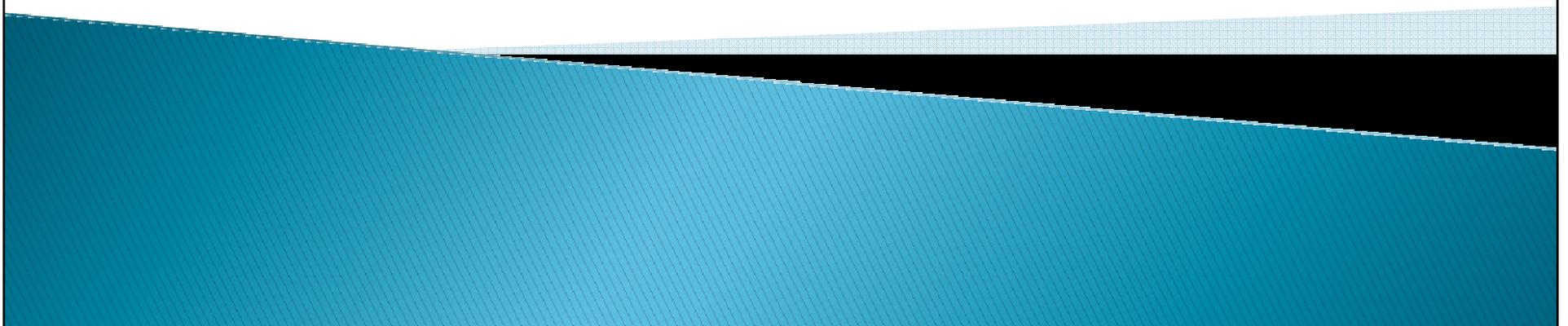


CSSE 220 Day 5

Inheritance



Announcements

- ▶ Demo making sure all files are committed.
- ▶ Questions?

Recap: "Resize" an array

- ▶ An array is inherently fixed-length.
- ▶ But we can get the effect of a "growable array":
 - Have two variables, `arr`, and `size`.
 - initialize `arr` to be an array of 5 elements
 - I choose 5 because that is what Mark Weiss does.
 - When we want to add a new element at the end:
 - if `size == arr.length`
 - call `resize` to give us an array twice as big.
 - Put the new element in `arr[size]` and increment `size`.
 - Code:

```
if (size == arr.length)
    arr = resize(arr, size, size*2);
arr[size++] = newValue;
```

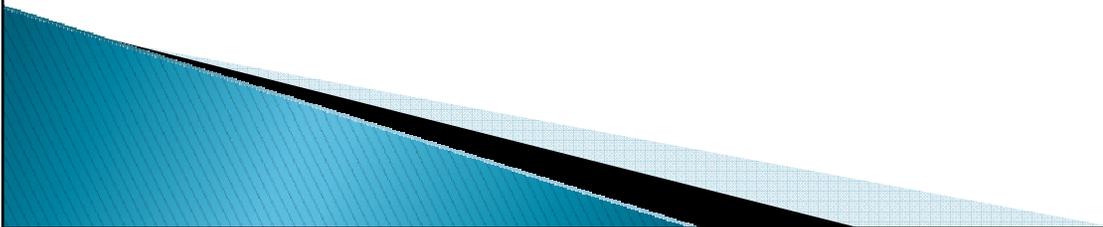
Write
`resize()`

Why `*2` instead of `+1`?

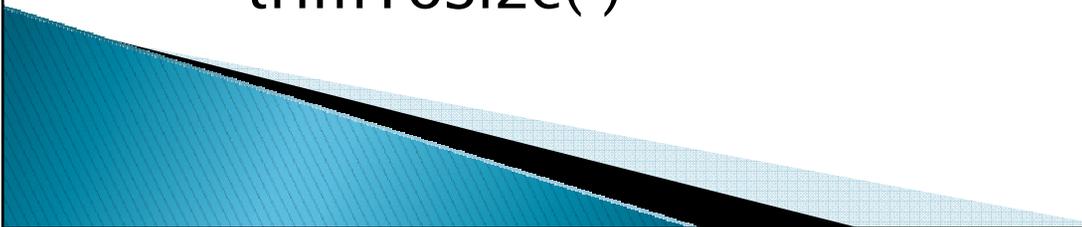
You'll answer that question mathematically on the first day of 230 (if not sooner)

