

# SUBVERSION, NESTED LOOPS, BOOLEAN VALUES

CSSE 120—Rose Hulman Institute of Technology

# Software Engineering Tools



- The computer is a powerful tool
- So use it to make software development easier and less error prone!
- Some software engineering tools:
  - ▣ IDEs, like Eclipse
  - ▣ Version Control Systems—like Subversion
  - ▣ Diagramming applications—like Violet or Visio
  - ▣ Modeling languages—like Alloy, Z, or JML

# Version Control Systems



- 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

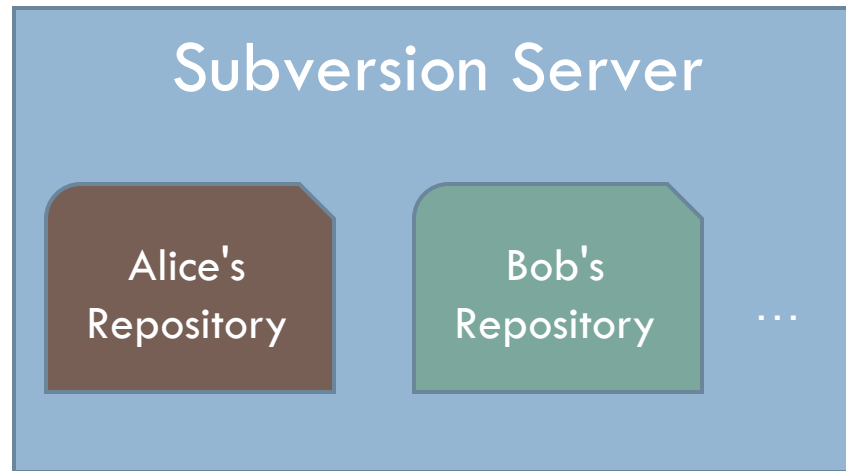
# Our Version Control System



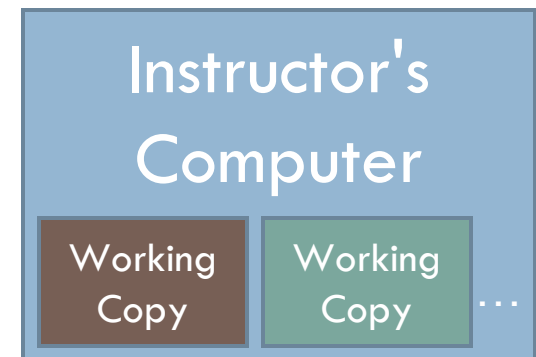
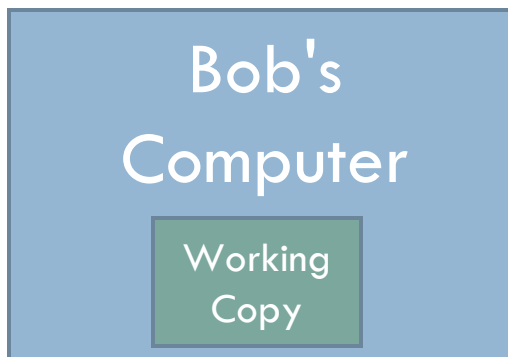
- Subversion, sometimes called SVN
- A free, open-source application
- Lots of tool support available
  - ▣ Works on all major computing platforms
  - ▣ TortoiseSVN for version control in Windows Explorer
  - ▣ Subclipse for version control inside Eclipse

