CS 162, Lecture 12: Pitfalls and Intro Inheritance

26 April 2018
Pitfalls

- Student s();
  Actually it is a prototype!

- Student x = s;
  Gut reaction: AOE
  Actually copy constructor

- If we only do shallow copy in Copy Constructor and Assignment Operator Overload

  Assume AOE only uses shallow

  \[ s_1 = s_2 \]
Inheritance

• The process by which a new class is created from another class
• Base (Parent) class == more general class which derived class are created from
• Derived (Child) class == new class
  • Has all of the member variables and functions as base class
• Examples:

  Parent: Animal
  Child: Dog, Cat, etc.

  Parent: Rectangle
  Child: Square

  Parent: Fruit
  Child: Apple
Defining Inheritance

```cpp
class Dog : public Animal {
};
```

- class Derived:public Base {};

- In the Derived class:
  - List only member variables you want to add, not what is inherited
  - Only redeclare inherited member functions if you want to redefine them
    - Redefining is when an inherited member function definition is changed in the derived class

- Derived classes can be used anywhere the base class would be used but not the other way around
Not Inherited

- Base class constructor – though it can be invoked from the derived class
  - Child::Child():Parent() {}
  - Base is called first to initialize all of the base member variables
  - If base constructor isn’t specified the base default constructor will be used

- Copy constructor
- Assignment Operator Overload
- Destructor
Inherited but Restricted

• Private member variables are inherited but cannot be accessed by name, need to use the accessor and mutator functions
• Private member functions are inherited but cannot be accessed by the derived class
• Protected access modifier allows for the derived class to be able to access things by name but every other class would view them as private