resize Solution

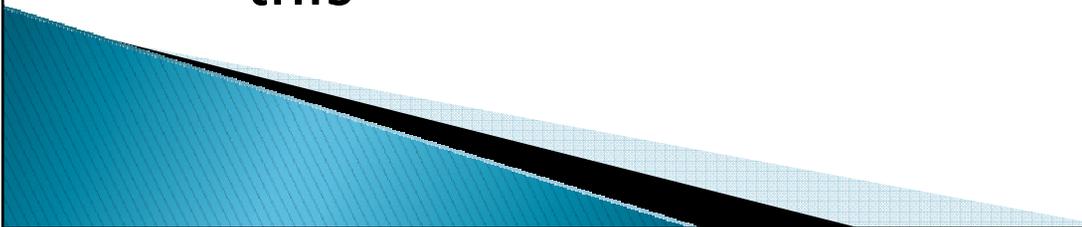
```
▶ int[] resize(int[] a, int oldsize, int newsize) {  
    int[] result = new int[newSize];  
    int numToCopy = Math.min(oldsize, newsize);  
  
    for (int i=0; i < numToCopy; i++) {  
        result[i] = a[i];  
    }  
  
    return result;  
}
```



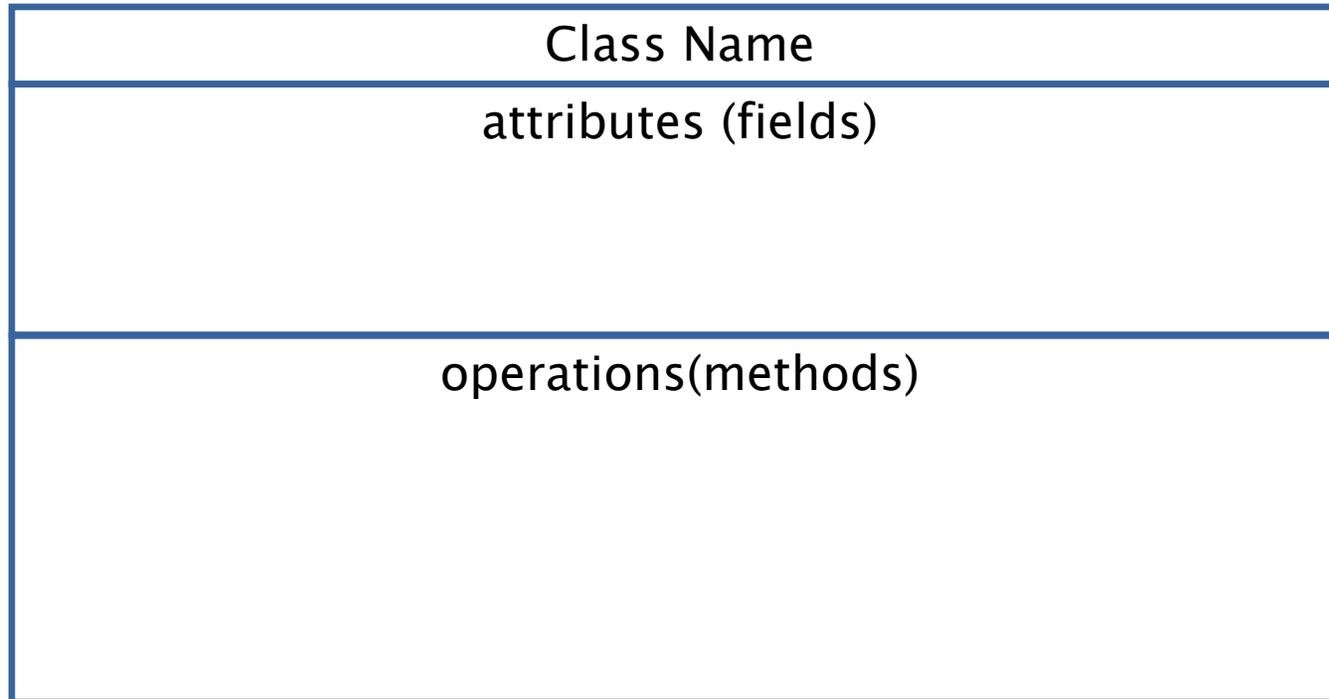
ArrayList: a class that implements a resizable array-like structure

- ▶ Full name: `java.util.ArrayList`
 - ▶ Methods include
 - `add(element)`
 - `add(index, element)`
 - `get(index)`
 - `size()`
 - `clear()`
 - `remove(object)`
 - `remove(index)`
 - `set(index, element)`
 - `toArray()`
 - `trimToSize()`
- 

