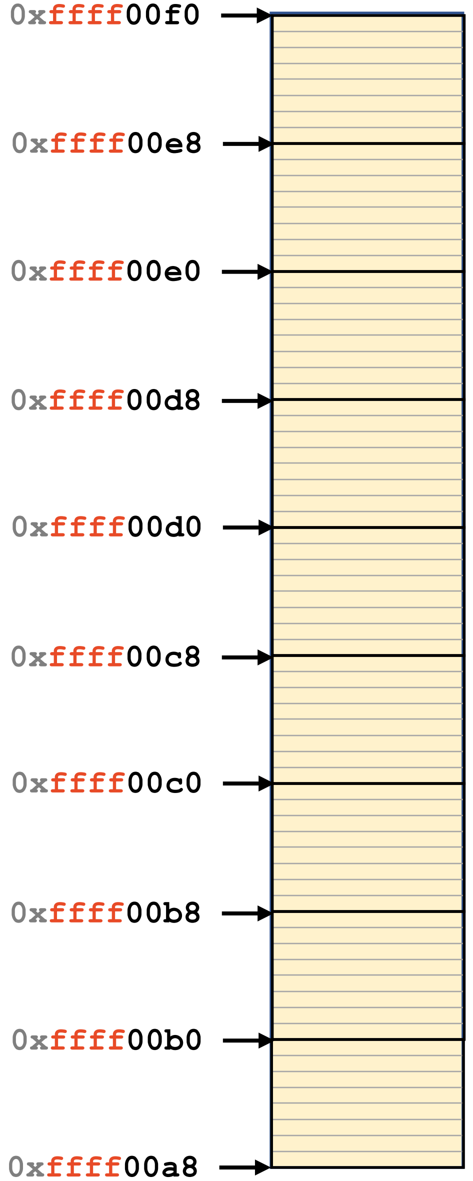
sphere.h

```cpp
#ifndef SPHERE_H
#define SPHERE_H
namespace cs225 {
  class Sphere {
    public:
      Sphere();
      Sphere(double r);
      double getRadius();
      double getVolume();




    private:
      double r_;

  };
}

#endif
```

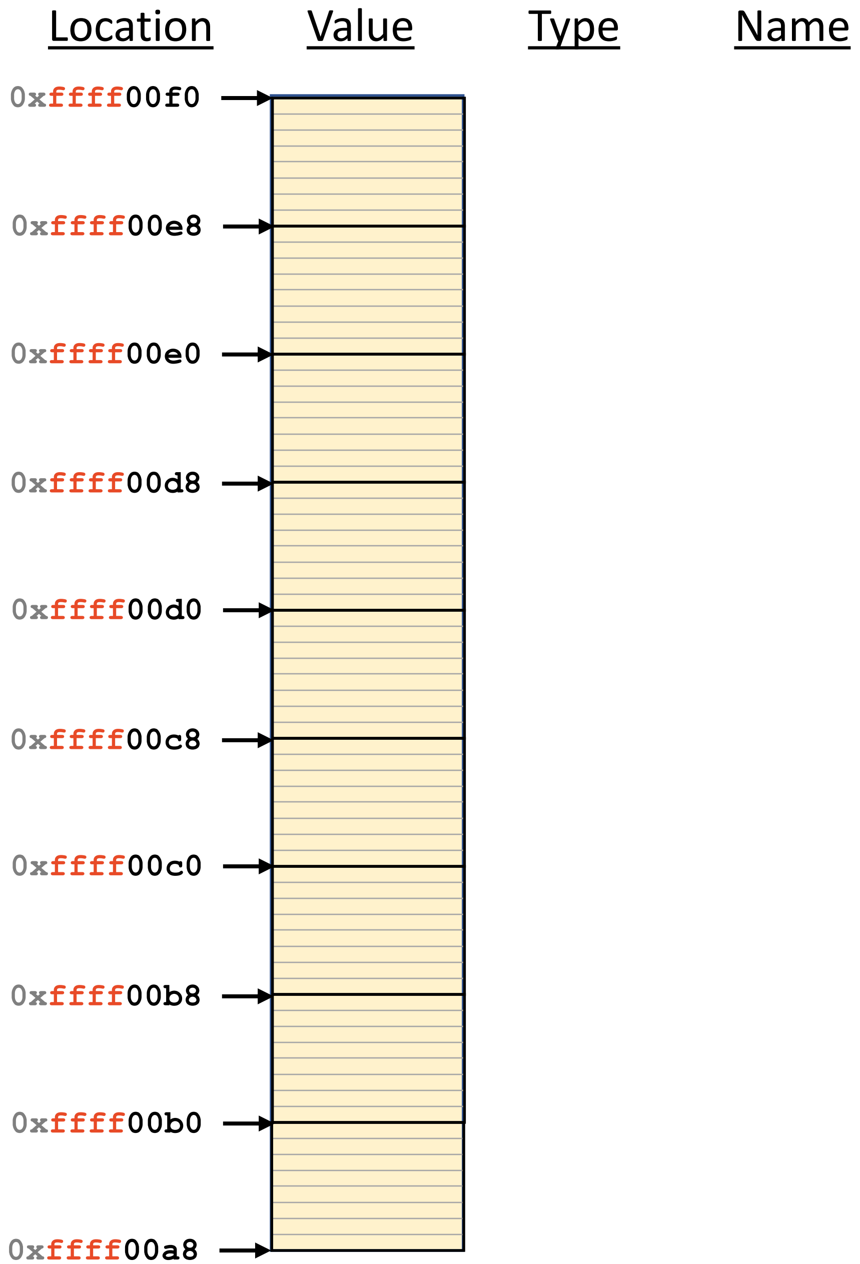
sphere.cpp

```cpp
#include "sphere.h"
namespace cs225 {
  Sphere::Sphere()           {

  }

  Sphere::Sphere(double r) {
     r_ = r;
     // …lots of other logic…
  }

  double Sphere::getRadius() {
    return r_;
  }

  double Sphere::getVolume() {
    return (4 * r_ * r_ * r_ *
            3.14159265) / 3.0;
  }
}
```

