Announcements

HW0, MP1 available, due 1/27, 11:59p.

```
class sphere{
public:



void setRadius(double newRad);
double getDiameter() const;
…
private:
double theRadius;
};
```

```
//constructor(s) (next page)
void sphere::setRadius(double newRad){
   if (newRad > 0)
       theRadius = newRad;
}
double sphere::getDiameter() const {
   return 2*theRadius;
}
```

```
int main(){



}
```

Constructors (intro): When you *declare* a sphere, a sphere class constructor is invoked.

Points to remember abt ctors:

1.

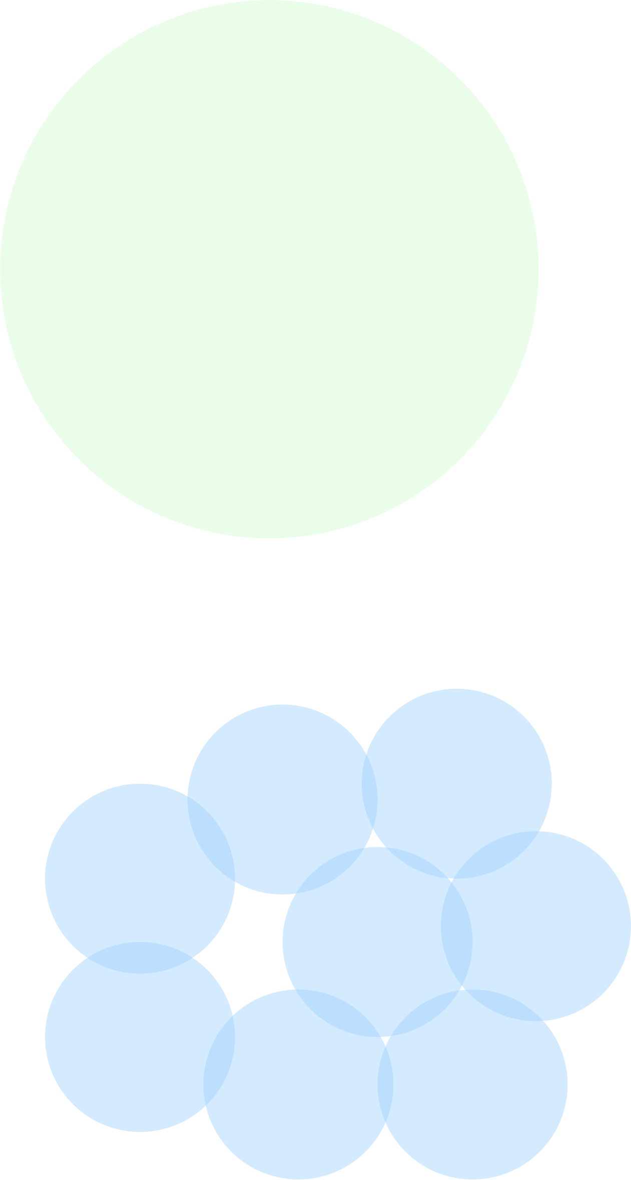
2.

3.

```
int main(){



}
```

```
…
//default constructor
sphere::sphere(){
       theRadius = 1.0;
}

//default constructor, alternative
sphere::sphere(): theRadius(1.0)
{

}

//constructor with given radius
sphere::sphere(double r)
{
       if (r>0)
              theRadius = r;
       else theRadius = 1.0;
}
...
```

Class Definition… where are we?



Encapsulation in C++:

1)

2)

Recap…

Ideas/concepts:

- Class definitions
- Class function implementation
- Constructors
- Clients

OOP: we now understand how C++ supports

- Inheritance
- Encapsulation (separation of interface from implementation)
  1)

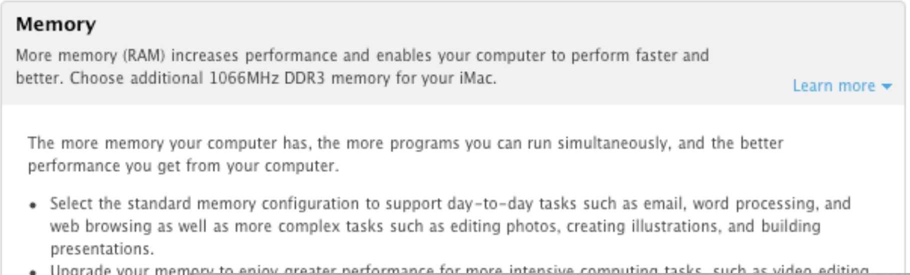
  2)

- Polymorphism

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 computing tasks, such as video editing and DVD authoring