Yesterday: Objects and Classes

- ▶ Hopefully after doing this exercise, you know experientially the meanings of these terms:
 - object
 - class
 - instance
 - field
 - method
 - constructor
 - private (information hiding)
 - encapsulation
 - this
- 

Simple UML Class Diagram



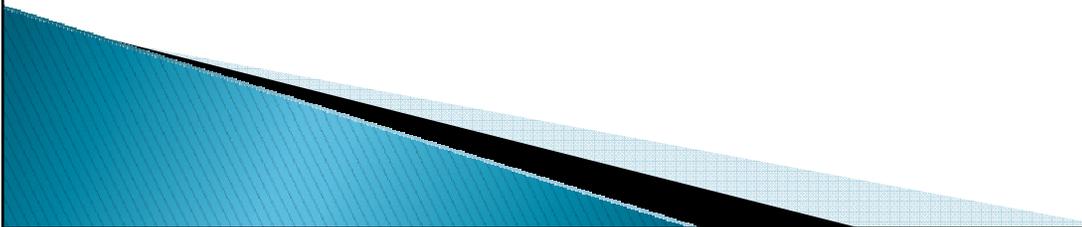
Draw part of the Rectangle diagram

Inheritance

- ▶ Inheritance allows you to **reuse** methods that you've already written to create more specialized versions of a class. The Java word for "inherits from" is **extends**.
- ▶ An extension class (subclass) has all of the fields and methods of the extended class (superclass), plus
 - perhaps some new fields
 - and almost always some new or overridden methods.
 - A term that almost always applies to inheritance is "IS-A".
 - Example: A square IS-A Rectangle
Square **extends** Rectangle

Other natural examples

- ▶ A Sophomore IS-A Student IS-A Person.
 - ▶ A Continent IS-A LandMass
 - ▶ An HPCompaqNW8440 IS-A Laptop Computer
 - ▶ An iPod IS-A MP3Player
 - ▶ A Square IS-A Rectangle

 - ▶ It is **not** true that a Continent IS-A Country or vice-versa.
 - ▶ Instead, we say that a Continent HAS-A Country.
- 

Examples From the Java API Classes

- | | | |
|------------------|---------|--------------------|
| ▶ String | extends | Object |
| ▶ ArrayList | extends | AbstractCollection |
| ▶ IOException | extends | Exception |
| ▶ BigInteger | extends | Number |
| ▶ BufferedReader | extends | Reader |
| ▶ JButton | extends | Component |
| ▶ MouseListener | extends | EventListener |
| ▶ Frame | extends | Window |

Extend the Rectangle Class

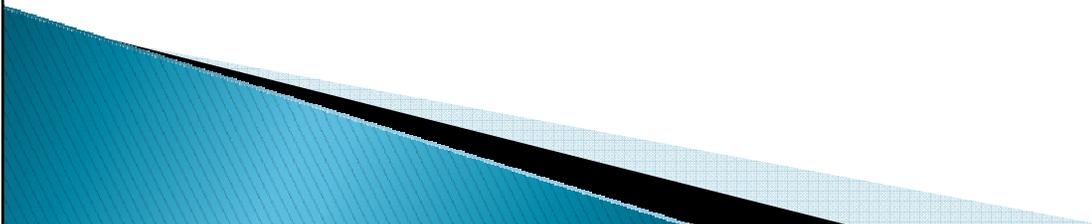
- ▶ Write Square
- ▶ Do we need new Instance variables?
- ▶ What methods can/should we override?
- ▶ Do it.
- ▶ Is this code legal:
 - `Rectangle r = new Square(...);`
 - `Square s = new Rectangle(...);`
- ▶ Note: r can do everything rectangles can do; it actually has more, but that's OK.
- ▶ What if s is asked to do something only squares can do? It's really just a rectangle!