Stack Memory



example1.cpp



```cpp
int main() {
  int a;
  int b = -3;
  int c = 12345;

  int *p = &b;

  return 0;
}
```

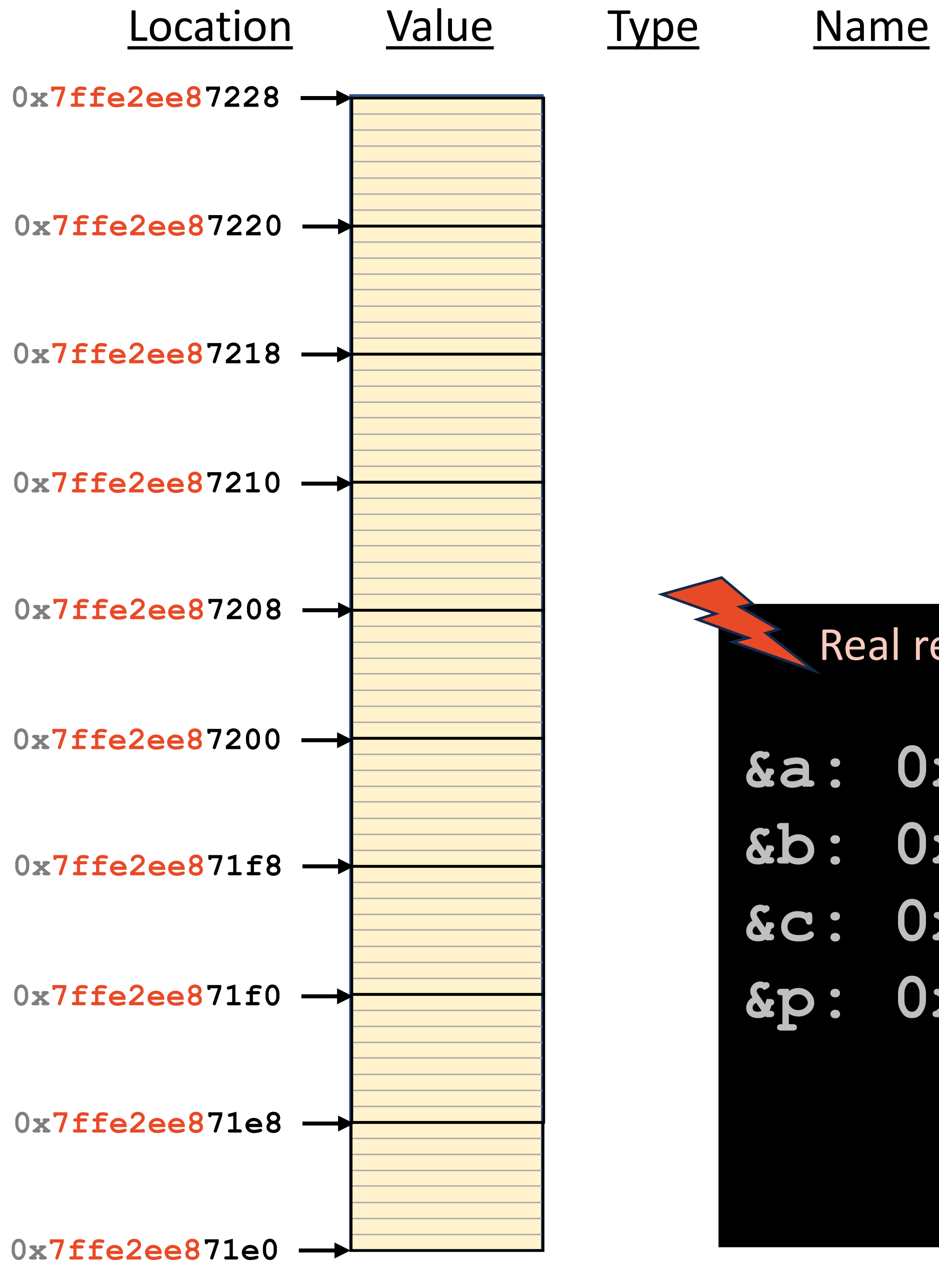
sizeof-int.cpp

```cpp
#include <iostream>

int main() {
  std::cout << sizeof(int) << std::endl;
  return 0;
}
```

sizeof-intptr.cpp

```cpp
#include <iostream>

int main() {
  std::cout << sizeof(int *) << std::endl;
  return 0;
}
```

