Step Into Java:
More on Inheritance

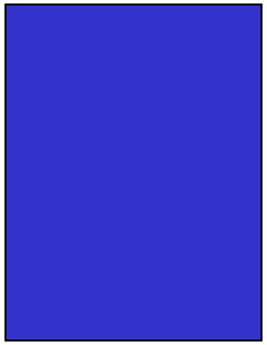
Mr. Neat

Java

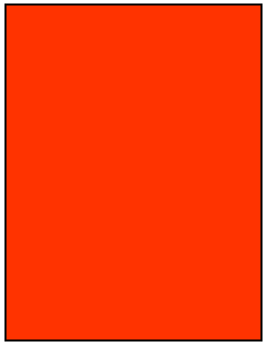
How did you solve the problem of putting a orbit around your ______?

Shift gears to SpongeBob Example…

Spongebob.java



Spongebobhat.java



```
Spongebob joe;
joe = new Spongebob(30,20);
```

```
Spongebobhat joe;
joe = new Spongebobhat(30,50);
```

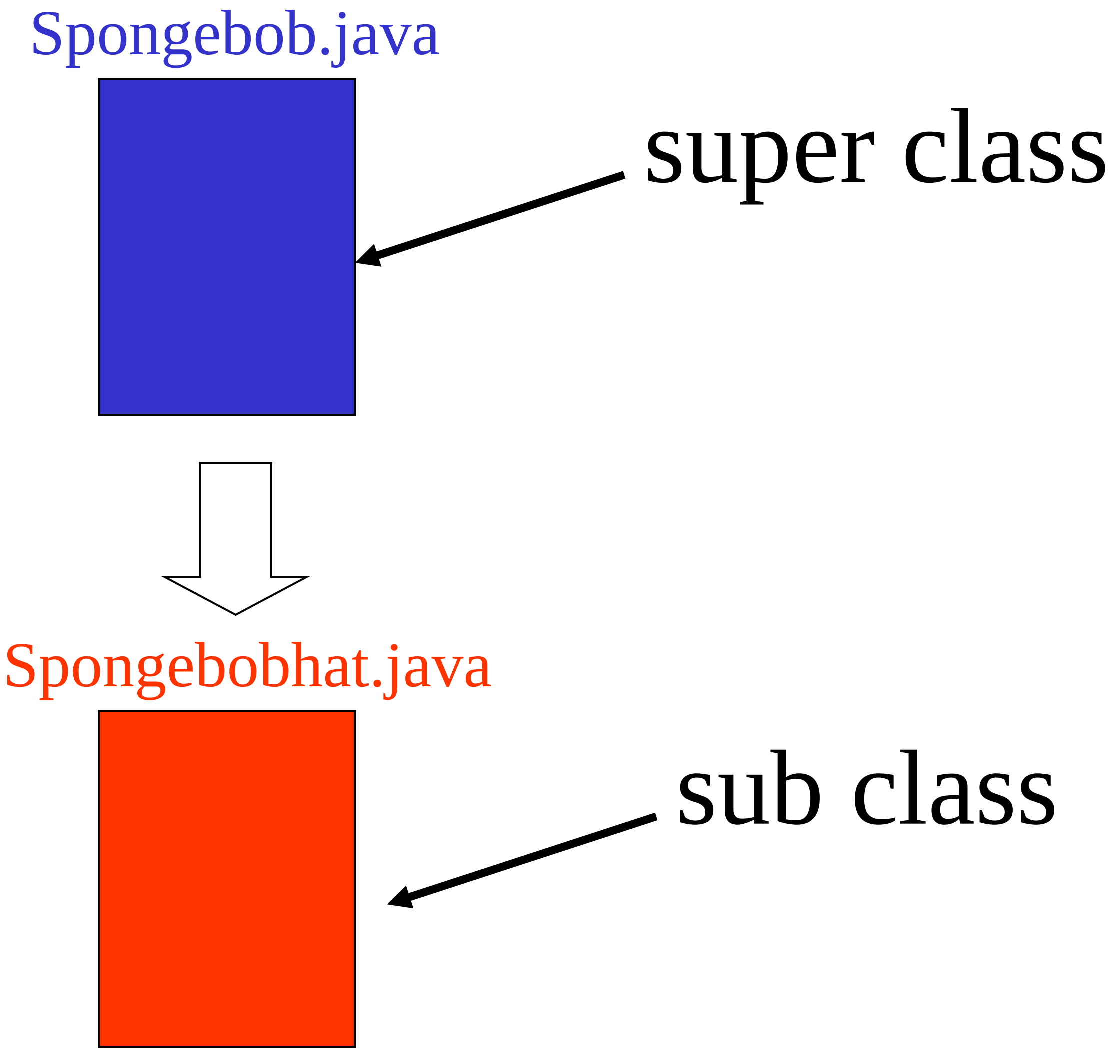
What if you want to change the constructor in the Spongebob class?