Version Control Systems (review)

- ▶ Store "snapshots" of all the changes to a project over time
- ▶ Benefits:
 - Allow multiple users to share work on a project
 - Act as a "global undo"
 - Record who made what changes to a project
 - Maintain a log of the changes made
 - Can simplify debugging
 - Allow engineers to maintain multiple different versions of a project simultaneously

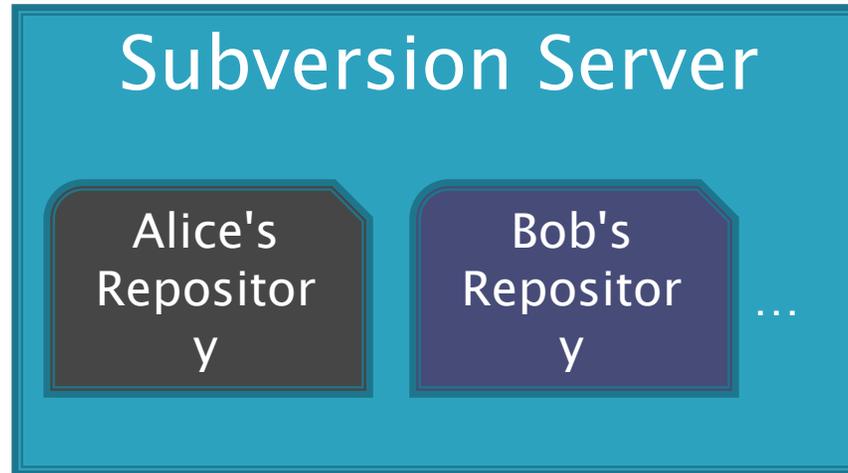
Our Version Control System

- ▶ Subversion, sometimes called SVN
- ▶ A free, open-source application
- ▶ Lots of tool support available
 - Works on all major computing platforms
 - TortoiseSVN for version control in Windows Explorer
 - Subclipse for version control inside Eclipse

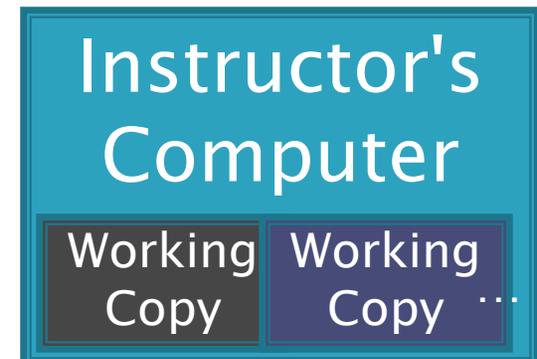
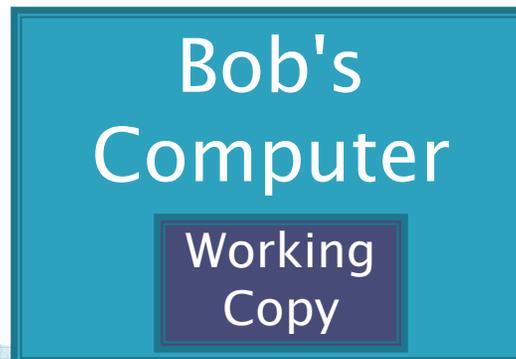