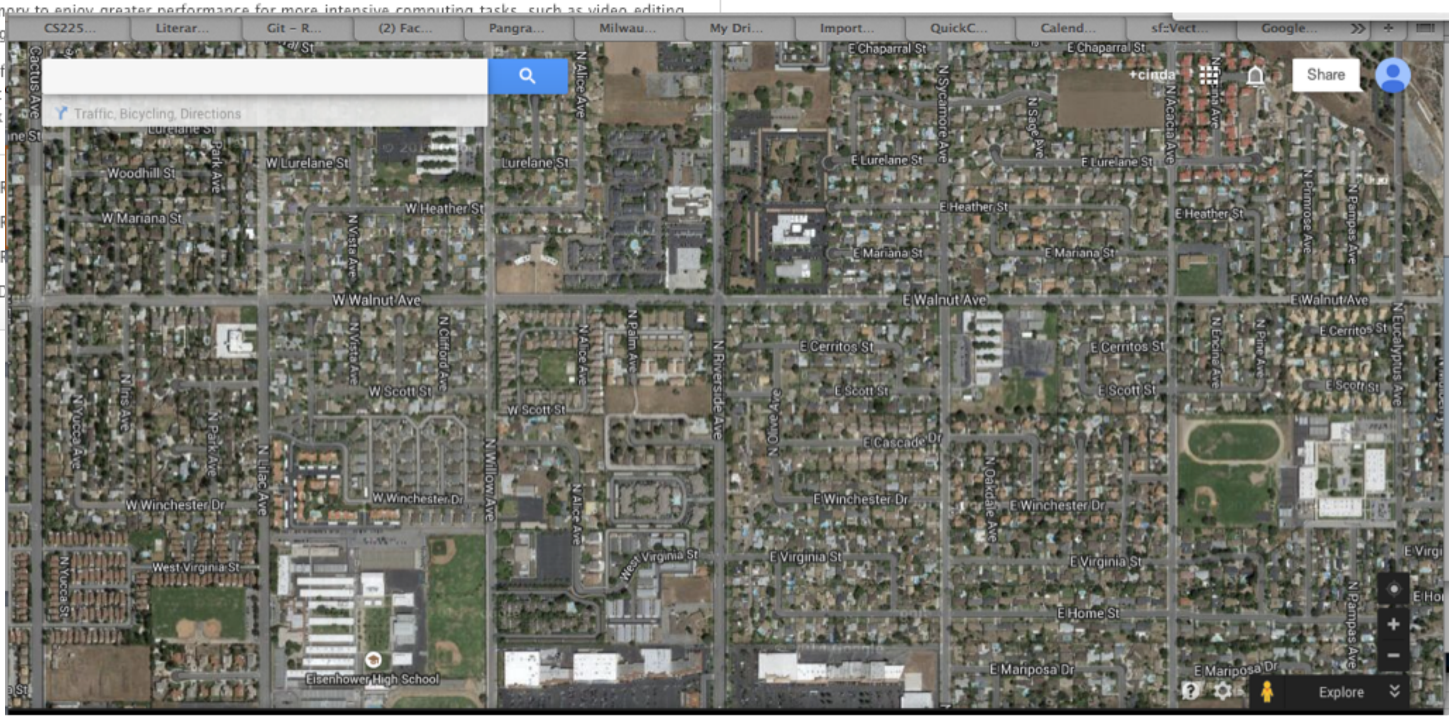
Your iMac uses one of
synchronous dynamic
without wasting clock

● 4GB 1066MHz DDR

○ 8GB 1066MHz DDR

○ 8GB 1066MHz DDR

○ 16GB 1066MHz DD



Variables and memory in C++

Stack memory

loc	name	value	type

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:

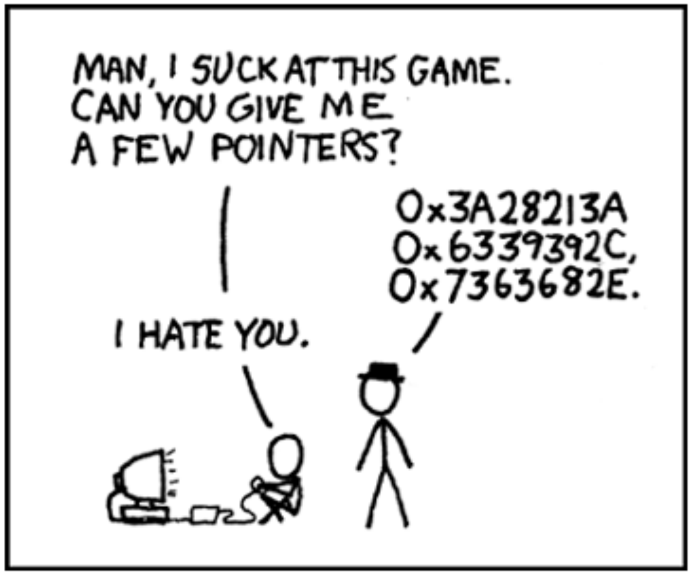
Deleting a null pointer:

Dereferencing a null pointer:

Fun and games with pointers:

```
int * p, * q;

p = new int;

q = p;

delete p;

… // some random stuff

cout << *q;
```

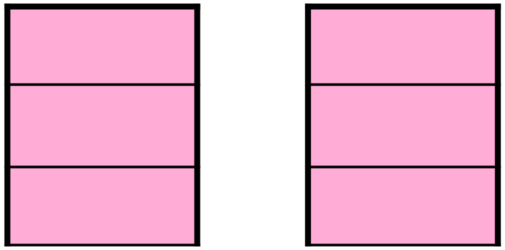
Do you like this?______________



Pointers and objects:

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

```
face a, b;
… // init b
a = b;
a.setName(“ann”);
b.getName();
```



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