SUBVERSION, OBJECTS, AND GRAPHICS

CSSE 120 – Rose-Hulman Institute of Technology

Software Engineering Tools

- The computer is a powerful tool
- We can 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

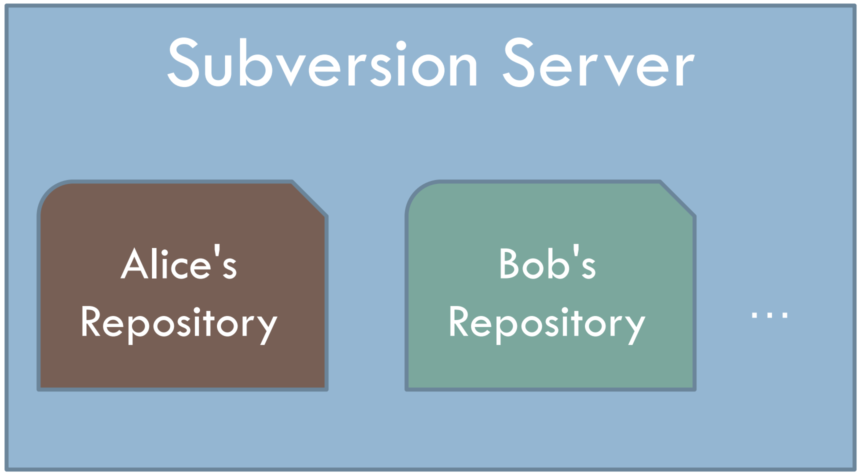
- Store "snapshots" of all the changes to a project over time
- Benefits:
  - Allow multiple users to share work on a project
  - Act as a "global undo"
  - Record who made what changes to a project
  - Maintain a log of the changes made
  - Can simplify debugging
  - Allow engineers to maintain multiple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

Our Version Control System

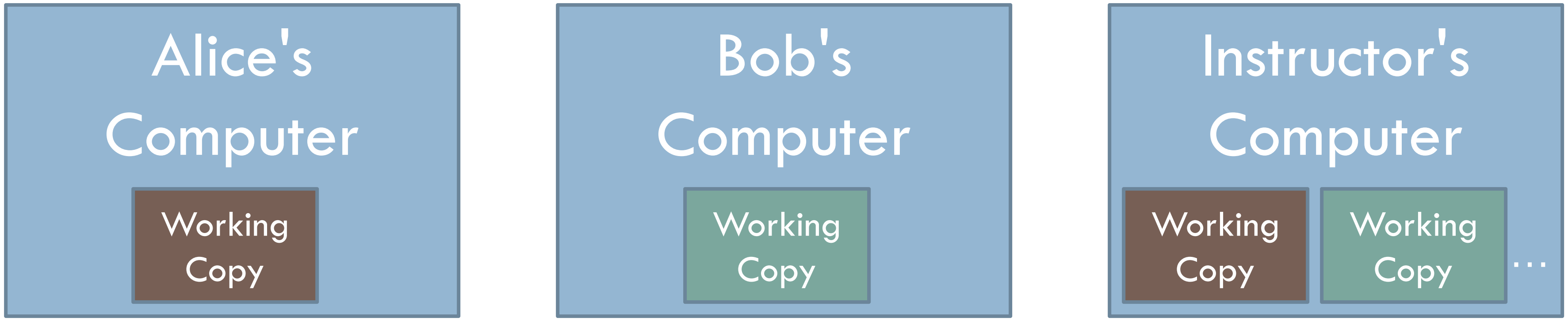
- Subversion, sometimes called SVN
- A free, open-source application
- Lots of tool support available
  - Works on all major computing platforms
  - **TortoiseSVN** for version control in Windows Explorer
  - **Subclipse** for version control inside Eclipse