Version Control Terms

Repository: the copy of your data on the server, includes *all* past versions

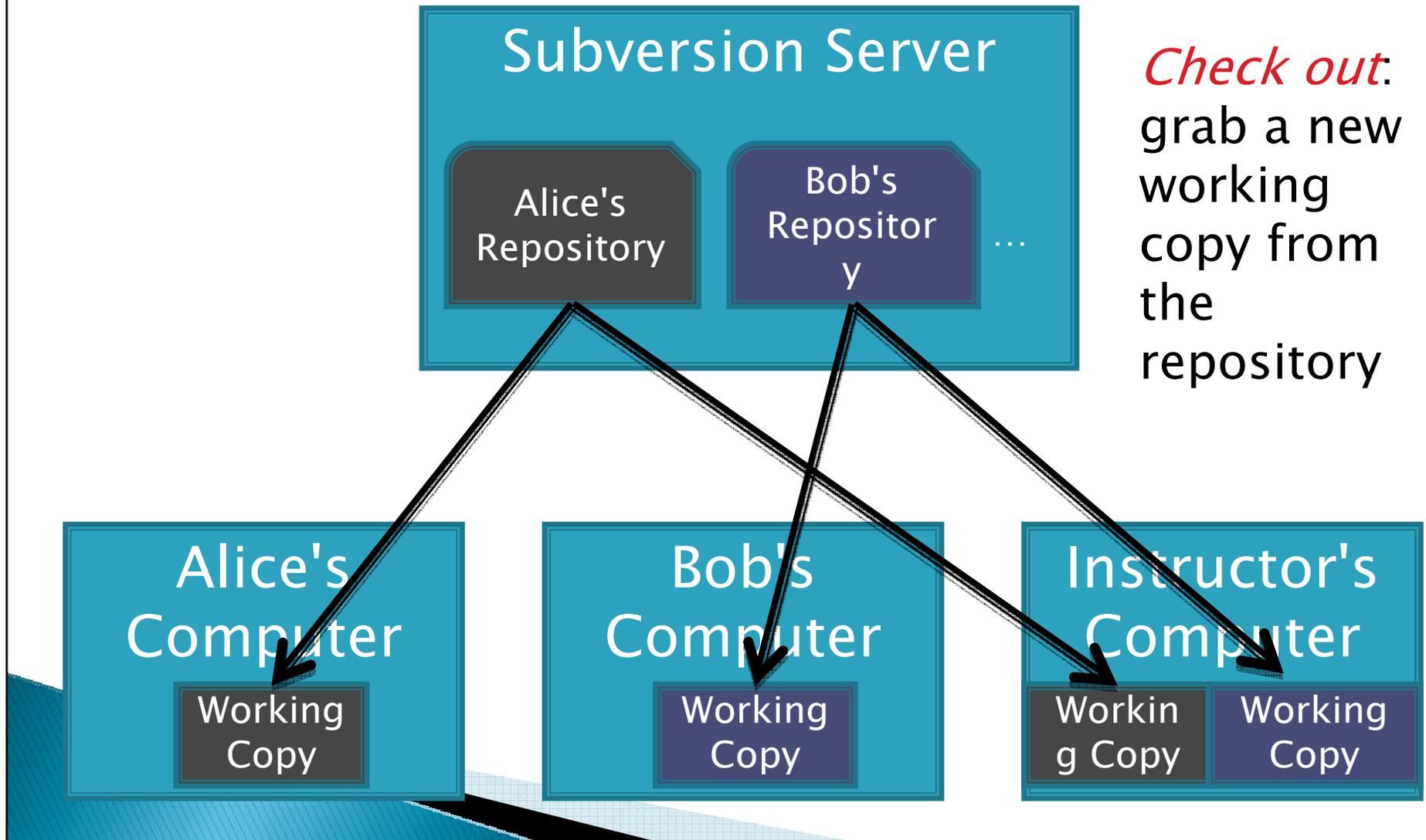


Working copy: the *current* version of your data on your computer

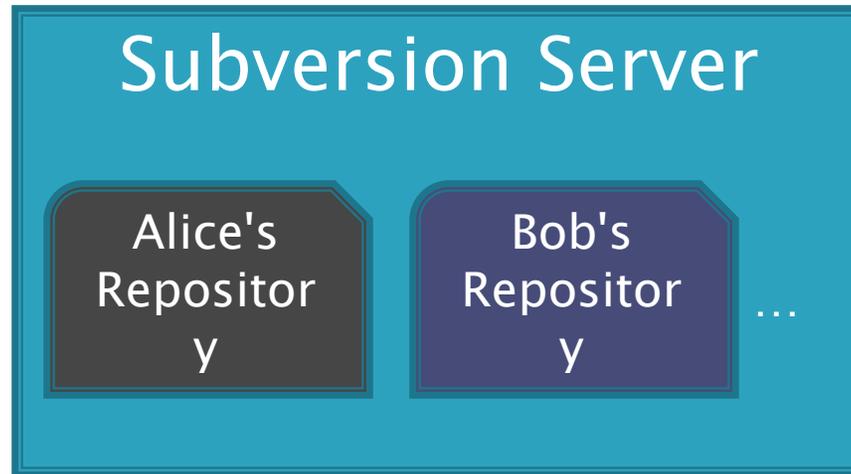


Version Control Steps—Check Out

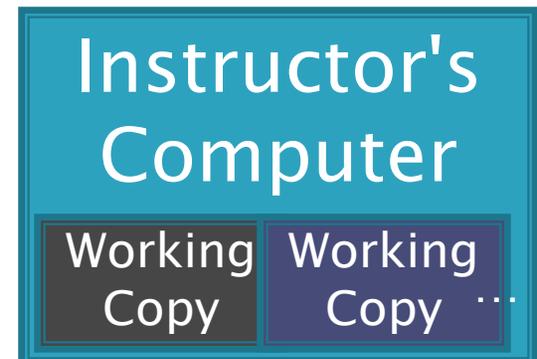
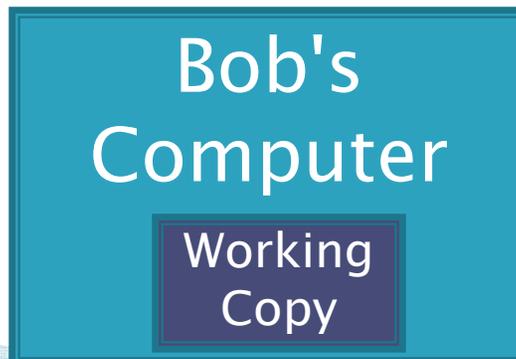
