Inheritance (Subclasses)

(in Java)

see Reges & Stepp chapters 9, 10, 11
Interfaces vs Inheritance

We saw:

```java
class foo implements bar { }
```

bar is an *interface*; foo is an *implementing class*

We can also extend or “inherit from” classes

```java
class foo extends zot { }
```

zot is a *superclass*; foo is a *subclass*

We inherit to modify and extend an *implementation*
Like that one, but ...

class Shape {
    public abstract int area();
}

class Rect extends Shape {
    public int area() {
        return height * width;
    }
    ...
}

class Square extends Rect {
    public Square(int width) {
        super(width, width);
    }
}
```
CIS 210

Shape[
  class
  height
  width
]
  super
  area
Square
  super
  area
Rect
  super
  area
Triangle
  super
  area
```
In project 7

Class Agenda is given
Class AvailTime should be a subclass of Agenda
   The “protected” field appts can be used in AvailTime

public class Agenda {
    protected ArrayList<Appt> appts;
    ...
}
You will need to add

Methods used in the main program Avail.java

AvailTime avail = new AvailTime(beginWindow, endWindow);
for (int i=2; i < args.length; ++i) {
    ...
    FileInputStream is = new FileInputStream(args[i]);
    SchedReader reader = new SchedReader(is);
    Agenda agenda = reader.readAgenda();
    System.out.println("Removing conflicts from " + args[i] + ":");
    System.out.println( agenda.toString() );
    avail.removeConflicts(agenda);
    ...
    System.out.println("Leaving " + avail.blocksRemaining() + " times in common");