Version Control Terms

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

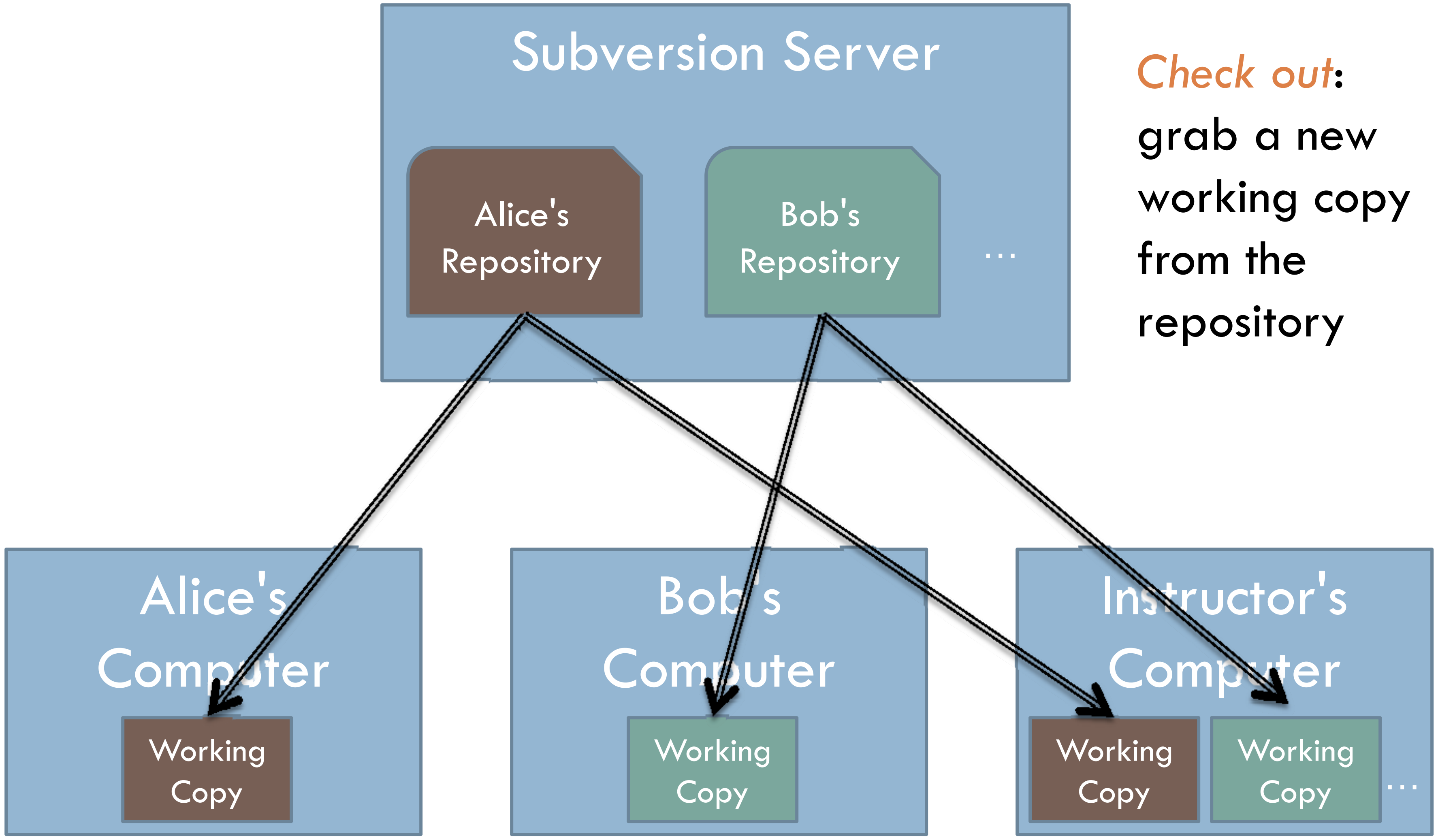


Working copy: the ***current*** version of your data on your computer