New Concept….extend an existing class

Allows you to create a specialized class from a more general class.

Let’s extend the Spongebob class to a class where Spongebob objects are wearing a hat.

Spongebobhat extends Spongebob

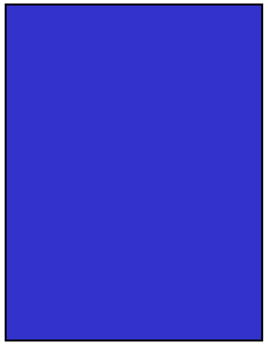


Spongebobhat inherits Spongebob methods and instance variables

Any public Spongebob class method can be called with a Spongebobhat object.

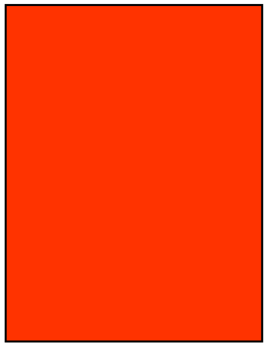
For Example

Spongebob.java



```
Spongebob joe;
joe = new Spongebob(30,20);
```

Spongebobhat.java



```
Spongebobhat joe;
joe = new Spongebobhat(30,20);
joe.move(5,5);
```

How Do You Make a Constructor for the Extended Class?

Key word: *super*

-*super* calls the super classes' constructor

- must be first line of sub class constructor

Spongebob.java

Spongebobhat.java

```
public Spongebob(int x, int y)
{
        //making sb
}
```

calls this constructor

```
public Spongebobhat(int x, int y)
{
        super(x,y);
        // stuff unique to sbh
}
```

Spongebob.java

Spongebobhat.java

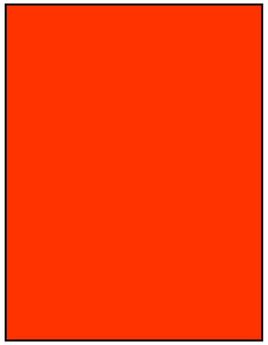
```
public Spongebob(int x, int y)
{
        //making sb
}
```

calls this constructor

```
public Spongebobhat(int x, int y)
{
        super(x,y);
        // stuff unique to sbh
}
```

What is good about this?

Spongebobhat.java



```
public class Spongebobhat extends Spongebob
{
        public Spongebobhat(int x, int y)
        {
                super(x,y);
                // stuff unique to sbh
        }
        ……
}
```

Remember Karel?

```
class farmerbot:ur_Robot
{
    harvestOneRow();
    harvestField();
    ….
};
```

How did you know how big
to make the hat on sb?

What were the parameters of
your *superclass* sb constructor?

How are we going to figure out how wide to make the hat?

Where is that information?

Another new reserved word….
protected
variables and methods

So far know about public
and private,…now *protected*!

The protected qualifier allows the instance variable or method of a superclass to be accessed by a subclass.

Word of caution….stay away from protected variables!

(Just like you stay away from public variables)

Better approach…

add accessor
methods to the superclass
which allows the information
to be accessed by the
subclass.

New Concept…

Add methods to the superclass in anticipation of it being extended.

How do you make the new sbh objects move while keeping his hat on?

What happens if you make the call:

```
// define joe to be sbh object

joe.move(5,5);
```

