CS2136: Paradigms of Computation

Class 12: Inheritance

Copyright 2001, 2002, 2003 Michael J. Ciaraldi and David Finkel
Inheritance
Inheritance

Java only supports **single** inheritance.

By default, each subclass inherits all data and methods from the parent class.

Can add data.

Can add and/or override methods.

Syntax example:

```java
    class X extends Y
    
    X = subclass
    Y = superclass
```
All classes extend the standard class `Object`. Some of the methods in the class called `Object`:

- `toString()`
- `clone()`
- `equals()`
- `wait()`, `notify()`  [for threads]
- `getClass()`

You may have to override some of them.
Original Inheritance Example (Eckel, Chap. 6)
Inheritance Example

// OriginalArt/Cartoon.java
class Art { // Base class
    Art() { System.out.println("Art constructor"); }
}
class Drawing extends Art { // Second level
    Drawing() { System.out.println("Drawing constructor"); }
}
Inheritance Example Cont.

```java
public class Cartoon extends Drawing { // Third level
    Cartoon() { System.out.println("Cartoon constructor"); }
    public static void main(String[] args) {
        Cartoon x = new Cartoon();
    }
}
```
Output

Art constructor
Drawing constructor
Cartoon constructor

z Why?

y By default, each constructor invokes the constructor of its parent.
class Drawing extends Art {
    Drawing() { System.out.println("Drawing constructor"); }
}

public class Cartoon extends Drawing {
    Cartoon() { System.out.println("Cartoon constructor"); }
    // Next line is the main method, executed first.
    public static void main(String[] args) {
        Cartoon x = new Cartoon();
    }
}
Construction From the Base Class Outward II

- If you want to invoke the constructor of your superclass explicitly, that call must be the first executable statement of the constructor.

- If you don’t, the compiler inserts one automatically.
Visibility (Access Specifiers) for Variables and Methods

**“friendly”** (default)
- Visible to other classes in same package.
- If not in a package, visible to other classes in same directory.

**public**
- Visible to all.

**private**
- Only visible within same class (same object, too).

**protected**
- Visible to same class and its subclasses.
Simple Inheritance Example

Art

String owner
private String Address

Drawing

double length
double width
Simple Inheritance Example (too big to fit)

class Art {
    String owner; private String address;
}
class Drawing extends Art {
    double length, width;
}

public static void main(String[] args) {
    Drawing b = new Drawing("George W. Bush");
    Drawing c = new Drawing("Al Gore", 20.5, 31.768);
    // System.out.println("Owner = " + owner);
    System.out.println("Owner = " + c.owner);
    // System.out.println("Address = " + c.address);
}
More on Visibility

- If a class extends a class from another package, it can only access the public methods.

- A class can only be “friendly” (default) or public, not private or protected.

- Can only have one public class per compilation unit (file); must have same name as file.
Enhanced Inheritance Example

Art
- Person owner
- Date creation

Drawing
- Size dimensions

Cartoon
- Person penciller
- Person inker
- Person colorist

Sketch
- Person artist
Enhanced Inheritance Example

- Too big to fit on the screen.
- See Handout
Enhanced Inheritance
Example Output

Art constructor
Drawing constructor
Cartoon constructor

No toString() defined for Cartoon!
Cartoon@72e62e3f
Owner of x = Person@71ce2e3f
Owner of x = Bill Gates
Art constructor
Drawing constructor
Cartoon constructor
Owner of y = Ed Parrish
Next Time

- More Java:
  - Leftovers on Inheritance
  - Arrays, Strings, and other Collections