# Version Control Terms

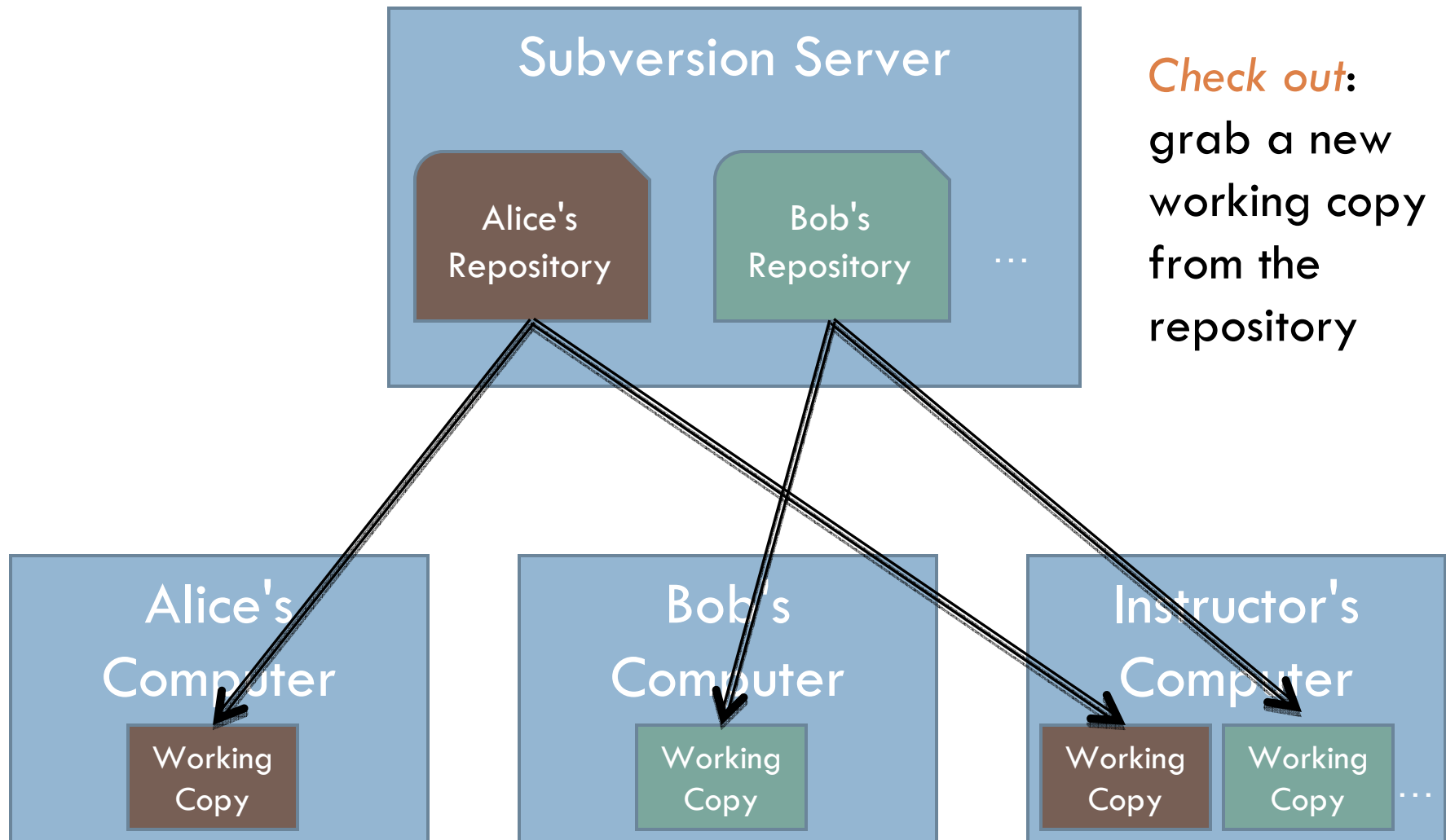
*Repository*: the copy of your data on the server, includes **all** past versions



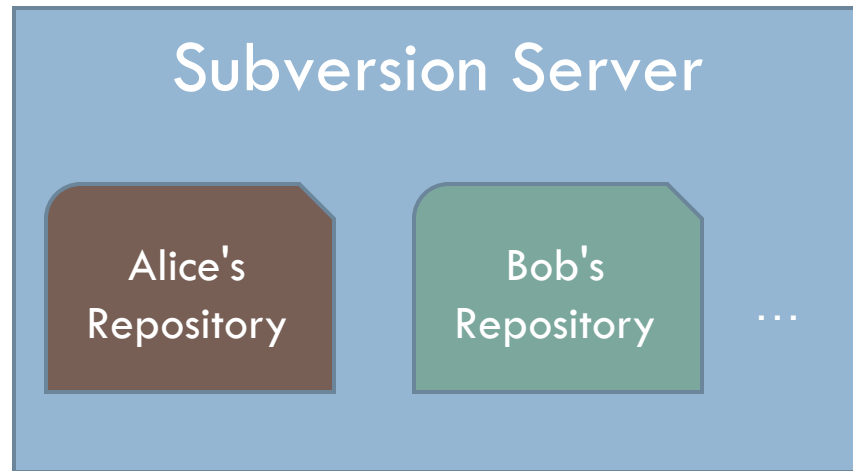
*Working copy*: the **current** version of your data on your computer