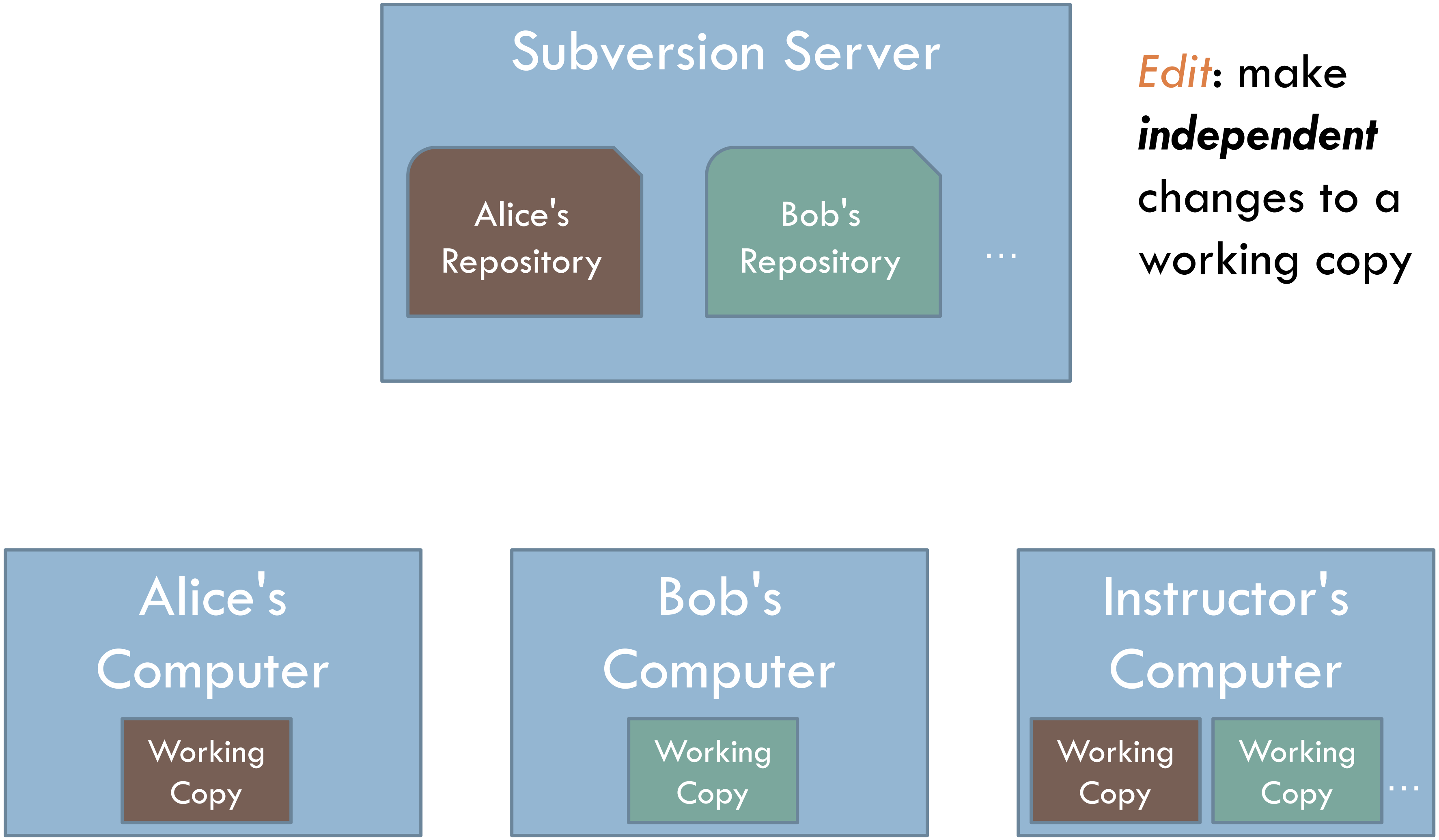
Q1b

Version Control Steps—Check Out



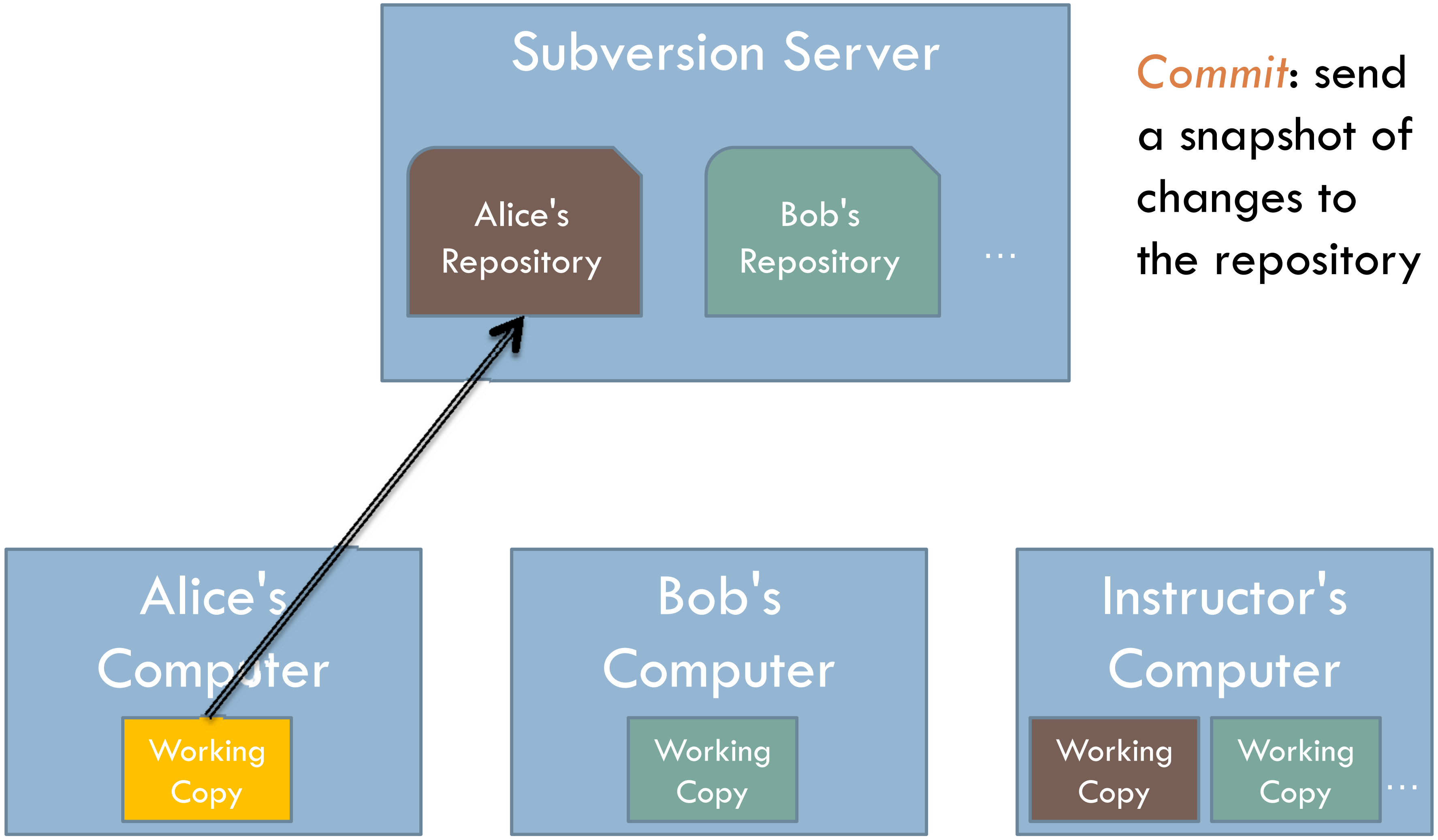
Check out: grab a new working copy from the repository

