Csc18a Chapter 9 Notes

**Inheritance**

Topics

* Inheritance
  + Superclass
  + Subclass
* Direct/ indirect superclass
* Class hierarchy
* Protected members

Inheritance

* Overall, similar to C++
* A new class acquires an existing class’s members and possibly use them with new or modified capabilities.
* The idea of having a (1) class inherit the attributes (member variables) and (2) methods of another class (Teacher def.)
  + 1) Subclass / child class
  + 2) Superclass / Parent class
    - Direct
      * The subclass that direct inherits from the parent
    - Indirect
      * Any class above the direct superclass in the hierarchy
* Java only supports single inheritance (*No multiple inheritance*)
* Syntax:
  + **Public class ChildClass extends Parentclass{**

//code…

**}**

Access Modifiers

* Public
  + Visible to all
* Private
  + Visible only to class
* Protected
  + Visible to parent class, child class, and package (if applicable)
  + Good practice to use when creating variables/methods that need to be accessed through inheritance.
  + Warning:
    - If a parent has a protected variable and data checks for errors, that does not mean that the child data checks!
    - Thus, incorrect data can sometimes go through the child

Subclasses constructors

* Constructors are *not* inherited
  + Must create a constructor
* First task of any subclass constructor is to call its direct superclass’s constructor
  + If not called, Java calls default constructor
  + If no default constructor exists, error will occur
* Explicit superclass call
  + Keyword: **super**
    - Calls the super class constructor
  + Syntax:
    - **super(arguments);**
  + Must be the first statement in the constructor’s body.

**@Override** annotation

* Indicates that the following method declaration should override an existing superclass method
* If a method for the superclass does not exist, an error will occur
  + Example:

**@Override**

**Public String toString(){**

//code…

**}**