Inheritance: The Fundamental Functions

Lecture 22
Chapter 1

Robb T. Koether

Hampden-Sydney College

Fri, Mar 6, 2009
Outline

1. Inheritance of Constructors
2. Inheritance of Destructors
3. Inheritance of the Assignment Operator
4. Assignment
Inheritance of Constructors

Constructor Rules

1. A derived-class constructor will automatically invoke the base-class default constructor, unless instructed otherwise.
2. We may instruct the derived-class constructor to invoke a specific base-class constructor.
3. The base-class constructor is invoked before the derived-class constructor is executed.
Example (A List Hierarchy)

- Suppose we create a `List` base class and then derive an `ArrayList` class and a `LinkedList` class from it.
Inheritance of Constructors

Example (A List Hierarchy)

- The `ArrayList` and `LinkedList` classes have `mSize` in common.
- Therefore, we could put `mSize` in the `List` base class.
Inheritance of Constructors

Example (A List Hierarchy)

- When we construct an `ArrayList` using
  ```java
  ArrayList list(5, 123);
  ```
  what would happen?
- The `ArrayList` constructor will first call a `List` constructor which will initialize `mSize`.
- Then it will initialize `element`. 
Inheritance of Constructors

Example (A List Hierarchy)
- When we construct an `ArrayList` using
  ```java
  ArrayList list(5, 123);
  ```
  what would happen?
- The `ArrayList` constructor will first call a `List` constructor which will initialize `mSize`.
- Then it will initialize `element`. 
Inheritance of Constructors

Example (A List Hierarchy)

- When we construct an `ArrayList` using
  ```java
  ArrayList list(5, 123);
  ```
  what would happen?
- The `ArrayList` constructor will first call a `List` constructor which will initialize `mSize`.
- Then it will initialize `element`. 
Example (A List Hierarchy)

- **Which** `List` constructor will it call?
- Unless we specify otherwise, it will call the default `List` constructor.
- Uh-oh.
Inheritance of Constructors

Example (A List Hierarchy)

- Which `List` constructor will it call?
- Unless we specify otherwise, it will call the default `List` constructor.
- Uh-oh.
Inheritance of Constructors

Example (A List Hierarchy)

- Which `List` constructor will it call?
- Unless we specify otherwise, it will call the default `List` constructor.
- Uh-oh.
Base Class Constructor

```java
ArrayList(parameters) : List(parameters)
{
    body of the ArrayList constructor
}
```

- We may specify other constructors through an initializer.
Invoking the Base Class Constructor

Base Class Constructor

```cpp
ArrayList(int sz, int value) : List(sz)
{
    if (mSize == 0)
        element = NULL;
    else
    {
        element = new int[mSize];
        for (int i = 0; i < mSize; i++)
            element[i] = value;
    }
    return;
}
```
Inheritance of Destructors

Destructor Rules

1. The derived-class destructor automatically invokes the base-class destructor.
2. The base-class destructor is invoked after the derived-class destructor is executed.
Assignment Operator Rules

1. The automatic assignment operator invokes the assignment operator for the base class.

2. A programmer-defined assignment operator must copy the base-class members. This is a problem if the base-class members are private.
Inheritance of the Assignment Operator

One can use the scope operator :: to invoke the assignment operator of the base class.
Inheritance of the Assignment Operator

One can use the **scope operator** `::` to invoke the assignment operator of the base class.
Inheritance: The Fundamental Functions

Robb T. Koether

Inheritance of Constructors
Inheritance of Destructors
Inheritance of the Assignment Operator
Assignment

Inheritance of the Assignment Operator

```c++
class A { protected: int aMem; };
class B : public A { private: int bMem; };
B& B::operator=(const B& b) {
    aMem = b.aMem // Legal
    bMem = b.bMem;
}
```

- One can use the `scope operator ::` to invoke the assignment operator of the base class.
Inheritance: The Fundamental Functions

Robb T. Koether

Inheritance of Constructors

Inheritance of Destructors

Inheritance of the Assignment Operator

Assignment

Inheritance of the Assignment Operator

Example (Subclass of `ArrayList`)

- Suppose we create a class `ArrayListwPos`, which is an `ArrayList` that keeps track of a current position.

- `ArrayListwPos` has one additional data member: `int currPos;`
Inheritance of the Assignment Operator

Example (Subclass of `ArrayList`)

```cpp
ArrayListwPos& operator=(const ArrayListwPos& list)
{
    if (this != &list)
    {
        ArrayList::operator=(list);
        currPos = list.currPos;
    }
    return *this;
}
```
Example

Example (Inheritance)

- Create the following classes:
  - Father
  - Man
  - MarriedMan
  - MarriedWoman
  - Mother
  - Person
  - Woman
Example (The IS-A Relation)

Person

Man

Married Man

Father

Woman

Mother

Married Woman
Example (The HAS-A Relation)

Example

Inheritance:
The Fundamental Functions
Robb T. Koether

Inheritance of Constructors
Inheritance of Destructors
Inheritance of the Assignment Operator
Assignment

Example (The HAS-A Relation)

Person

Man

Woman

Married Man

Father

Mother

Married Woman
Example (Inheritance)

- **Header files**
  - `person.h`
  - `man.h`
  - `woman.h`
  - `father.h`
  - `mother.h`
  - `marriedman.h`
  - `marriedwoman.h`
Example (Inheritance)

- Programs
  - DefaultPeople.cpp
  - FamilyTree.cpp
Assignment

Inheritance: The Fundamental Functions
Robb T. Koether

Homework

Read Section 14.3, pages 794 - 801.