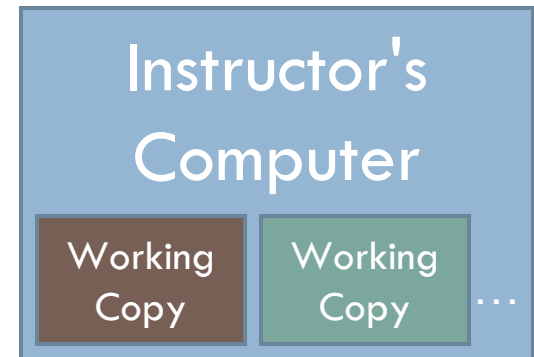
# Version Control Steps—Check Out



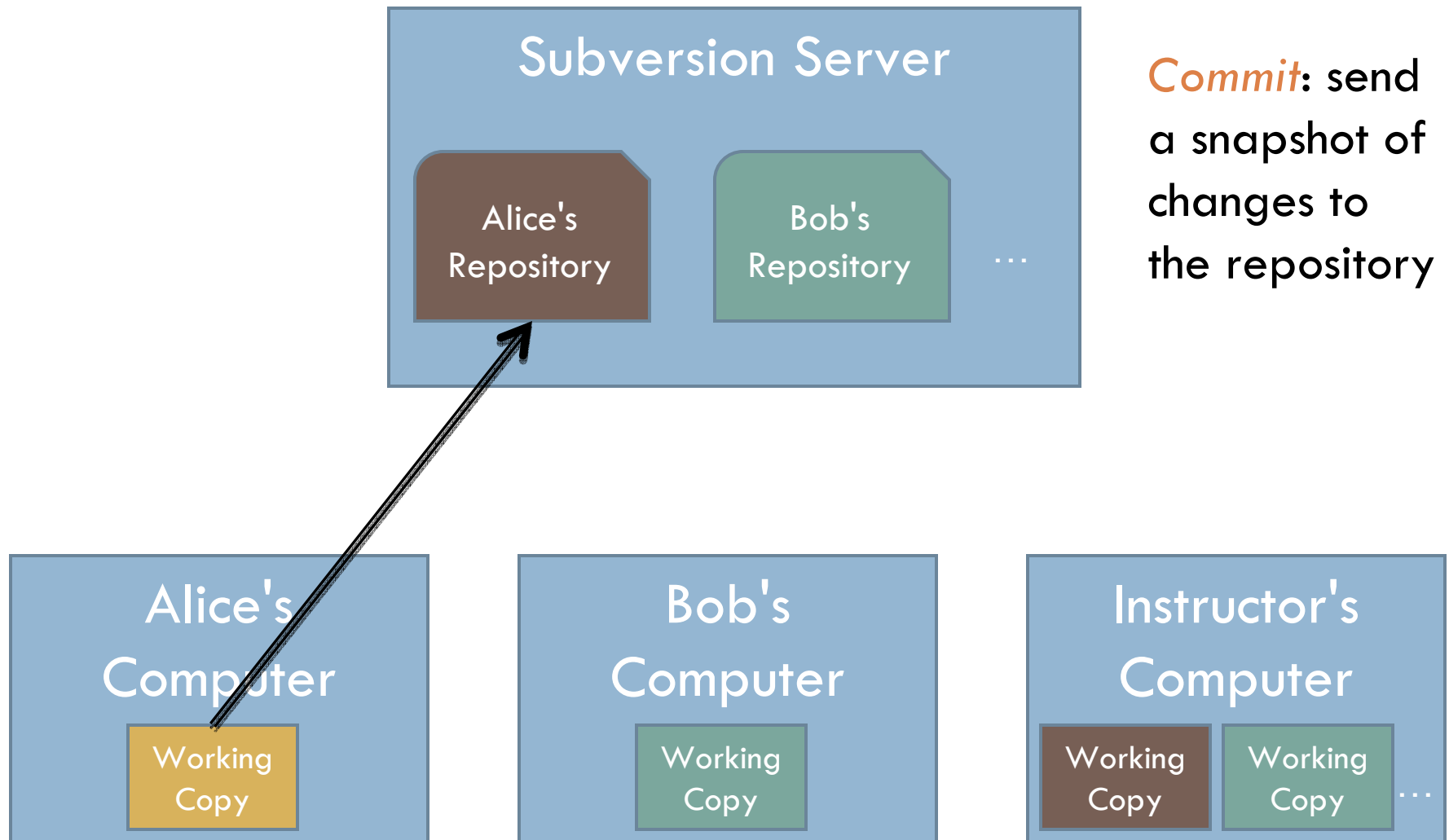
# Version Control Steps—Edit



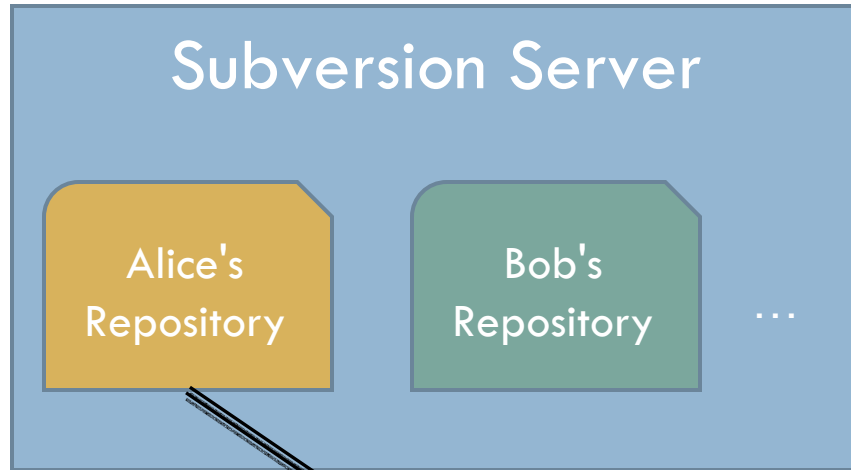
