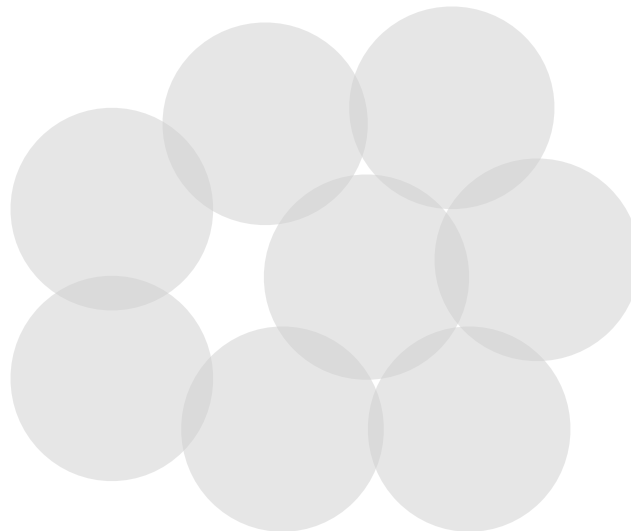
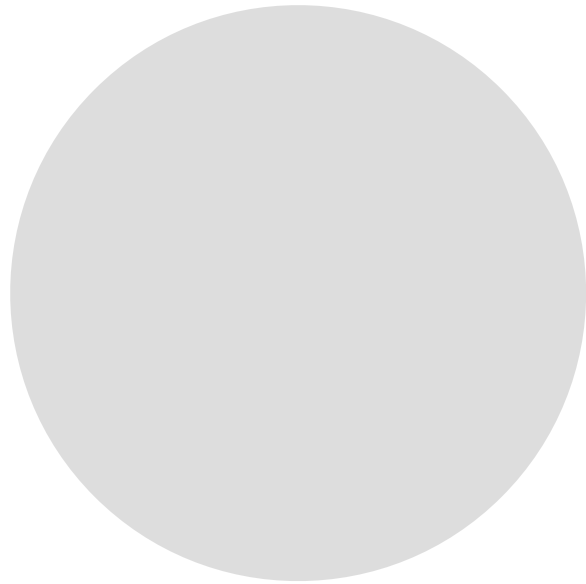


Announcements

HW0 available, due 8/29 before lecture.

MP1 available, due 8/30, 11:59p.

Exam 1: 9/6-9/9, <https://edu.cs.illinois.edu/testcenter/>



Encapsulation in C++:

1)

2)

Switching gears...



Configure your iMac 27-inch

Use the options below to build the system of your dreams

Memory

More memory (RAM) increases performance and enables your computer to perform faster and better. Choose additional 1066MHz DDR3 memory for your iMac.

[Learn more](#)

The more memory your computer has, the more programs you can run simultaneously, and the better performance you get from your computer.

- Select the standard memory configuration to support day-to-day tasks such as email, word processing, and web browsing as well as more complex tasks such as editing photos, creating illustrations, and building presentations.
- Upgrade your memory to enjoy greater performance for more intensive computation tasks, such as video editing and DVD authoring.

Your iMac uses one of synchronous dynamic without wasting clock

- ☒ 4GB 1066MHz DDR3
- ☐ 8GB 1066MHz DDR3
- ☐ 8GB 1066MHz DDR3
- ☐ 16GB 1066MHz DDR3



Variables and memory in C++

Stack memory

[illegible]

Pointers - Intro

```
int x;  
int * p;
```

How do we assign to p?

p =

p =

_____ operator: &

_____ operator: *

Stack memory

loc	name	value	type
a20	x	5	int
a40	p		int *

Pointer variables and dynamic memory allocation:

```
int * p;
```

Stack memory

loc	name	type	value
a40	p	int *	

Heap memory

loc	name	type	value

Youtube: [pointer binky c++](#)

Fun and games with pointers: (warm-up)

```
int * p, q;
```

What type is q? _____

```
int *p;
```

```
int x;
```

```
p = &x;
```

```
*p = 6;
```

```
cout << x;
```

What is output? _____

```
cout << p;
```

What is output? _____

Write a statement whose output is the value of `x`, using variable `p`: _____

```
int *p, *q;  
p = new int;  
q = p;  
*q = 8;  
cout << *p;   What is output?_____
```

```
q = new int;  
*q = 9;  
p = NULL;     Do you like this?_____
```

```
delete q;  
q = NULL;     Do you like this?_____
```

Memory leak:

Deleting a null pointer:

Dereferencing a null pointer:

Fun and games with pointers:

```
int * p, * q;
```

```
p = new int;
```

```
q = p;
```

```
delete p;
```

```
... // some random stuff
```

```
cout << *q;
```

Do you like this?_____



Stack vs. Heap memory:

```
void fun() {  
    string s = "hello!";  
    cout << s << endl;  
}  
  
int main() {  
    fun();  
    return 0;  
}
```

System allocates space for s and takes care of freeing it when s goes out of scope.

Data can be accessed directly, rather than via a pointer.

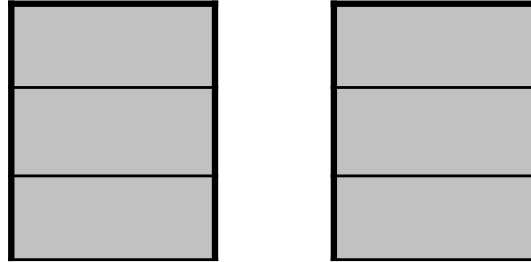
```
void fun() {  
    string * s = new string;  
    *s = "hello?";  
    cout << *s << endl;  
    delete s;  
}  
  
int main() {  
    fun();  
    return 0;  
}
```

Allocated memory must be deleted programmatically.

Data must be accessed by a pointer.

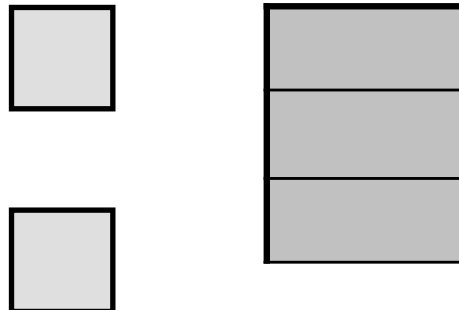
Pointers and objects:

```
face a, b;  
... // init b  
a = b;  
a.setName("ann");  
b.getName();
```



```
class face {  
public:  
    void setName(string n);  
    string getName();  
    ...  
private:  
    string name;  
    PNG pic;  
    boolean done;  
};
```

```
face * c, * d;  
... // init *d  
c = d;  
c->setName("carlos");  
(*d).getName();
```



Arrays: static (stackic)

```
int x[5];
```

Stack memory

[illegible]

Arrays: dynamic (heap)

```
int * x;
```

```
int size = 3;
```

```
x = new int[size];
```

```
for(int i=0, i<size, i++)
```

```
x[i] = i + 3;
```

```
delete [] x;
```

Stack memory

[illegible]

Heap memory

[illegible]