example1.cpp

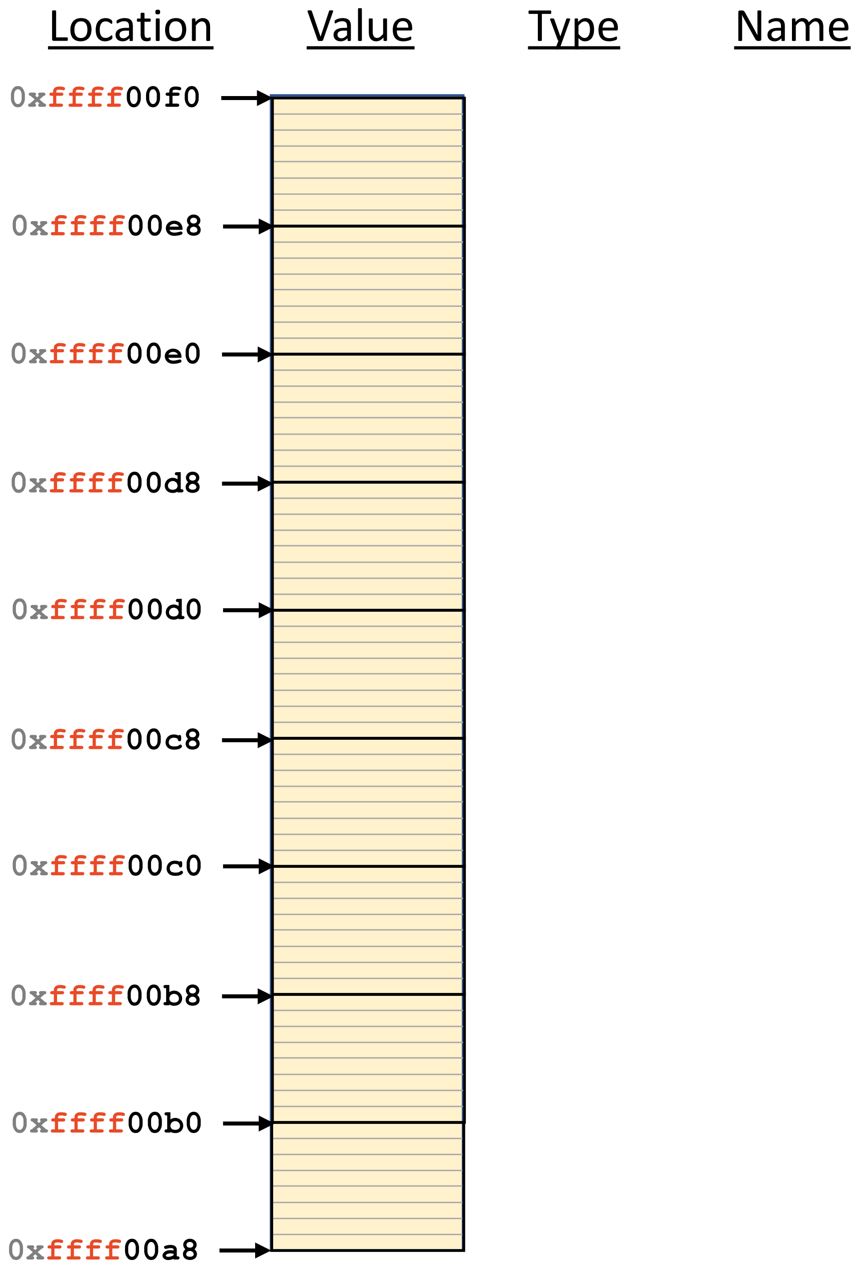


```cpp
int main() {
  int a;
  int b = -3;
  int c = 12345;

  int *p = &b;

  return 0;
}
```

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

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

example2.cpp

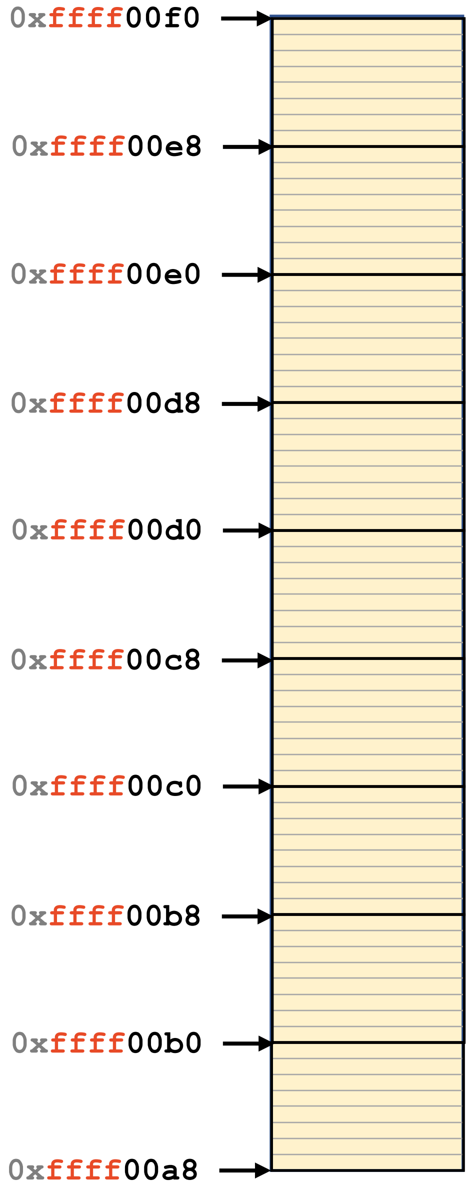


```cpp
#include "sphere.h"

int main() {
  cs225::Sphere s;
  cs225::Sphere *p = &s;

  return 0;
}

```

sizeof-sphere.cpp

```cpp
#include <iostream>
#include "sphere.h"

int main() {
  std::cout << sizeof(cs225::Sphere) << std::endl;
  std::cout << sizeof(cs225::Sphere *) << std::endl;
  return 0;
}
```