*Edit:* make **independent** changes to a working copy



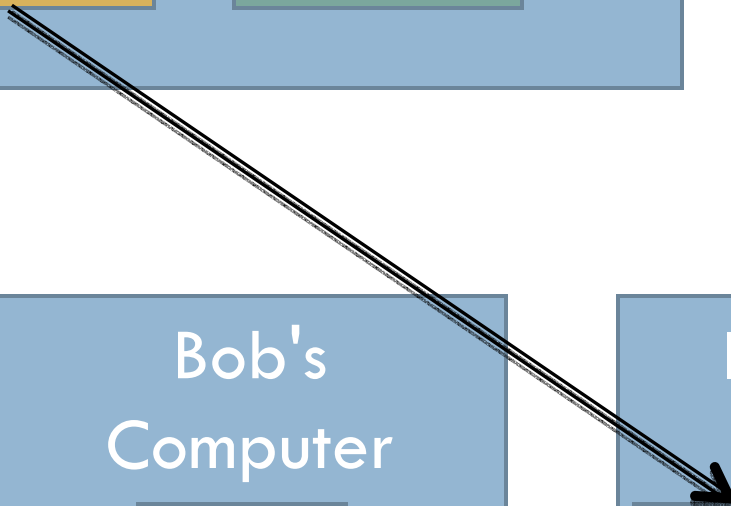
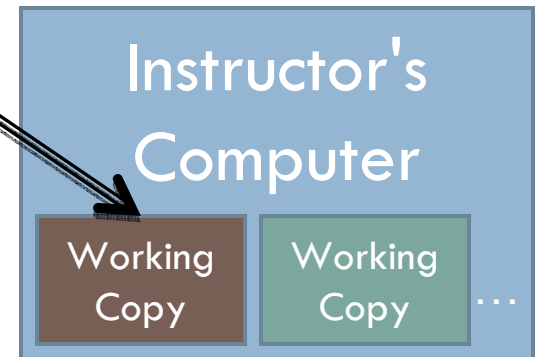
# Version Control Steps—Commit