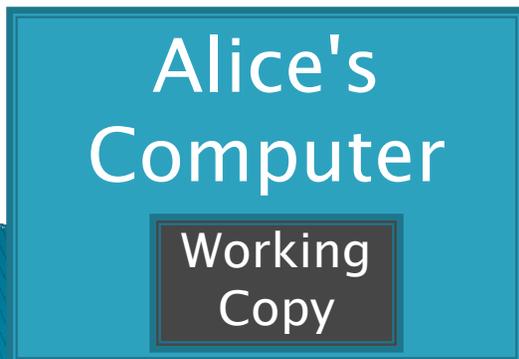
Check out:
grab a new
working
copy from
the
repository



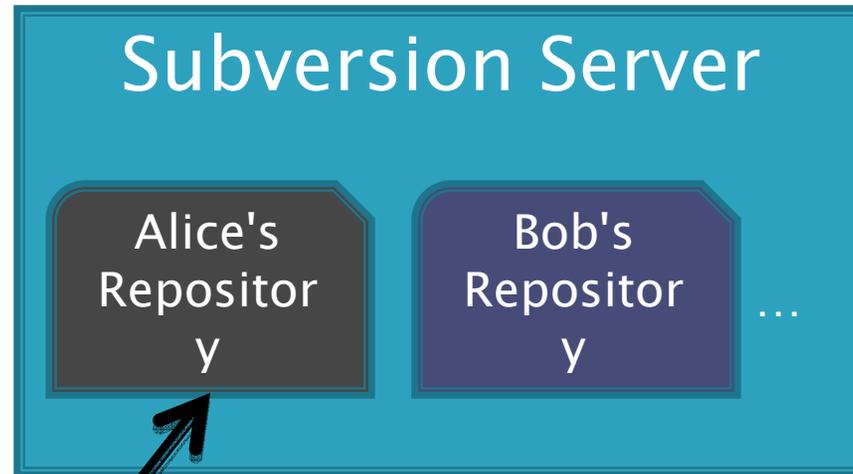
Version Control Steps—Edit



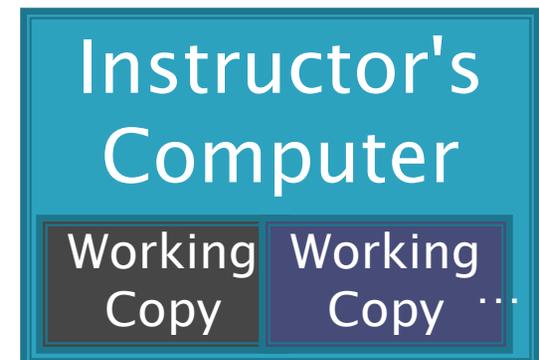
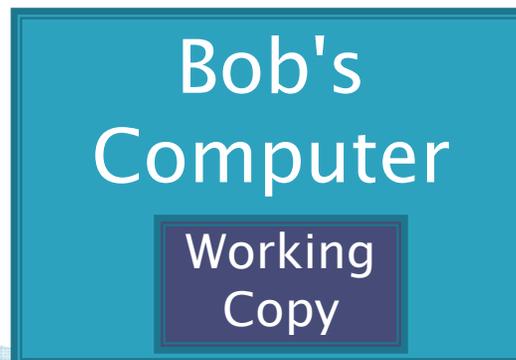
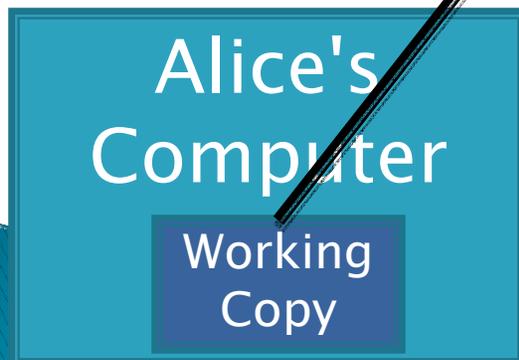
Edit: make *independent* changes to a working copy



