CS 162
Intro to CS II

“Has a” vs. “Is a” Relationship
What is inheritance?

• Webster Definition?
  – the reception of genetic qualities by transmission from parent to offspring
  – the acquisition of a possession, condition, or trait from past generations

• CS Definition?
  – Base class (Parent) and Derived class (Child)
  – Ancestor class and Descendant class (generations)
Inheritance Interface

```cpp
class parent {
    public:
        parent(); // Have a constructor
        void print_mssg();
        int get_shared_var();
    private:
        int shared_var;
};
class child : public parent {
    public:
        child(); // This constructor needs to call parent() constructor
        void print_mssg(); // Redefine or Override inherited function
    private:
        int unique_var;
};
```
Inheritance Implementation

parent::parent() {
    shared_var = 0;
}
int parent::get_shared_var() {
    return shared_var;
}
void parent::print_mssg() {
    cout << “I’m parent!” << endl;
}

//child class implementation
child::child() : parent() { //Need to call inherited constructor first
    unique_var = 0;
}
void child::print_mssg() {
    cout << “I’m child!” << endl; //This will take precedence over parent
}
What is not inherited?

- Constructors
- Destructors
- Friends
- Assignment Op Overload

- Inherited, but not accessible: Private Members
Protected vs. Private Inheritance

• Public, protected, and private members
  – Who has access to these?

• Public, protected, and private inheritance
  – What does this mean?
    class child : public parent { ... }
    class child : protected parent { ... }
    class child : private parent { ... }

Anybody outside the class
inherited members have direct access
only accessible in the class and from friends
Public members are now protected
Public & protected become private
Protected vs. Private Inheritance

• Protected
  – Public members are protected in child

• Private
  – All members are private to child

• Either case:
  – Breaks “is a” relationship
Demo: Vehicle Toll ...

• Get into groups of 4-5 people
• Design the classes for a vehicle and bike to provide the toll amount based on the wheels for all vehicles, except bikes that are free.
  – Mutator to set wheels
  – Accessor function for the wheels
  – Provide toll amount for vehicles and bikes
• How will you make sure it is working?
```cpp
#ifndef VEHICLE_H
#define VEHICLE_H

class Vehicle {
  private:
    int wheels;
  public:
    int get_wheels();
    void set_wheels(int);
    int get_toll();
};
#endif
```

```cpp
#include "Vehicle.h"

int Vehicle::get_wheels() {
  return wheels;
}

void Vehicle::set_wheels(int w) {
  wheels=w;
}

int Vehicle::get_toll() {
  return 1*wheels;
}
```
```cpp
#include "Vehicle.h"

#ifndef BIKE_H
#define BIKE_H

class Bike : public Vehicle {
    public:
        int get_toll();
};
#endif

#include "Bike.h"

int Bike::get_toll() {
    return 0;
}
```
```cpp
#include "Vehicle.h"
#include "Bike.h"
#include <iostream>
using namespace std;

int main()
{
    Vehicle v;
    Bike b;

    v.set_wheels(4);
    b.set_wheels(2);

    cout << "Vehicle Toll: " << v.get_toll() << endl;
    cout << "Bike Toll: " << b.get_toll() << endl;
}
```