Version Control Steps—Edit



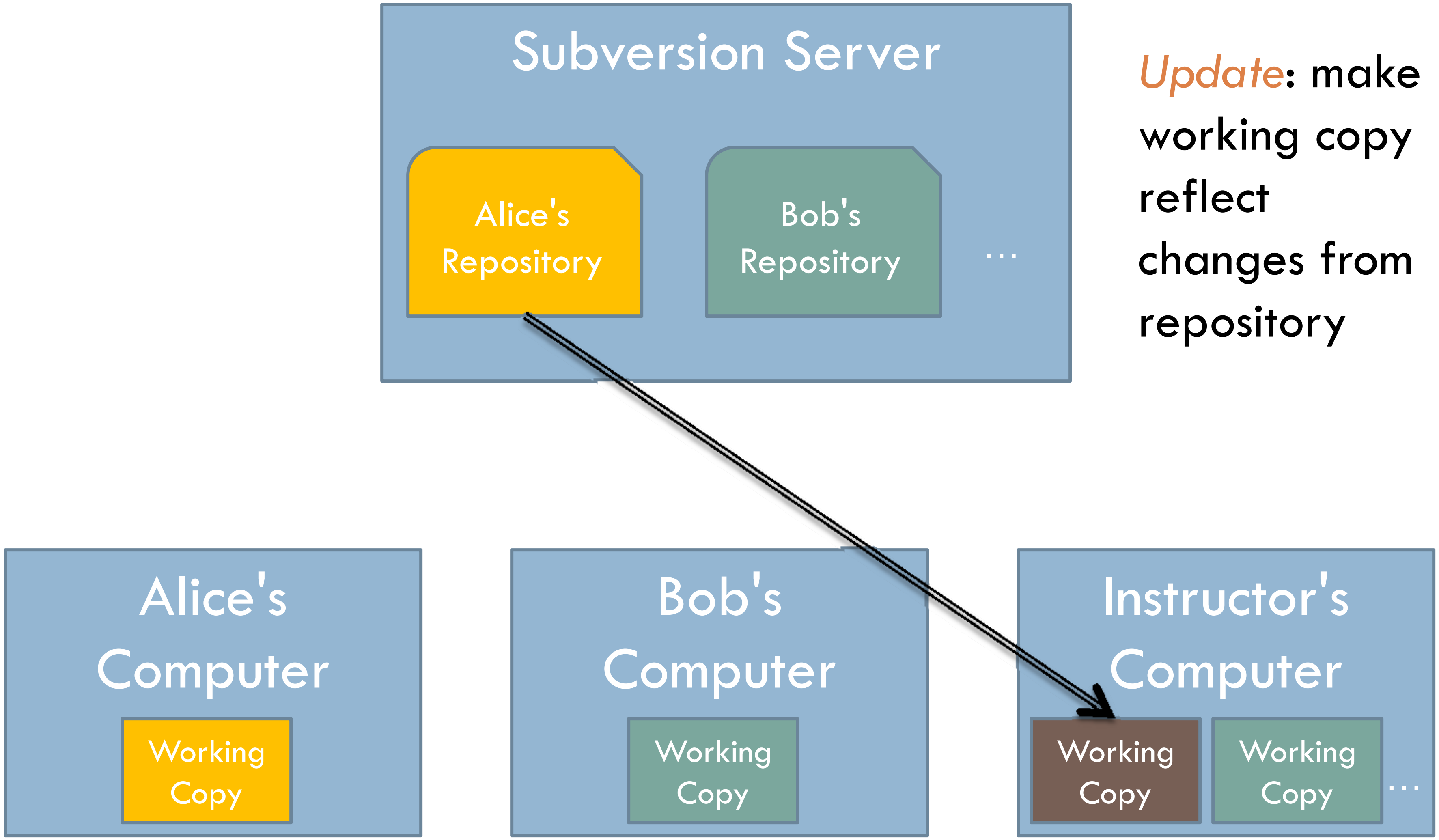
Edit: make ***independent*** changes to a working copy