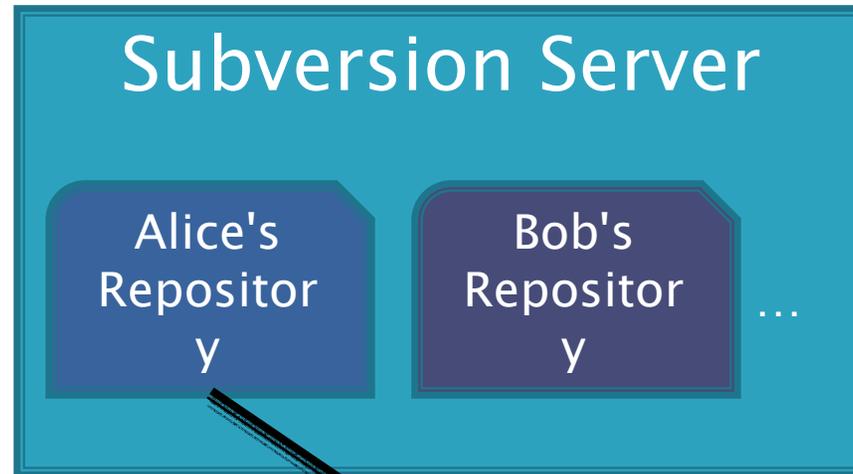
Version Control Steps—Commit



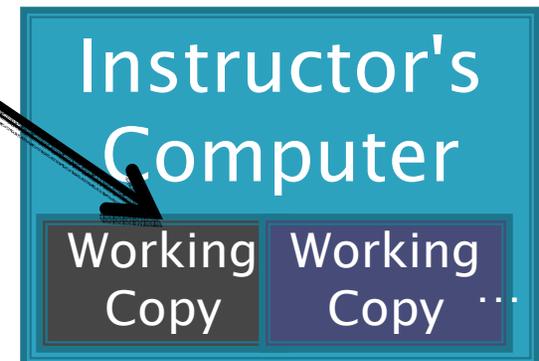
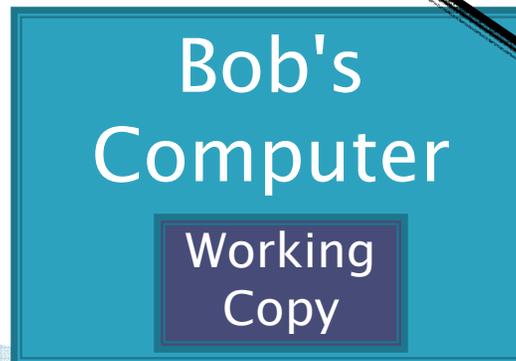
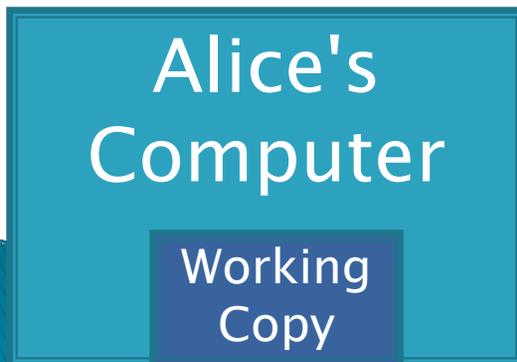
Commit:
send a
snapshot
of changes
to the
repository