Stack Frames

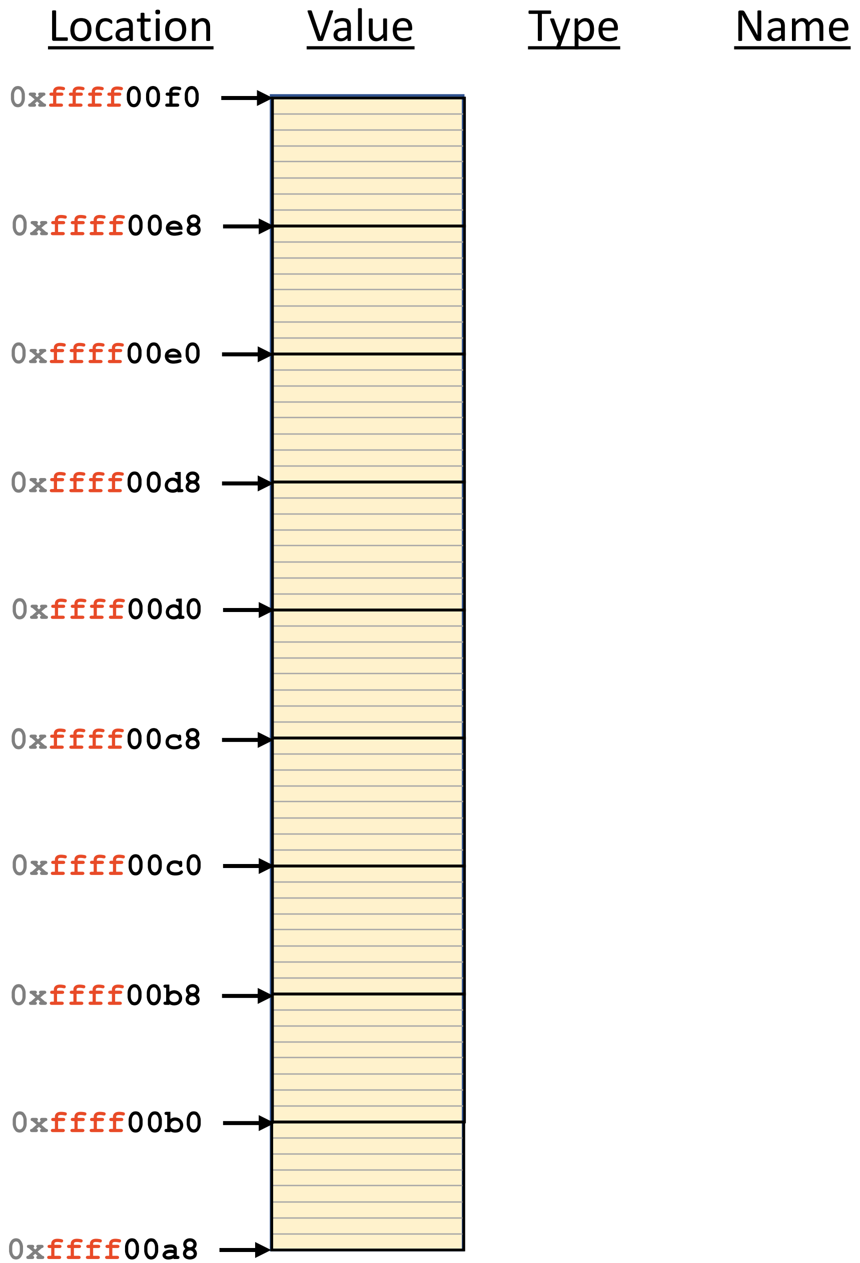


stackframe.cpp

```
int hello() {
  int a = 100;
  return a;
}

int main() {
  int a;
  int b = -3;
  int c = hello();
  int d = 42;

  return 0;
}
```