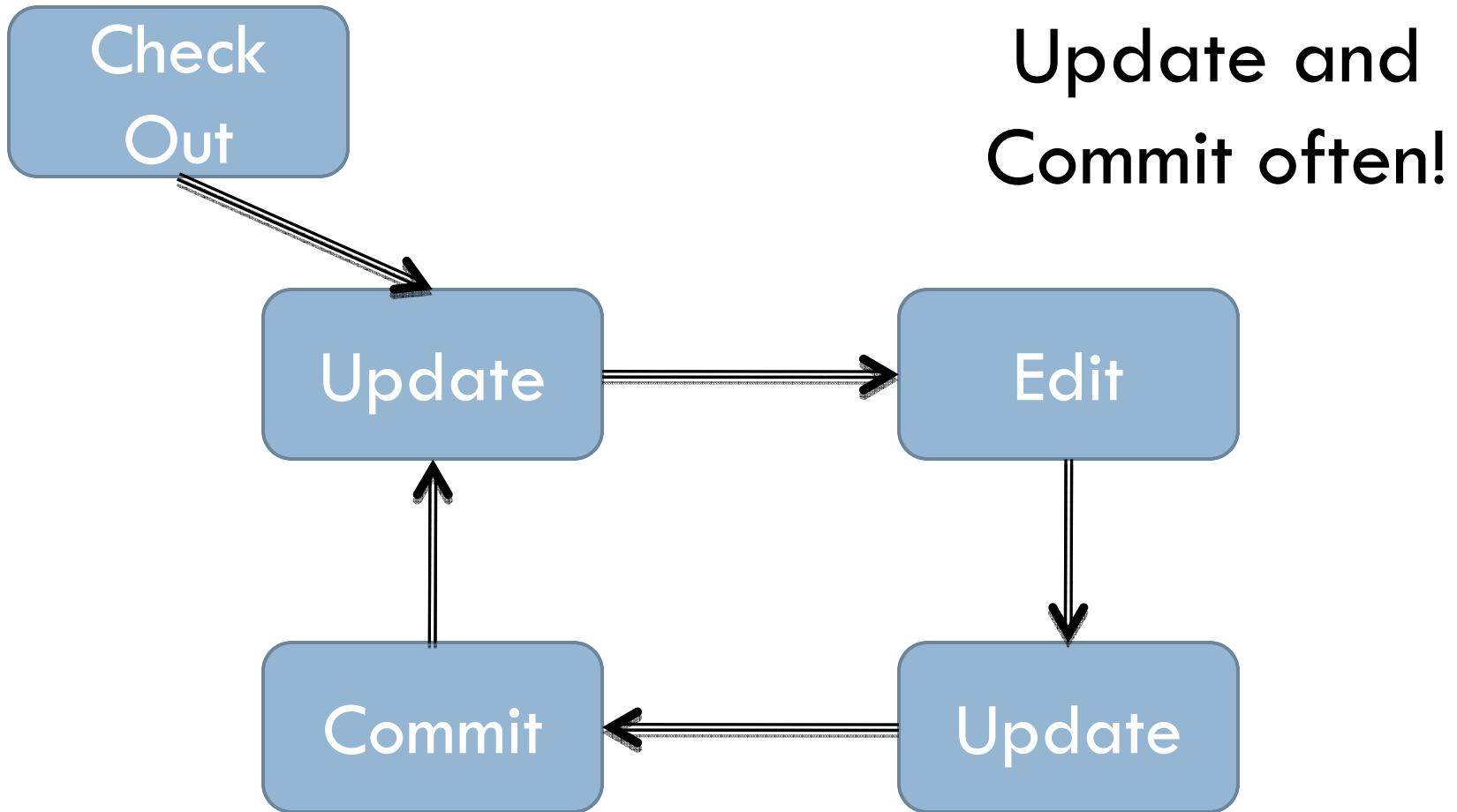
# Version Control Steps—Update



*Update:* make working copy reflect changes from repository



# The Version Control Cycle



# Subversion in Eclipse—Subclipse

## SVN Repository Exploring perspective

The screenshot displays the Eclipse IDE in the SVN Repository Exploring perspective. The main window is titled "SVN Repository Exploring - swap.py 405 - Eclipse SDK". The interface is divided into several panes:

