C++ Derived Classes

Introduction to Programming

1/07/2016

C++ Derived Classes

- Example:
  - exs1.cpp
    - two classes for employee and manager
  - exs2.cpp
    - derived class
  - exs3.cpp
    - derived class, assigning base to derived

Derived Classes

- Support object oriented programming
- Enable relationship between different object/classes
- Example

```
class Employee {  // for all employees
  string name;
  int eNumber;
};
// class Manager {  // For managers
// string name;
// int eNumber;
// int level; // addition attributes
// };
class Manager: public Employee {  // using inheritance
  int level
};
```

Derived Classes

- A manager is an employee
- Manager class is derived from Employee class
- Derived class (manager) vs. Base Class (Employee)
  - Subclass vs Superclass
- Manager class is inherited from Employee class
- A derived class variable can be assigned to a base type variable, but not the other way around
- Assigning base to derived must be explicit.

Member Data and Functions

- A member of a derived class can use the public and protected members, both data and functions, of its base class as if they were declared in the derived class itself.
- However, a derived class cannot use a base class' private names
- Protected data can be accessed by derived classes.

Constructors and Destructors

- If base class has a constructor, then a constructor must be invoked
- If all base constructors require arguments, then a constructor for the base must be explicitly called
- A derived class constructor can specify initializers for its own members and the immediate bases only; it cannot initialize members of a base directly
- Class objects are constructed from the bottom up: first the base, then the members and then the derived class itself
- They are destroyed in the opposite order: first the derived class, its member then the base.
- Members and bases are constructed in order of declaration in the class and destroyed in the reverse order

Copying

- Derived objects can be assigned (copied) to a base object, but only the base members are copied (slicing)
- Assignment operators are not inherited
- Constructors are never inherited

Class Hierarchies

- A derived class can itself be a base class
  - Class hierarchy
- A class can be derived from multiple base classes
- Example:

```
class T1: public S1, public S2 { };
```

Virtual Functions

- Virtual function overcome the problem of type-field by allowing programmers to declare functions in a base class that can be redefined in each derived class.
- A virtual function is sometimes called a method.
- Virtual functions support polymorphism
- To get polymorphic behavior in C++, the member function call must be virtual and objects must be manipulated through pointers or references.
- Example: exs4.cpp

Summary

- Related classes
- Derived classes
- Class hierarchy
- Virtual functions