

# CSCI261A

Lab 6: `strings`, `LetterFrequency`, `PowerPlantDailyAverage`  
October 2, 2009

# OOP Review

- Remember, *objects* encapsulate attributes (values) and behavior (functions) of real world things.
- Remember, our car example
- A *class* defines the rules that *objects* must follow
- Objects are instances of classes
  - Each one of you is an instance of homo-sapien

# Class v. Object

**class** Homo-Sapien

```
int age;  
string name;  
void walk();  
void burp();
```

**object** DavidHasselhoff

age: 61

name: "David Hasselhoff"

burp()

"uuurrrrrpp!"

walk()

(he moves with confident gait)

**object** RachelRay

age: 42

name: "Rachel Ray"

burp()

"rrrruuuuuuuhhp!"

walk()

(she moves quickly)

# You've Been Using an Object

`string`

# String Class v. String Object

class string

- has a value (a bunch of characters)

- has behavior

  - size(); // tell me how long you are

  - empty(); // tell me if you are empty

  - substr(); // tell me part of you

  - c\_str(); // describe me in c-style

object

“Just show me  
some code man”

object

“He’s wearing the  
same jeans today”

object

“Did he really just  
say that?”

# String Class v. String Object

The class just defines what values its objects can have, and what functions/behaviors its objects have.

In general, we work with objects only (to get stuff done)...  
... but the computer uses classes (to give us our objects).

# string

```
#include<string>
// ...
string mycar;
string mfg = "Lamborghini";
string model = "Countach";
int x;
bool y;
string z;

x = model.size(); // x = ?
y = mycar.empty(); // y = ?
mycar = mfg + model; // mycar = ?
z = mfg.substr(0, 5); // z = ?
```

# LetterFrequency

# PowerPlantDailyAverage

# Homework

- Read chapter 6.9 (p293 - 294)
- Complete assignments *LetterFrequency* and *PowerPlantDailyAverage*