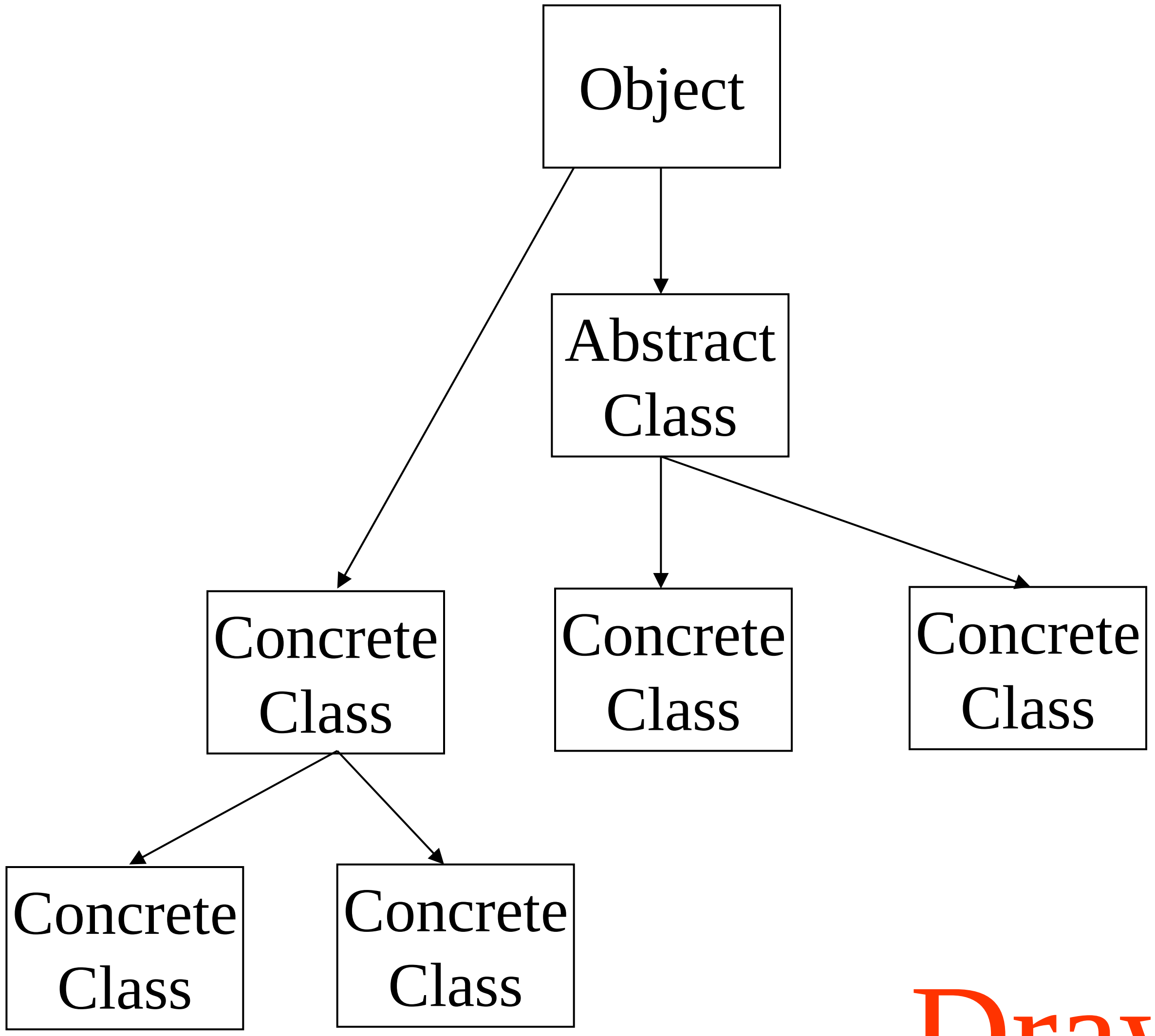
How do you move each part of sb inside of sbh?

Let’s Override the move method….

```
public move(double x, double y)
{
    super.move(x,y);
    // then move the new part
}
```

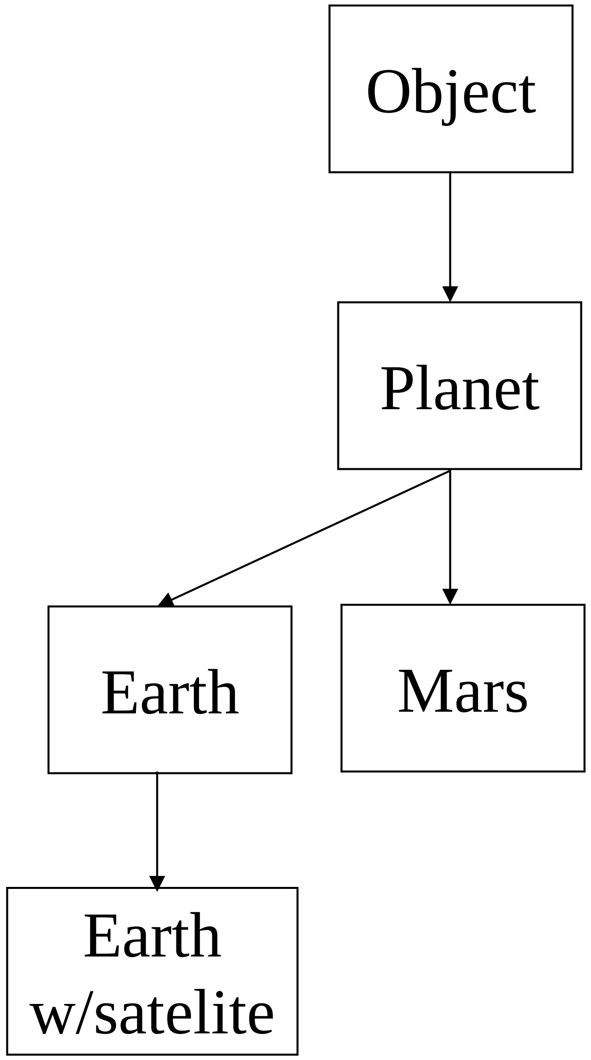
calls the move method of the superclass

Inheritance Family Tree



Draw these!!

