Pitfalls

• Student s();

• Student x = s;

• If we only do shallow copy in Copy Constructor and Assignment Operator Overload
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:
Defining Inheritance

• class Derived:public Base {};

• In the Derived class:
  • List only member variables you want to add, not what is inherited
  • Only redefine 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