Today’s topic:

- Using Inheritance
OO Concept: Inheritance

- Objects in the real world are often a type of or specialization of another object.

- We would say these objects are a subclass of the base class.

- Inheritance is an “is-a” relationship.
  - A dog “is-a” mammal.

OO Concept: Inheritance

- Our base class (or super class) is a generalization.
  - It contains attributes and behaviors that are common to all of the sub classes.

- Our sub classes are specializations.
  - They contain specialized functionality.
OO Concept: Inheritance

- Two types of specializations:
  - Do the same behavior as the super class, with a different implementation.
  - Additional behavior that the super class cannot do.

Generalization and specialization
UML Class Diagram

- Inheritance Relationship
  - Used to represent that one class is subclass of another.
  - The Dog and Cat classes are subclasses of Mammal.

---

OO Concept: Inheritance

- Let’s look at a real world example:
  - *This is a knife.*
    - It has a blade
    - It has a handle
    - It can cut
Let’s look at a real world example:

- This is a switchblade knife
  - It has a blade.
  - It has a handle.
  - It can cut.
  - It folds up.


Let’s look at a real world example:

- This is a Swiss Army knife
  - It has a blade.
  - It has a handle.
  - It can cut.
  - It folds up.
  - It has a can opener.
  - It has a screwdriver.
  - It has a bottle opener.
  - It has a wire stripper.
  - It has a leather punch.

Image Source: http://www.swissknifeishop.com/swiss-army-tinker
OO Concept: Inheritance

Let’s look at a real world example:

- The switch blade and Swiss Army are types of knives.

- They share the basic features of a knife, but also have additional ones.

Inheritance

- Super Class

- Sub class

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>Method3</th>
<th>Method4</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance variables</td>
<td>private methods</td>
<td>protected methods</td>
<td>New private methods</td>
</tr>
<tr>
<td>Overwritten</td>
<td>New</td>
<td>New instance variables</td>
<td>New protected methods</td>
</tr>
<tr>
<td>New</td>
<td>New</td>
<td>Overwritten</td>
<td>New</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>Method3</th>
<th>Method4</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance variables</td>
<td>private methods</td>
<td>protected methods</td>
<td>New private methods</td>
</tr>
<tr>
<td>Overwritten</td>
<td>New</td>
<td>New instance variables</td>
<td>New protected methods</td>
</tr>
<tr>
<td>New</td>
<td>New</td>
<td>Overwritten</td>
<td>New</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>Method3</th>
<th>Method4</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance variables</td>
<td>private methods</td>
<td>protected methods</td>
<td>New private methods</td>
</tr>
<tr>
<td>Overwritten</td>
<td>New</td>
<td>New instance variables</td>
<td>New protected methods</td>
</tr>
<tr>
<td>New</td>
<td>New</td>
<td>Overwritten</td>
<td>New</td>
</tr>
</tbody>
</table>

New Instance Variables

Method1 Overwritten

New private methods

New protected methods
OO Concept: Inheritance

- Some vocabulary:
  - The base class is called the super class (i.e. knife).
  - The sub class is the specialization (i.e. folding knife).
  - The sub class inherits instance variables and methods from the super class.

OO Concept: Inheritance

- Using inheritance in your design.
  - Sometimes you will know in advance that you will use inheritance.
  - Other times you may anticipate future inheritance.
  - Even if you don’t use inheritance in your first iteration, it is a good idea to design with it in mind.

- The book gives an example of a system for a veterinarian.
  - You may not treat cats now, but sometime in the future you may want to.
OO Concept: Inheritance

- Some vocabulary:
  - You will find the term polymorphism used in different ways in OOP. We’ll talk about one way today.
  - The word comes from poly = many and morph = forms.
  - In general polymorphism, an objects of different subtypes can be used in place of each other in your code.

OO Concept: Inheritance

- We have already seen and used sub classes in our examples.
- In Java, all classes are a subclass of the base class Object.
- That’s why all objects have a toString() method. They inherit it from the class Object.
- You will also see that many methods take an Object as their input parameter. This means you can literally pass in any instance of any Object.
Rules for overriding a method

- The argument list has to be the same as that of the overridden method.
- The return type has to be the same or a subtype.
- The access level cannot be more restrictive than the overridden method's access level.
- A method declared final cannot be overridden.
- A method declared static cannot be overridden but can be re-declared.
- Constructors cannot be overridden.