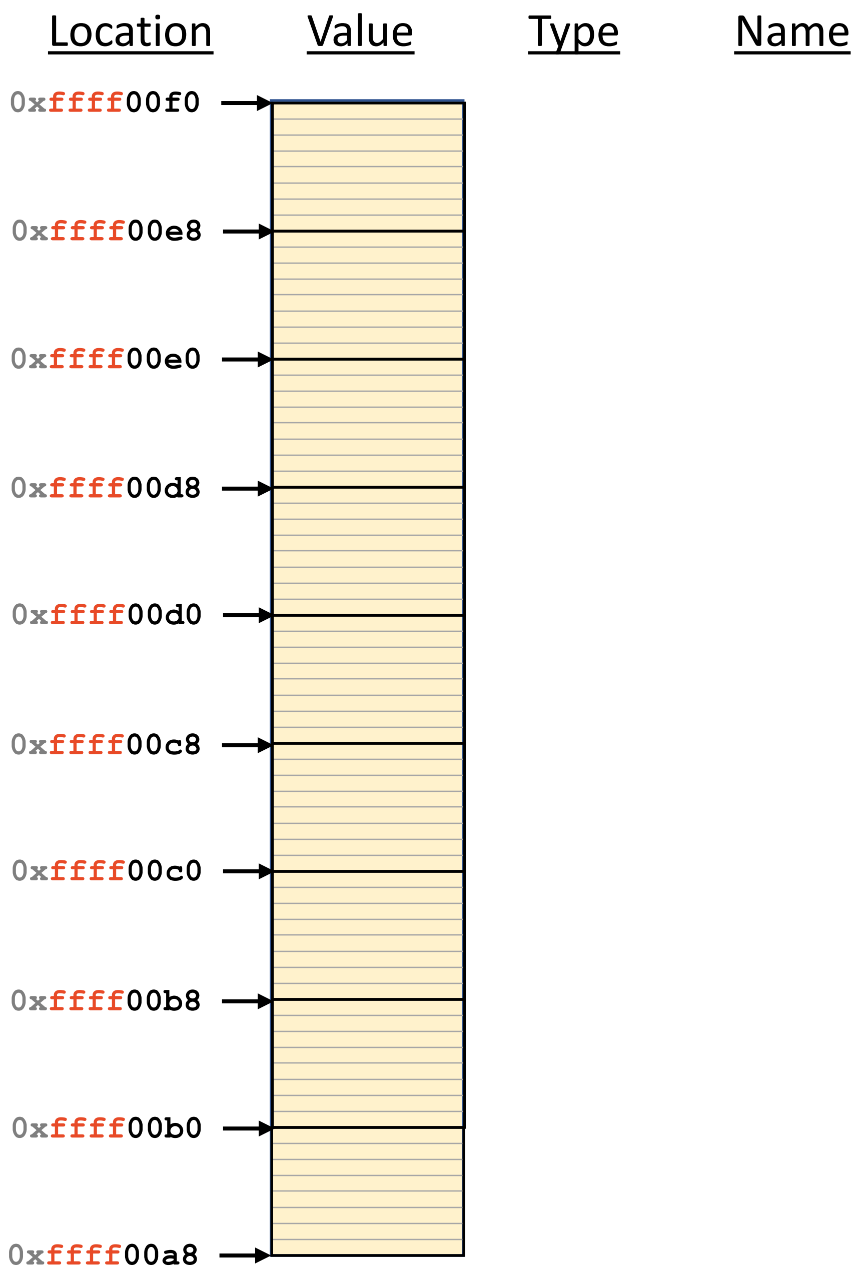
Return Values

sphere-badCreate.cpp



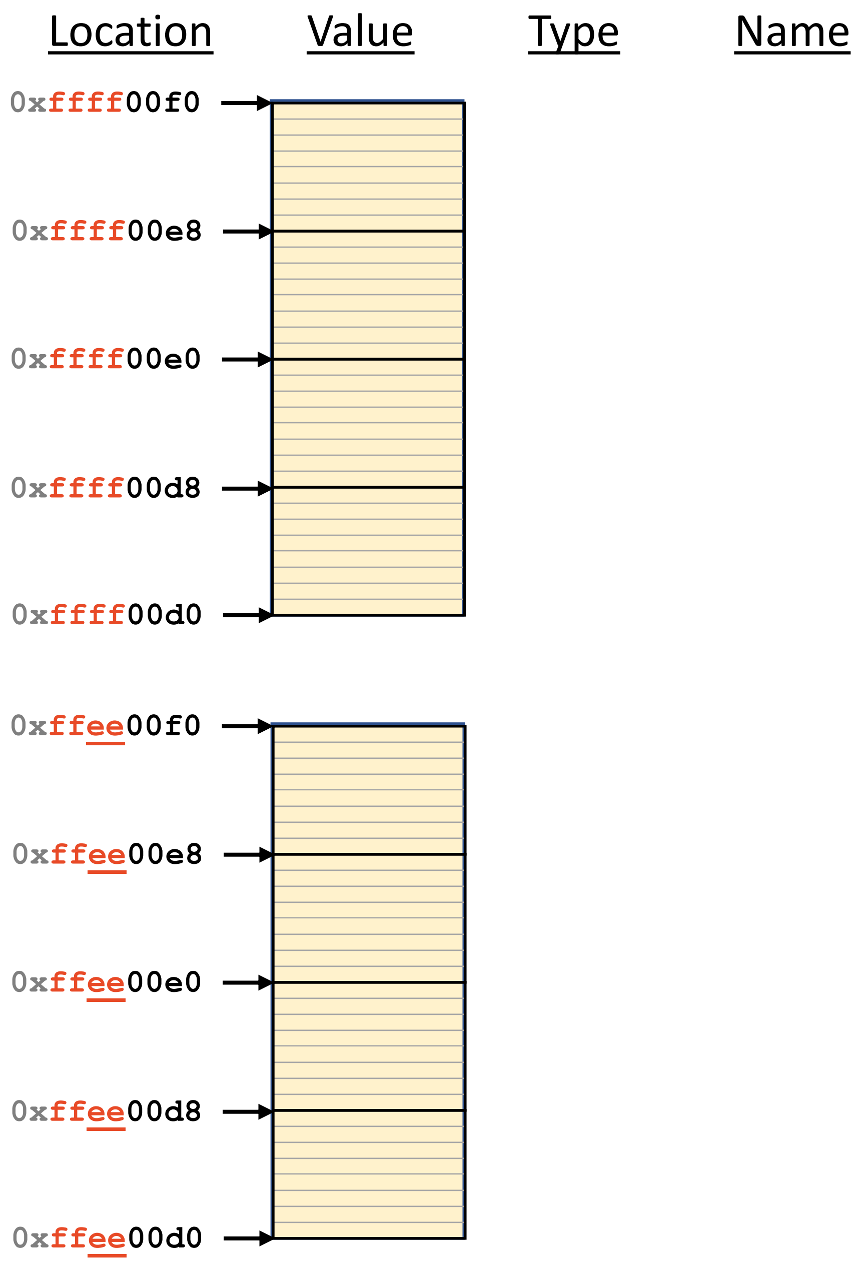
```cpp
#include "sphere.h"
using namespace cs225;

Sphere *CreateUnitSphere() {
  Sphere s(1);
  return &s;
}

int main() {
  Sphere *s = CreateUnitSphere();
  double r = s->getRadius();
  double v = s->getVolume();
  return 0;
}
```

sphere-badCreate.cpp



```
#include "sphere.h"
using namespace cs225;

Sphere *CreateUnitSphere() {
  Sphere s(1);
  return &s;
}

int main() {
  Sphere *s = CreateUnitSphere();
  double r = s->getRadius();
  double v = s->getVolume();
  return 0;
}
```

sphere-badCreate.cpp



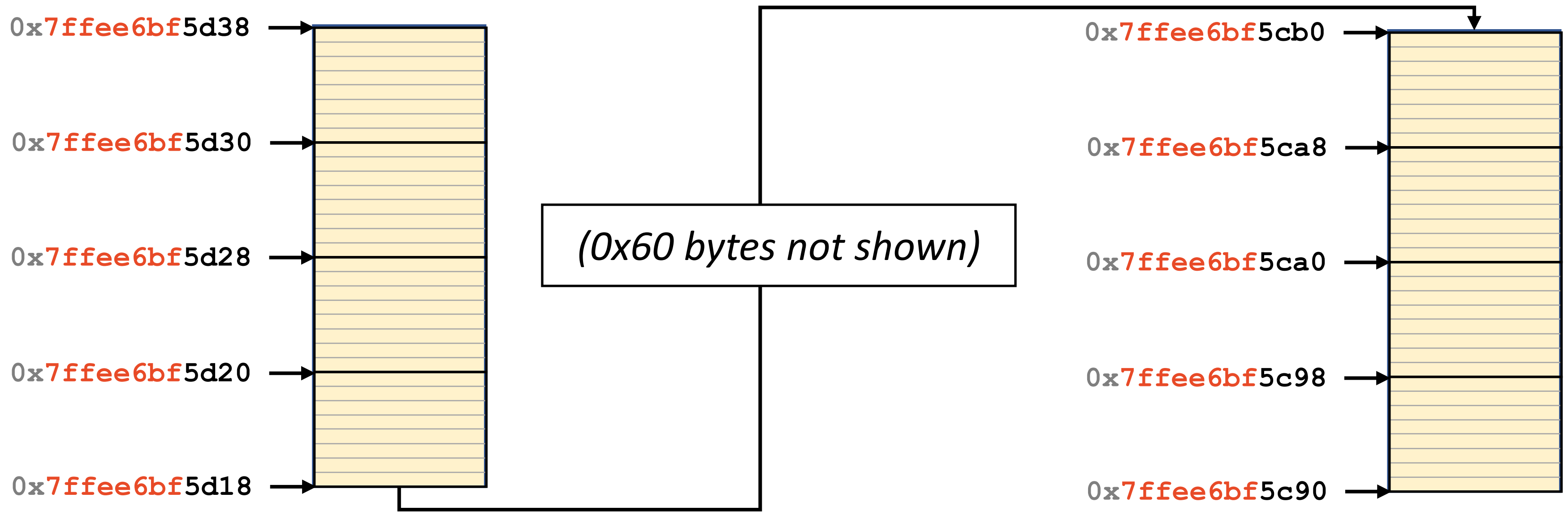
```cpp
#include "sphere.h"
using namespace cs225;

Sphere *CreateUnitSphere() {
  Sphere s(1);
  return &s;
}

int main() {
  Sphere *s = CreateUnitSphere();
  double r = s->getRadius();
  double v = s->getVolume();
  return 0;
}
```