Version Control Steps—Commit



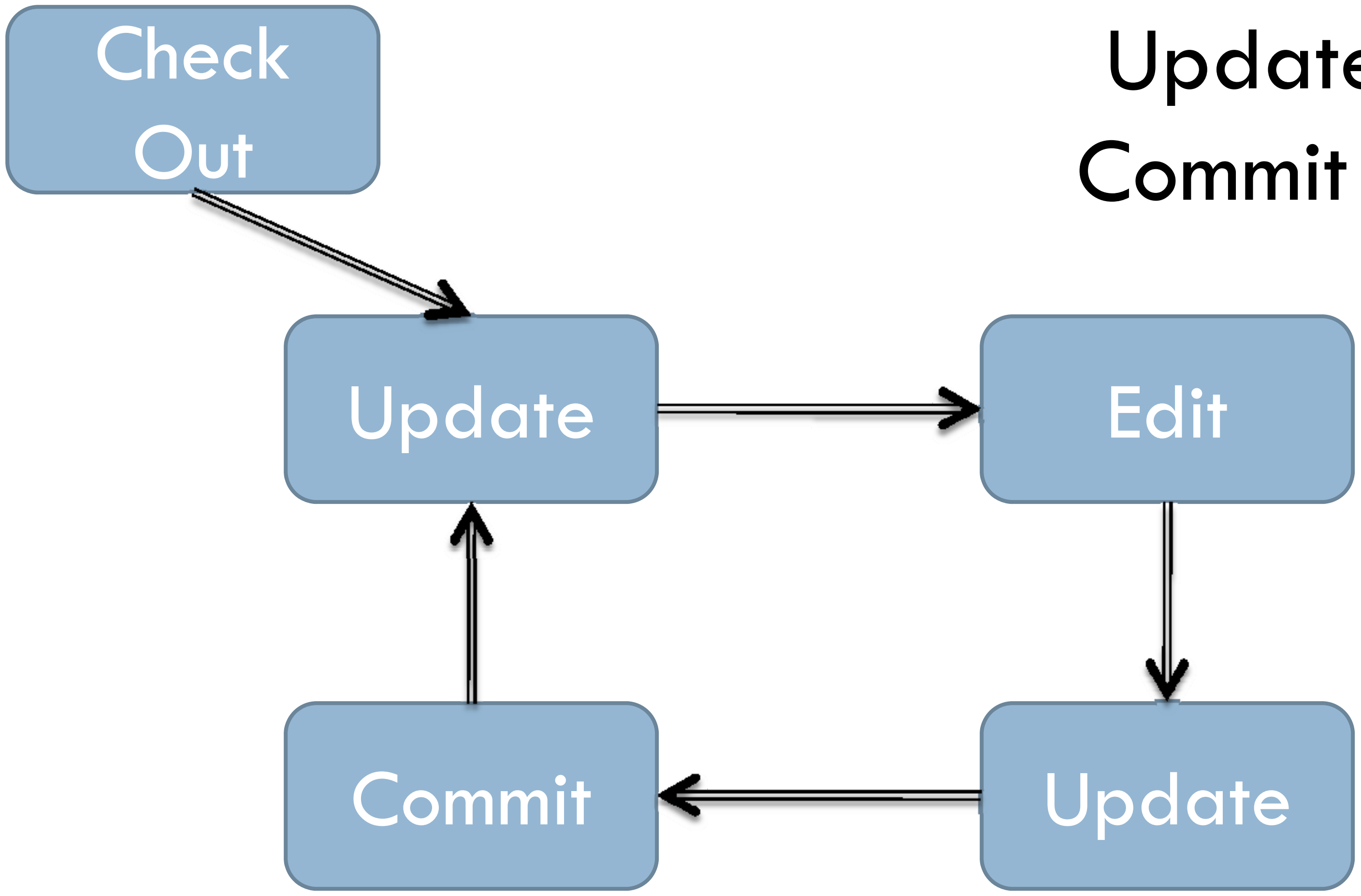
Commit: send a snapshot of changes to the repository

Version Control Steps—Update



Update: make working copy reflect changes from repository

The Version Control Cycle



Update and Commit often!

If you're stuck, get help and see http://www.rose-hulman.edu/class/csse/resources/Subclipse//installation.htm

Check out today's exercise

- Go to the SVN Repository view at the bottom or left of the workbench
  - If it is not there, Window→Show View→Other→SVN Repositories→OK
- Browse your SVN Repository view for Session08 project
- Right-click it, and choose Checkout
- Confirm all of the options presented
- In Package Explorer, find alienFace.py inside your Session08 project
- Add your name to the comments, then commit changes

The object of objects

- Data types for strings and numbers are passive
  - Each represents set of values
    - Passive
  - Each has set of operations
    - Active
- Most modern computer programs are built using Object-Oriented (OO) approach
  - An object is an active data type
    - Knows stuff
    - Can do stuff

The object of objects

- Basic Idea of OO development
  - View a complex system as interaction of simple objects
  - Example: the human body is a complex system

