

#4: Heap Memory

September 4, 2019 \cdot *G Carl Evans*

Puzzle from last Friday:

puzzle.cpp		
4	Cube *CreateCube() {	
5	Cube c(20);	
6	return &c	
7	}	
8		
9	<pre>int main() {</pre>	
10	Cube *c = CreateCube();	
11	SomeOtherFunction();	
12	<pre>double v = c->getVolume();</pre>	
13	<pre>double a = c->getSurfaceArea();</pre>	
14	return 0;	
15	}	

Takeaway:

Heap Memory:

As programmers, we can use heap memory in cases where the *lifecycle* of the variable exceeds the lifecycle of the function.

1. The only way to create heap memory is with the use of the **new** keyword. Using **new** will:

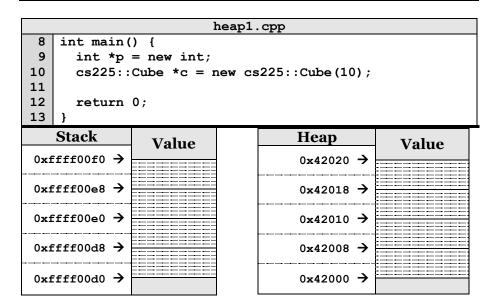
•

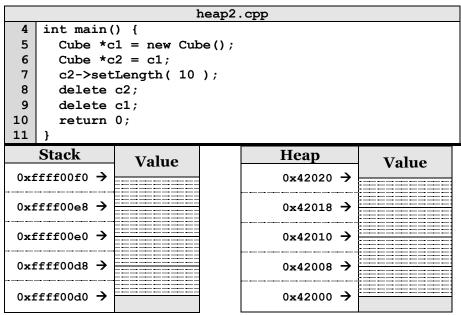
•

- 2. The only way to free heap memory is with the use of the **delete** keyword. Using **delete** will:
 - •

•

3. Memory is never automatically reclaimed, even if it goes out of scope. Any memory lost, but not freed, is considered to be "leaked memory".





Copying Memory – Deep Copy vs. Shallow Copy

	copy.cpp
6	int $i = 2$, $j = 4$, $k = 8$;
7	int $*p = \&i, *q = \&j, *r = \&k$
8	
9	$\mathbf{k} = \mathbf{i};$
10	cout << i << j << k << *p << *q << *r << endl;
11	
12	$\mathbf{p} = \mathbf{q};$
13	cout << i << j << k << *p << *q << *r << endl;
14	
15	*q = *r;
16	cout << i << j << k << *p << *q << *r << endl;

Consider how each assignment operator changes the data:

	Type of LHS	Type of RHS	Data Changed?
Line 8-9	i = p =	j = q =	k = r =
Line 11-12	i =	j =	k =
	p =	q =	r =
T			
Line 14-15	<u>i =</u> p =	j = q =	k = r =
		1 🔺	

Reference Variable

A reference variable is an <u>alias</u> to an existing variable. Modifying the reference variable modifies the variable being aliased. Internally, a reference variable maps to the same memory as the variable being aliased. Three key ideas:

1.

2.

reference.cpp				
3	<pre>int main() {</pre>			
4	int i = 7;			
5	int & j = i; // j is an <u>alias</u> of i			
6				
7	j = 4; // j and i are both 4.			
8	std::cout << i << " " << j << std::endl;			
9				
10	i = 2; // j and i are both 2.			
11	std::cout << i << " " << j << std::endl;			
12	return 0;			
13	}			

	heap-puzzle1.cpp				
6	<pre>int *x = new int;</pre>				
7	<pre>int &y = *x;</pre>				
8					
9	y = 4;				
10					
11	cout << &x << endl;				
12	<pre>cout << x << endl;</pre>				
13	cout << *x << endl;				
14					
15	cout << &y << endl;				
16	<pre>cout << y << endl;</pre>				
17	<pre>cout << *y << endl;</pre>				

	heap-puzzle2.cpp		
6	int *p, *q;		
7	<pre>p = new int;</pre>		
8	$\mathbf{q} = \mathbf{p};$		
9	*q = 8;		
10	<pre>cout << *p << endl;</pre>		
11			
12	q = new int;		
13	*q = 9;		
14	<pre>cout << *p << endl;</pre>		
15	<pre>cout << *q << endl;</pre>		

CS 225 – Things To Be Doing:

- **1.** Exam o starts on Thursday, know your time slot!
- 2. Finish up MP1 Due Monday, Sept. 9 at 11:59pm
- 3. Complete lab_debug this week in lab sections (due Sunday)
- **4.** POTDs are released daily, worth +1 extra credit point! ©