What happens on a real system?

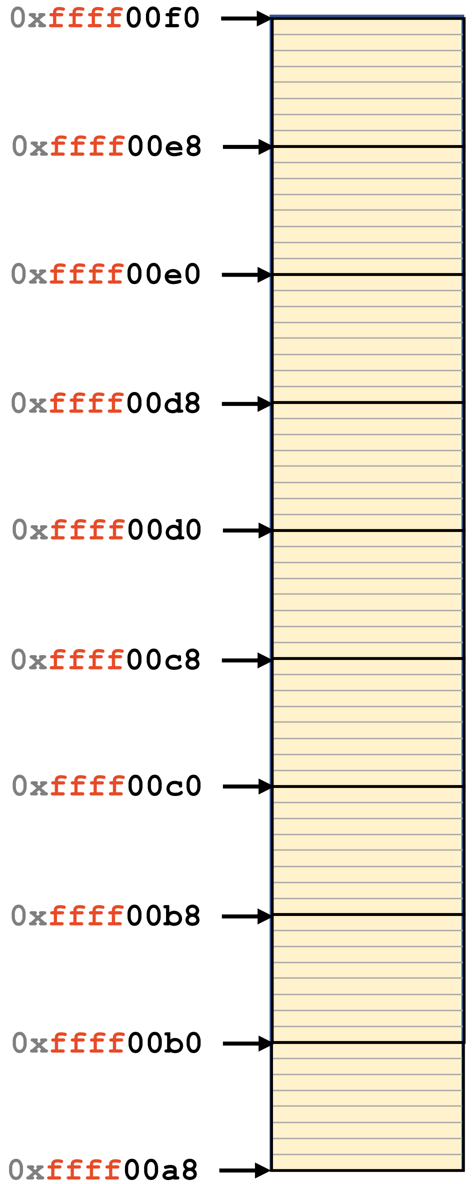
```cpp
int main() {
  Sphere *s = CreateUnitSphere();
  cout << s->getRadius() << endl;
  cout << "s->getRadius(): "
       << s->getRadius() << endl;
  cout << "&s (main): " << &s << endl;
  cout << " s (main): " <<  s << endl;
  double r = s->getRadius();
  cout << "&r (main): " << &r << endl;
  cout << " r (main): " <<  r << endl;
  double v = s->getVolume();
  cout << "&v (main): " << &v << endl;
  cout << " v (main): " <<  v << endl;
}
```

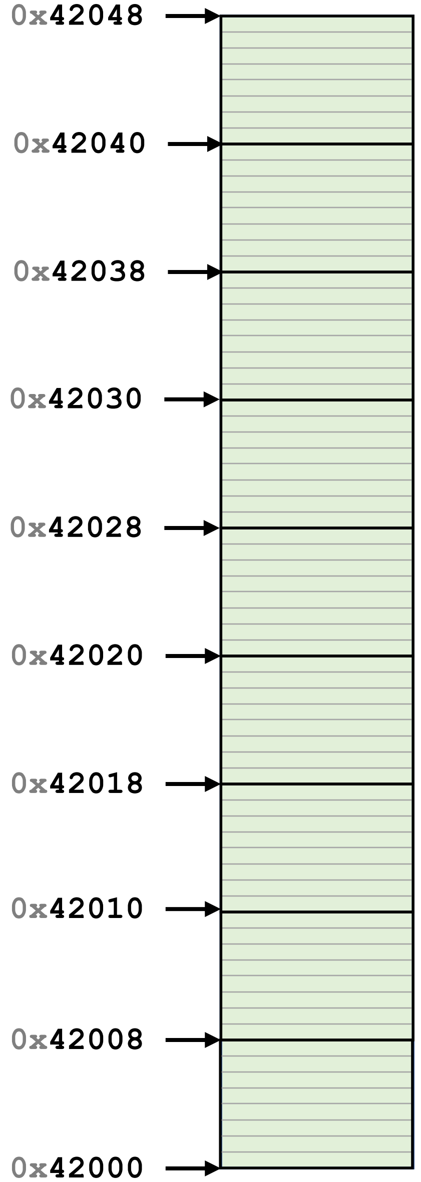
Real results when running on **linus.ews.illinois.edu**

```
&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
```