Q2

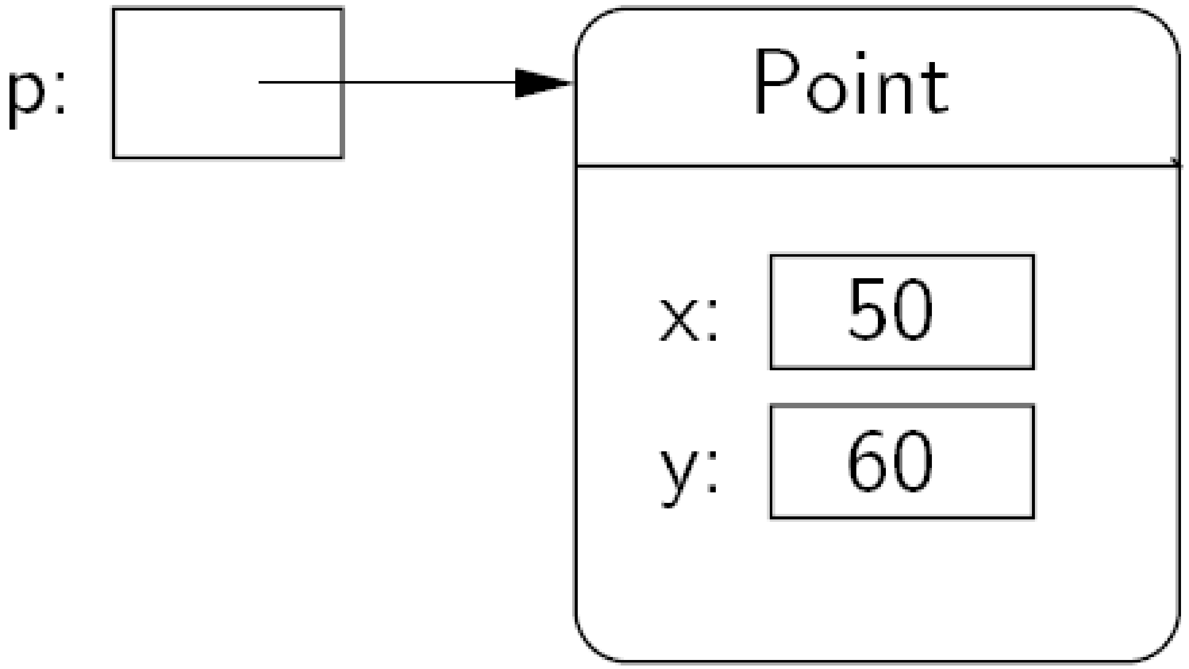
How do objects interact?

- Objects interact by sending each other messages
  - Message: request for object to perform one of its operations
  - Example: the brain can ask the feet to walk
  - In Python, messages happen *via* method calls.
- >>> win = GraphWin()   # constructor
- >>> p = Point(50, 60)   # constructor
- >>> p.getX()   # *accessor* method
- >>> p.getY()   # *accessor* method
- >>> p.draw(win)   # method

Q3

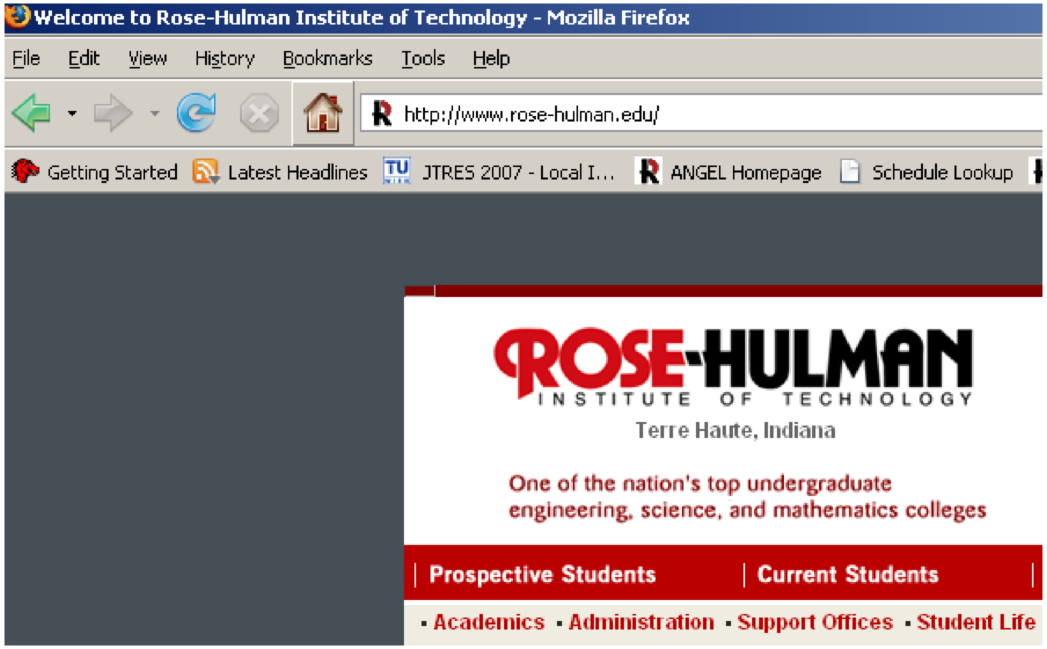
How do objects interact? Point

p = Point(50, 60)



Simple graphics programming

- Graphics is fun and provides a great vehicle for learning about objects
- Computer Graphics: study of graphics programming
- Graphical User Interface (GUI)


Welcome to Rose-Hulman Institute of Technology - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.rose-hulman.edu/
Getting Started
Latest Headlines
JTRES 2007 - Local I...
ANGEL Homepage
Schedule Lookup
ROSE-HULMAN INSTITUTE OF TECHNOLOGY
Terre Haute, Indiana
One of the nation's top undergraduate engineering, science, and mathematics colleges
Prospective Students
Current Students
Academics
Administration
Support Offices
Student Life