Next Lab ….



abstract class

concrete classes

Next Lab

-Add an orbit to your Earth

-Could be a natural (moon) or manmade satelite.

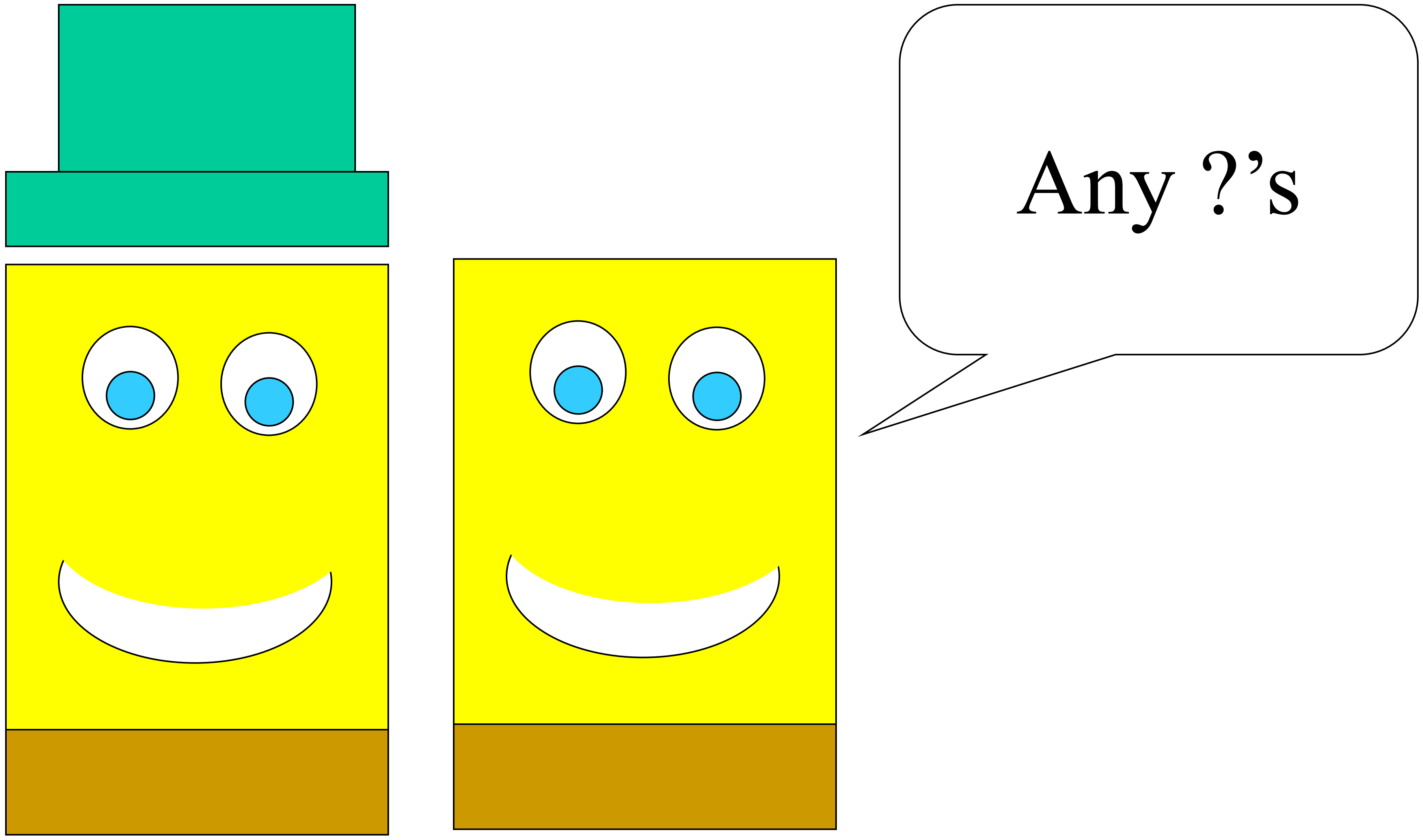
-Add necessary accessor methods if necessary.

- make a constructor and verify it works

- then try to drag it

Depends on selected topic

SpongeBob with Hat
Earth with Satelite
BlackKnight with Sword


Any ?'s