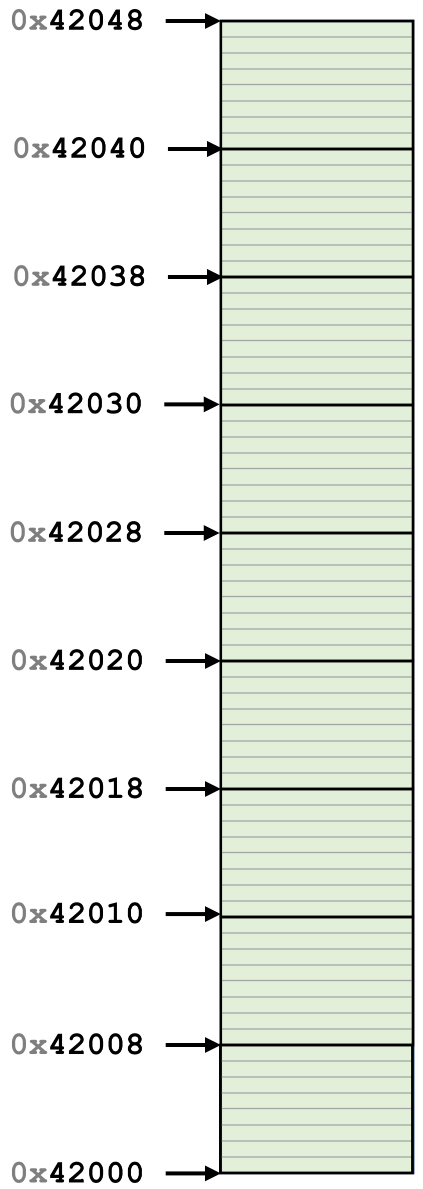


Stack Memory vs. Heap Memory



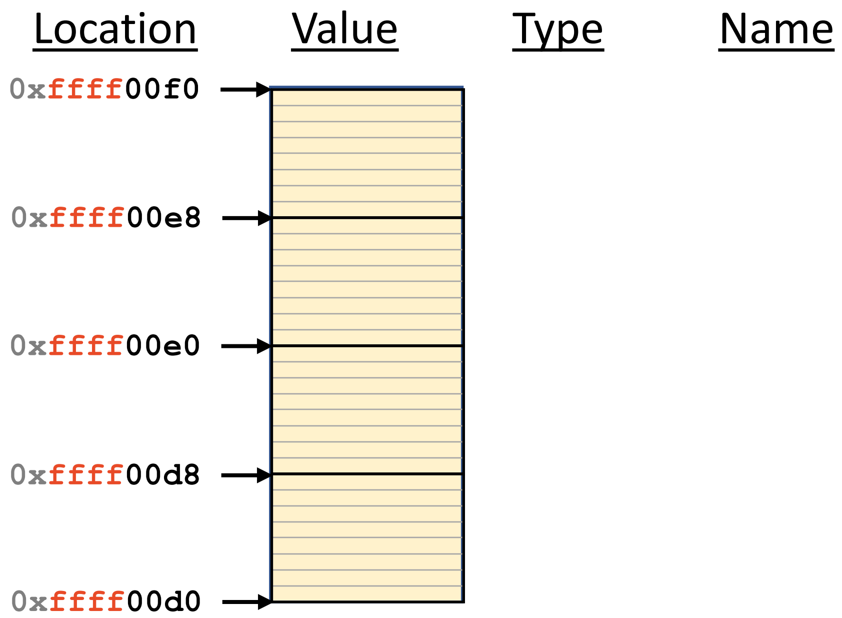


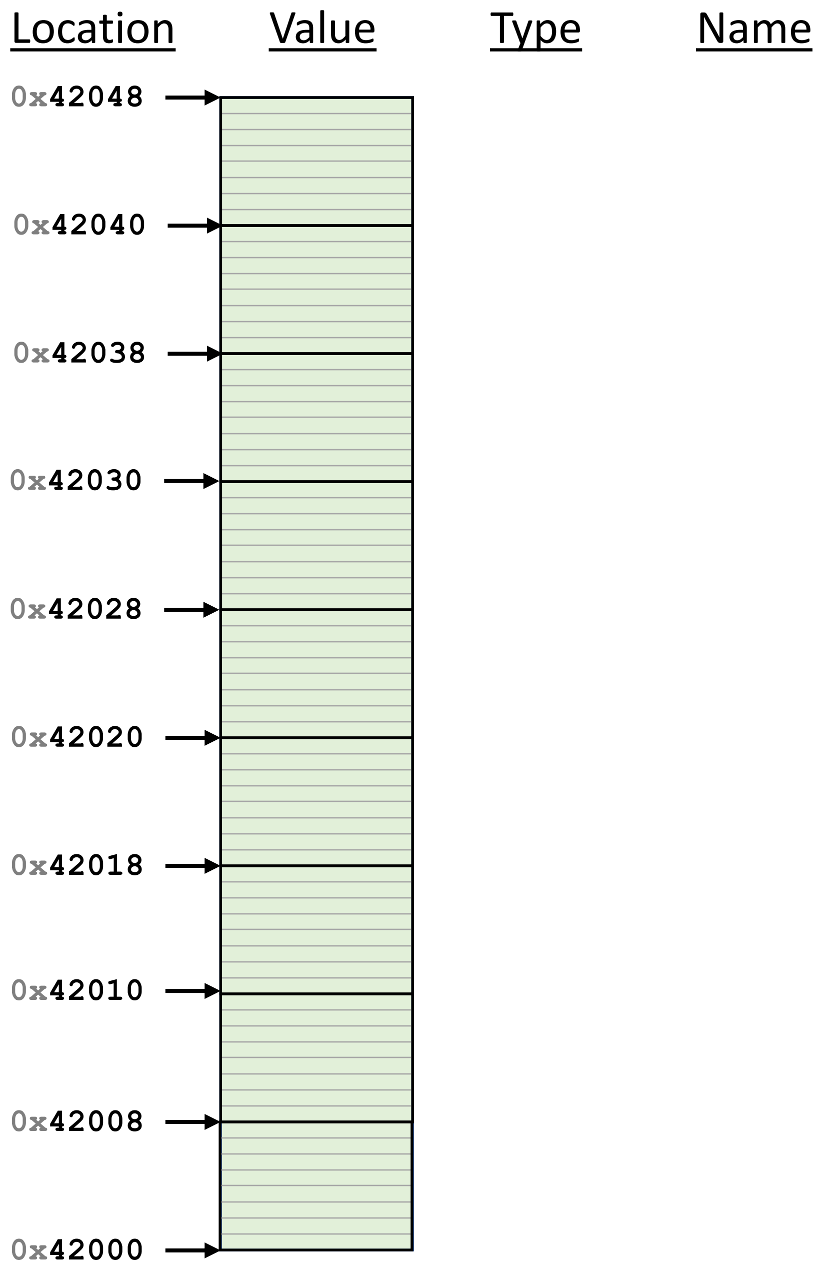
Heap Memory


0x42048
0x42040
0x42038
0x42030
0x42028
0x42020
0x42018
0x42010
0x42008
0x42000

heap1.cpp

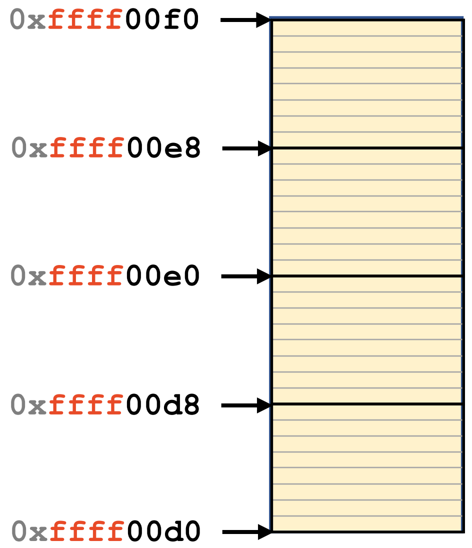
```
#include "sphere.h"
using namespace cs225;

int main() {
  int *p = new int;
  int *s = new Sphere(10);

  return 0;
}
```

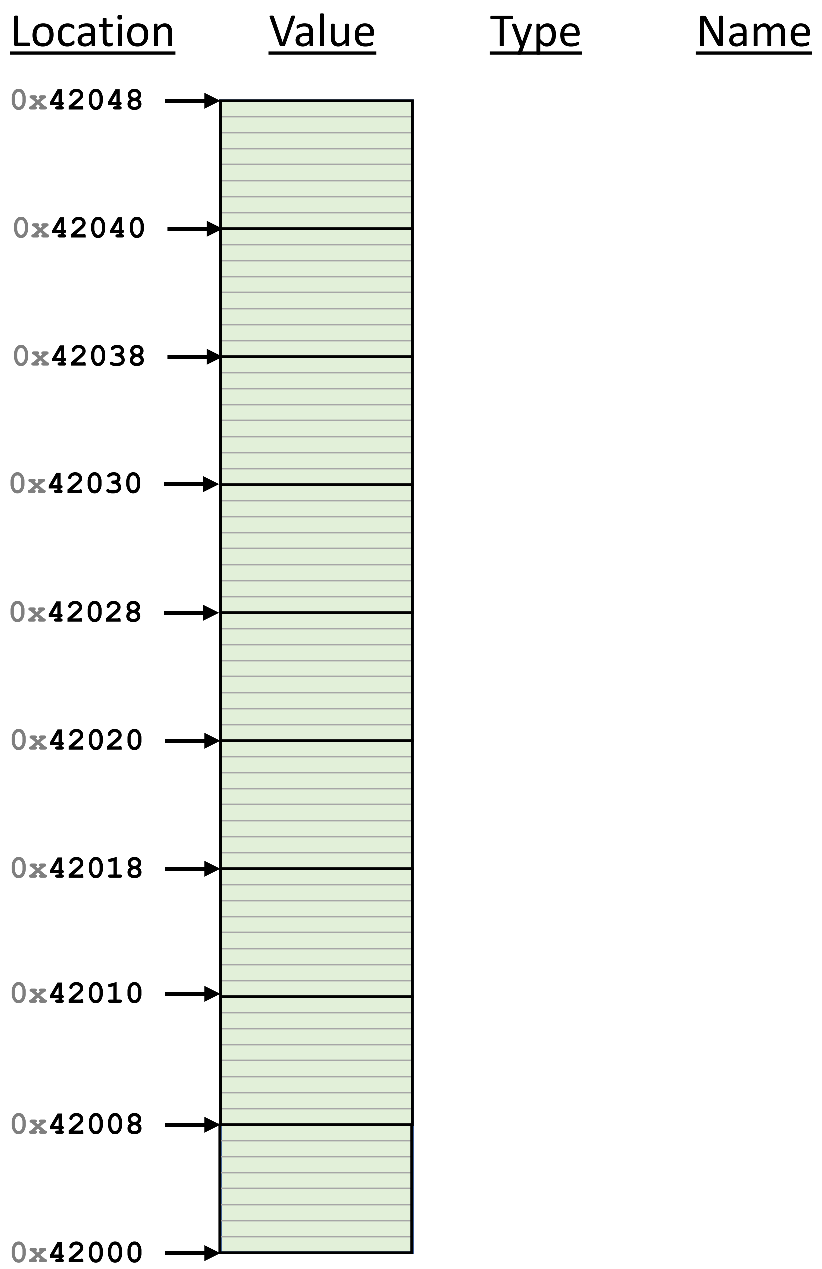




heap2.cpp

```cpp
#include "sphere.h"
using namespace cs225;

int main() {
  Sphere *s1 = new Sphere();
  Sphere *s2 = s1;

  s2->setRadius( 10 );

  return 0;
}
```





extra-puzzle1.cpp

```cpp
#include <iostream>
using namespace std;

int main() {
  int *p;
  int x;

  p = &x;
  x = 6;

  cout << x << endl;
  cout << p << endl;

  return 0;
}
```

extra-puzzle2.cpp

```cpp
#include <iostream>
using namespace std;

int main() {
  int *p, *q;
  p = new int;
  q = p;
  *q = 8;
  cout << *p << endl;

  q = new int;
  *q = 9;
  cout << *p << endl;
  cout << *q << endl;

  return 0;
}
```

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?*

***H*ave an awesome holiday weekend!**
…see you next Wednesday!