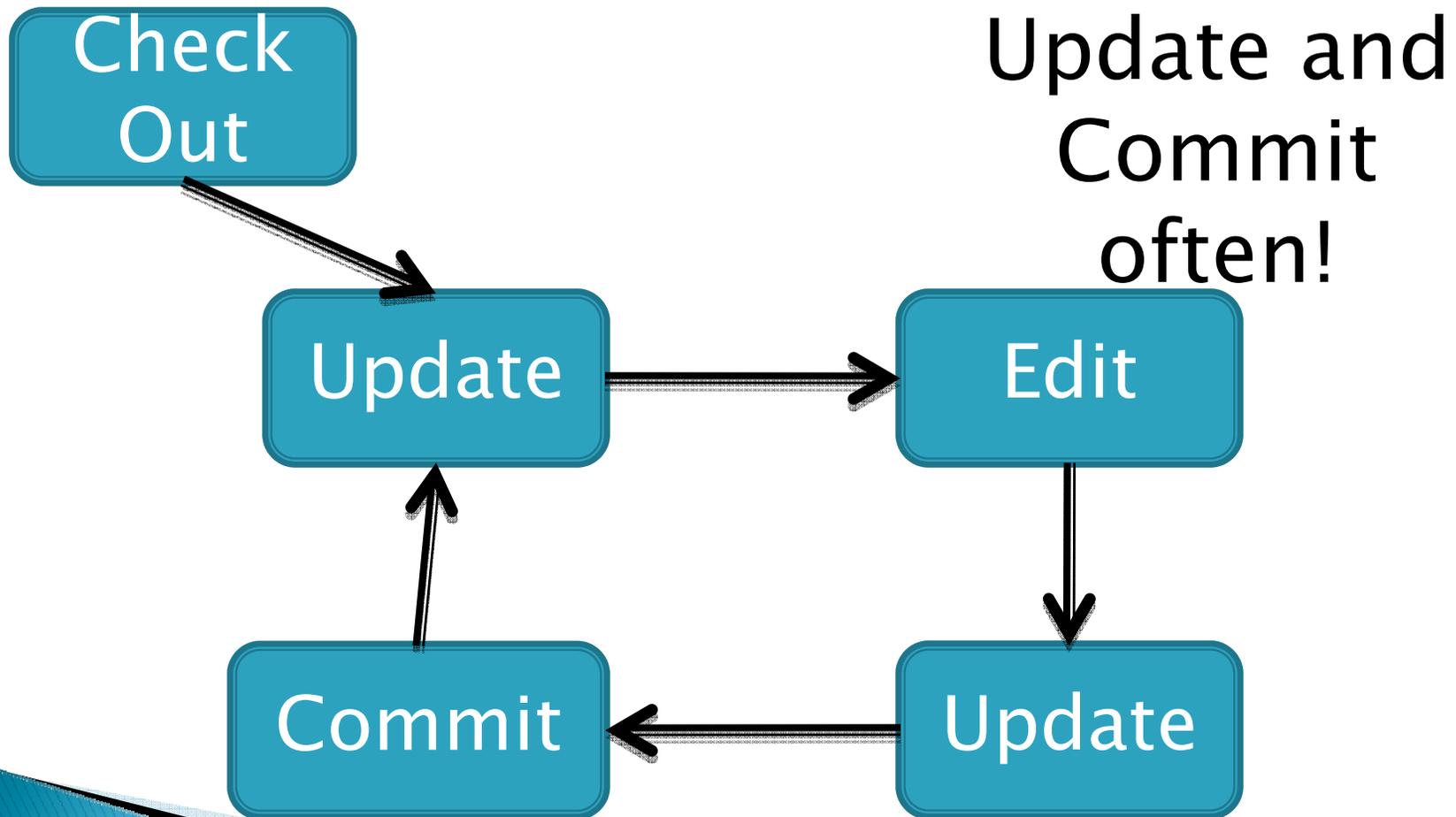
Version Control Steps—Update



Update:
make
working
copy
reflect
changes
from
repository



The Version Control Cycle



Subversion in Eclipse—Subclipse

SVN Repository Exploring perspective

The screenshot shows the Eclipse IDE in the SVN Repository Exploring perspective. The window title is "SVN Repository Exploring - swap.py 405 - Eclipse SDK". The menu bar includes File, Edit, Source, Refactoring, Navigate, Search, Project, Run, Window, and Help. The toolbar contains icons for SVN Repository, Debug, and Pydev. The left sidebar shows a tree view of the repository structure, including branches, tags, trunk, and various project folders like Administration, CatapultMaterials, C Materials, EclipseProjects, InstructorCode, and src. The main editor displays the contents of a file named "swap.py 405", showing Python code for a swap function. The bottom view shows the history of the file, with a table of revisions.

View showing repository directories and files

Tiny button to link to new repository

View showing contents of a file in the repository

View showing history of a file

Tags	Date	Author	Comment
*405	9/22/07 2:34 ...	clifton	Moved all Python code in Instru...
326	9/18/07 12:06 PM	defoe	Solution to file Averages problem fro...

Getting the code for BigRational

- ▶ ... from your SVN Repository
 - Go to HW5
Then click the BigRational link
 - Let's start it together
 - I moved written problems to HW6 to free up room for you do get a solid start on this tonight, so you can ask questions tomorrow.