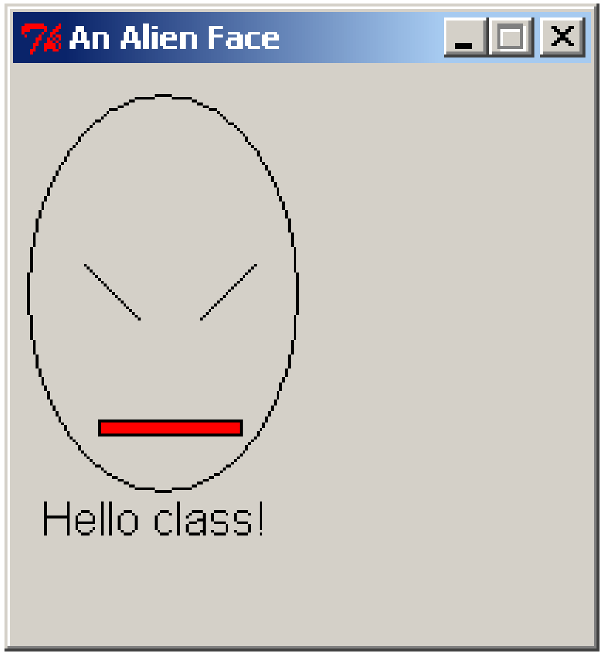
Q4

You choose how to import

- Must import graphics library before accessing it
  - `>>> import zellegraphics`
  - `>>> win = zellegraphics.GraphWin()`
- Another way to import graphics library
  - `>>> from zellegraphics import *`
  - `win = GraphWin()`

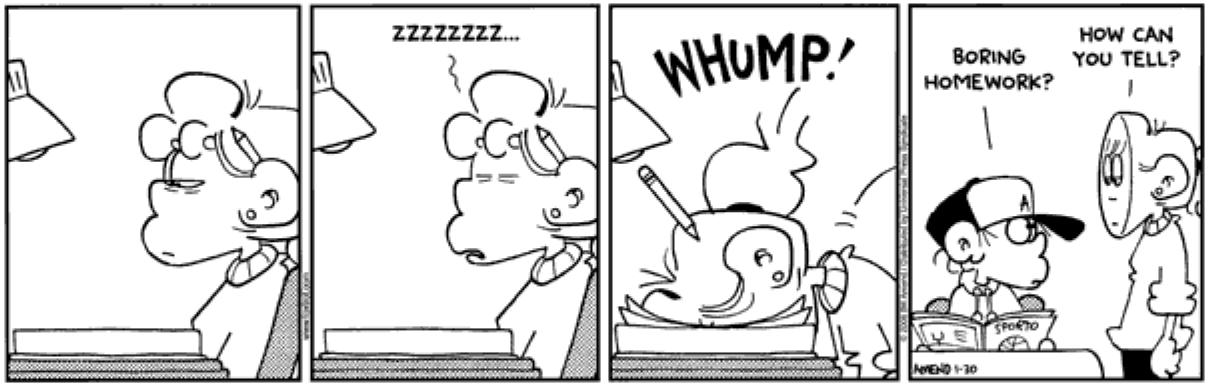
Using graphical objects

- Using different types of objects from the graphics library, draw the following alien face and message



Hello class!

Paige clearly isn't working on homework for CSSE120



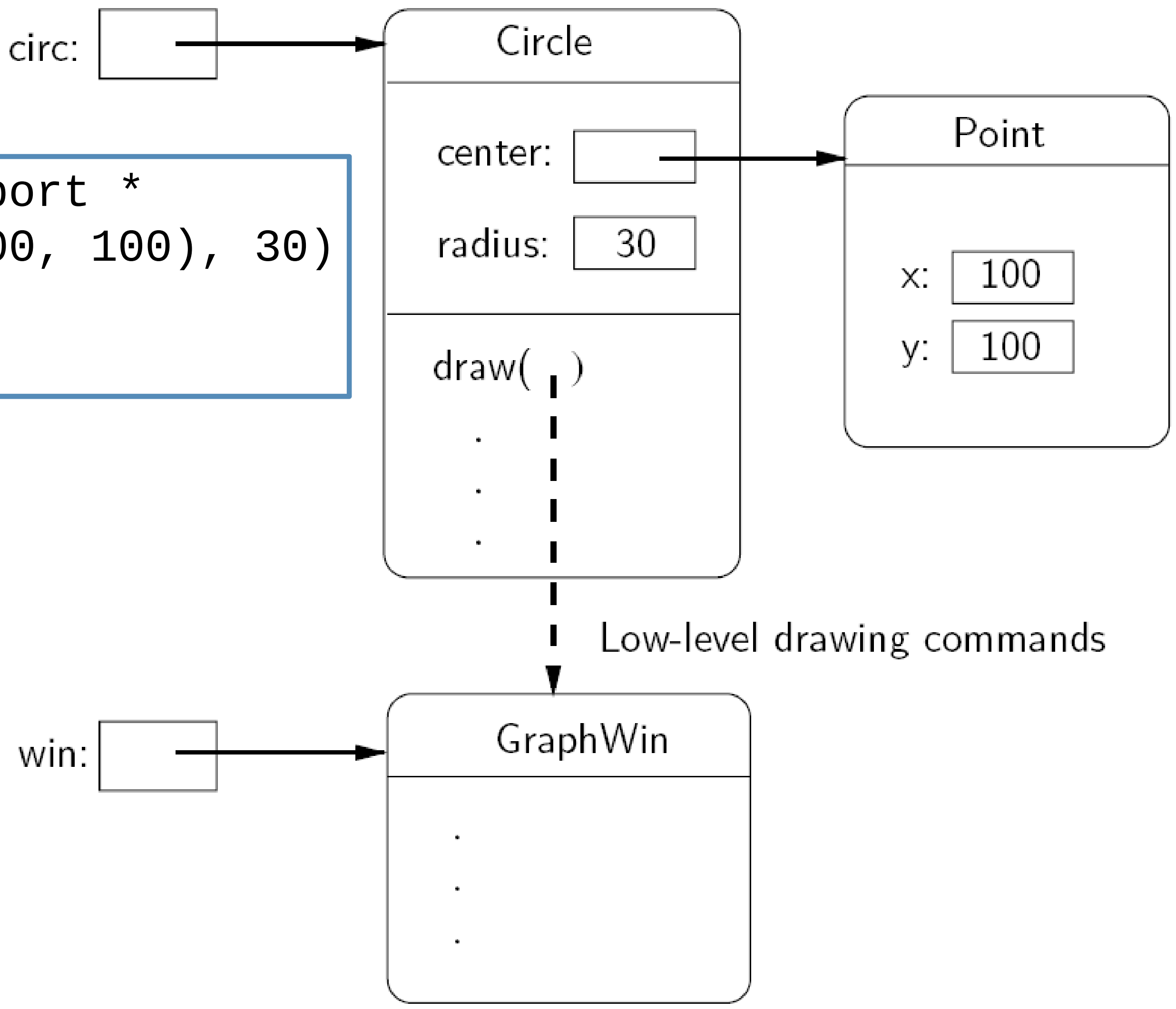
- Preview of tonight's homework:
  1. Read in and draw cool plots from the points in the files you generated in HW5 and 7
  2. Create a cool slideshow picture viewer!

