Announcements

HW0 available, due 8/29 before lecture.
MP1 available, due 9/2, 11:59p.

class sphere{
public:

    void setRadius(double newRad);
    double getDiameter() const;

private:
    double theRadius;
};

void sphere::setRadius(double newRad){
    radius = newRad;
}
double sphere::getDiameter() const {
    return 2*radius;
}

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(){
}

//default constructor, alternative
sphere::sphere(){
}

//constructor with given radius
sphere::sphere(double r){
}

...
Class Definition… where are we?

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.

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 the fastest memory technologies available today—1066MHz, Double Data Rate (DDR3), synchronous dynamic random-access memory (SDRAM)—ensuring that the processor is constantly fed with data without wasting clock cycles.

- 4GB 1066MHz DDR3 SDRAM – 2x2GB
- 8GB 1066MHz DDR3 SDRAM – 4x2GB [Add $200.00]
- 8GB 1066MHz DDR3 SDRAM – 2x4GB [Add $600.00]
- 16GB 1066MHz DDR3 SDRAM – 4x4GB [Add $1,400.00]
Variables and memory in C++

<table>
<thead>
<tr>
<th>loc</th>
<th>name</th>
<th>value</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pointers - Intro

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

How do we assign to p?

```c
p =
```

```c
p =
```

____________ operator: &

____________ operator: *

Stack memory

<table>
<thead>
<tr>
<th>loc</th>
<th>name</th>
<th>value</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>a20</td>
<td>x</td>
<td>5</td>
<td>int</td>
</tr>
<tr>
<td>a40</td>
<td>p</td>
<td></td>
<td>int *</td>
</tr>
</tbody>
</table>
Pointer variables and dynamic memory allocation:

```c
int * p;
```

<table>
<thead>
<tr>
<th>loc</th>
<th>name</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a40</td>
<td>p</td>
<td>int *</td>
<td></td>
</tr>
</tbody>
</table>

Stack memory

Heap memory

Youtube: pointer binky c++
Fun and games with pointers: (warm-up)

```cpp
int * p, q;  // What type is q?______________
```

```cpp
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: ____________