# Class Hierarchies

**Big Ideas**

**Products can be designed for life cycle**

This activity will take you through the process of designing superclasses and subclasses. As you do so, consider carefully how your superclass could be reused and repurposed, later, by another programmer to create something different.

Many of the programs you create could be altered, remixed and tweaked to create new and innovative software that is slightly different than the original. This is a great thing about Object Oriented Programming. Much of the code and be reused and repurposed to create things we haven’t even thought of yet.

With your partner, complete the steps outlined on this sheet to construct model code to share with the rest of the class. You should use this worksheet to draft and correct your ideas; you can write a final, neat version on the lined paper that has been handed out to you.

## Step 1

Take a moment to brainstorm some real-life objects or situations that have a hierarchical relationship. In the space below, draw a hierarchy with 2 or 3 tiers or levels. The connectors between classes should be arrows, with the head of the arrow pointing towards the superclass.

Step 2

In the space below, write the class header for each class in the hierarchy you sketched above. If you created additional subclasses, you can write those headers in the margin below. If you don’t have 4 subclasses, leave the extra spaces blank.

Superclass header:

Subclass header:

Subclass header:

Subclass header:

Subclass header:

Step 3

In the space below, write declarations for at least 3 PRIVATE fields for your superclass, a constructor, and a method that would make sense for every child class of your superclass.

Step 4

In the space below, write declarations for 1 - 2 PRIVATE fields, a constructor, and a method for each of your subclasses.

Code for subclass \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

(write the name of your subclass here)

Code for subclass \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

(write the name of your subclass here)

Code for subclass \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

(write the name of your subclass here)

Code for subclass \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

(write the name of your subclass here)

Step 5

Take a few minutes to check all of your code with your partner. Don’t forget to use your notes, 4 Commandments of Scope, error-checking guide, textbook, and classroom posters to help you check for mistakes or logic errors.

Step 6

Now you’re ready to create your poster! In the space below, sketch out where you will place the different parts of your poster. Your poster should illustrate the hierarchal relationship between your different classes, and show how inheritance allows programmers to reuse code. Check each item off the checklist as you incorporate it into your sketch:

* The names of the super- and subclasses.
* Complete code for each class (written on the lined paper), including the class header, object fields, constructor, and methods.
* The relationships between classes, drawn as arrows with the head of the arrow pointing to the superclass.
* Examples of instance objects, cut out from magazines/flyers (or drawn) and placed near/around the appropriate class.
* Client code (written on lined paper).

Step 7

Create the components of your poster, and put it together:

1. On the lined paper provided, write out your final version of code for each class, and for your client code.
2. Using the magazines, newspapers, and flyers, cut out several examples (instances) of objects in each class. If you can’t find instances of each object, you can draw them.
3. Using your sketch from Step 6, arrange all the parts of your poster and glue them down using the glue provided by your instructor.
4. Using the markers provided, finish off your poster by drawing in the hierarchical relationships, labeling each class (big so it can be easily read!), and adding any extra information you think will help your classmates understand inheritance.
5. Draw a star next to your superclass.

Step 8

Have your instructor come by to check your work. Once you’ve received approval, hang up your work where your instructor tells you to, and begin Step 9.

Step 9

Once you have finished, walk around the room to others’ examples, and take a look at their code. In your notebook, take notes on (1) what you liked about others’ code, (2) what you had questions about, (3) what you thought was well explained, and (4) what you would change, or what you change or clarify.