- Left Pane (Repository Explorer):** Shows a tree view of the SVN repository structure. The root is `http://svn.cs.rose-hulman.edu/repos/csse120-python`, with subdirectories for `branches`, `tags`, and `trunk`. Under `trunk`, there are folders for `Administration`, `CatapultMaterials`, `C Materials`, `EclipseProjects`, and `InstructorCode`. The `src` folder under `InstructorCode` contains several Python files, including `circleOfCircles.py 405`, `ClickInsideCircle.py 405`, `ClickInsideCircleSimple.py 405`, `decode.py 405`, and `nadiasSavings.py 405`.
- Center Pane (Code Editor):** Displays the contents of the file `swap.py 405`. The code includes a comment `# attempt to exchange two integers` and a function definition `def swapInts(x, y):` with the body `x, y = y, x`. Below this, there is a call to `listElements` and a print statement: `print 'Before "swapInts": x=%d, y=%d' % (x, y)`. The code is color-coded by syntax.
- Bottom Pane (History View):** Shows the history of the file. The title is `History x`. The path is `/trunk/EclipseProjects/InstructorCode/src/fileAveragesSolution.py` in `http://svn.cs.rose-hulman.edu/repos/csse120`. The table below shows the history entries:

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...


Tiny button to link to new repository

View showing contents of a file in the repository

View showing repository directories and files

View showing history of a file

# Individual Exercise, Part 1

- Install (or verify) Subclipse on your laptop
  - ANGEL Course → Resources → Course Resources → Software Installation → Installing Subclipse
- Open the SVN Repository Exploring perspective
  - Window → Open Perspective → Other...
- Add a new repository:
  - Click the tiny icon: 
  - URL: <http://svn.cs.rose-hulman.edu/repos/csse120-200820-username> replacing *username* with your actual username
  - **When prompted use the password that we emailed!**

Get help if you're stuck!  
Go on to Part 2 if you're done.

# Individual Exercise, Part 2

---

- ❑ Browse the SVN Repository view for the `FirstSVNProject` project
- ❑ Right-click it, and choose `Check Out`.
- ❑ Confirm all of the options presented
- ❑ Switch to `PyDev` perspective
- ❑ In Package Explorer, find `spam.py` inside your `FirstSVNProject` project
- ❑ Follow the instructions in the comments at the start of that file

# Boolean Variables and Operations

- Boolean constants: **True False**
- Relational operators (<, etc.) produce Boolean values.

```
>>> 4 < 5
True
>>> 6 != 6
False
```

- Other Boolean operators: **and or not**

<i>P</i>	<i>Q</i>	<i>P and Q</i>
T	T	T
T	F	F
F	T	F
F	F	F

<i>P</i>	<i>Q</i>	<i>P or Q</i>
T	T	T
T	F	T
F	T	T
F	F	F

<i>P</i>	not <i>P</i>
T	F
F	T

# Nested Loops

- A *nested if* is an **if** inside an **if**.
- A *nested loop* is a loop inside a loop.
- Example: 

```
for i in range(4):  
    for j in range(3):  
        print i, j, i*j
```
- What does it print?
- What if we change the second range expression to `range(i+1)`?

# Nested Loop Practice—Example

- ❑ Put this code inside `NestedLoopPatterns.py` in `FirstSVNProject`
- ❑ You will do several exercises that involve writing functions to generate patterned output.
- ❑ In each, you will accumulate each line's output in a string, then print it.
- ❑ First, a function to generate a pattern of asterisks like

```
*****
*****
*****
```
- ❑ To produce the above pattern, call `rectangleOfStars(3, 11)`

# Nested Loop Practice – Your Turn

□ Complete these definitions and test your functions

□ `triangleOfStars(n)` produces a triangular pattern of asterisks. For example, `triangleOfStars(6)` produces

```
*
**
***
****
*****
*****
```

**Hint:** Use the same idea as the previous example. Start each line with an empty string. As you go through your inner loop, accumulate the line's characters. Print the line before the next iteration of the outer loop.

□ `triangleOfSameNum(n)` produces a triangular pattern of numbers. For example, `triangleOfSameNum(5)` produces

```
1
22
333
4444
55555
```

If you finish with these in class, continue with the remaining homework problems.