Review: Class and object terminology

- Different types of objects
  - Point, Line, Rectangle, Oval, Text
  - These are examples of *classes*
- Different objects
  - head, leftEye, rightEye, mouth, message
  - Each is an *instance* of a class
  - Created using a *constructor*
  - Objects have *instance variables*
  - Objects use *methods* to operate on instance variables

Object interaction to draw a circle

```
from zellegraphics import *
circ = Circle(Point(100, 100), 30)
win = GraphWin()
circ.draw(win)
```



Interactive graphics

- *GUI*—Graphical User Interface
  - Accepts input
    - Keyboard, mouse clicks, menu, text box
  - Displays output
    - In graphical format
    - On-the-fly
- Developed using *Event-Driven Programming*
  - Program draws interface elements (*widgets*) and waits
  - Program responds when user does something

Q8

getMouse

- `win.getMouse()`
  - Causes the program to pause, waiting for the user to click with the mouse somewhere in the window
  - To find out where it was clicked, assign it to a variable:
    - `p = win.getMouse()`

Q9-10

Mouse Event Exercise

Together, lets' solve the following problem:

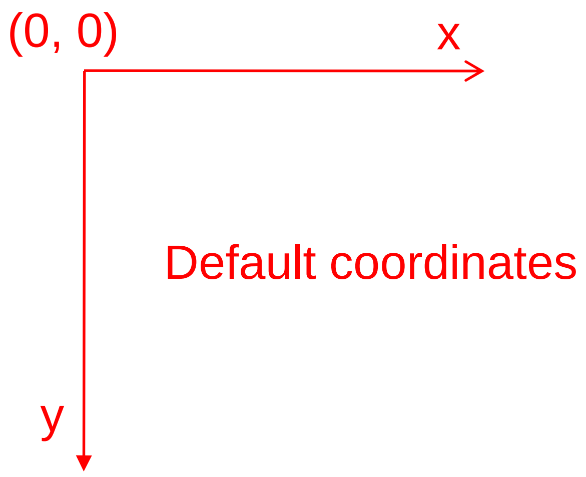
Create a program, `clickMe.py`, with a window labeled "Click Me!" that displays the message *You clicked (x, y)* the first 5 times the user clicks in the window.

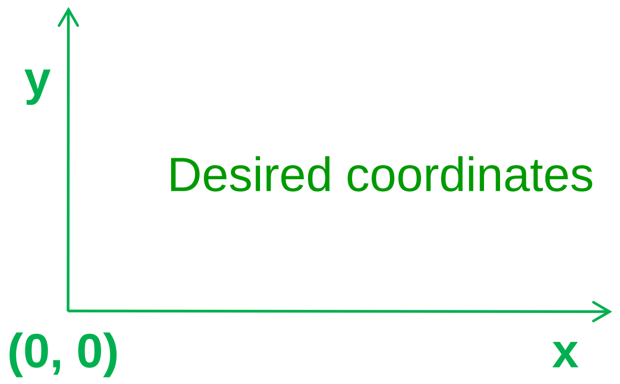
The program also draws a red-filled circle, with blue outline, in the location of each of these first 5 clicks.

The program closes the window on the 6th click

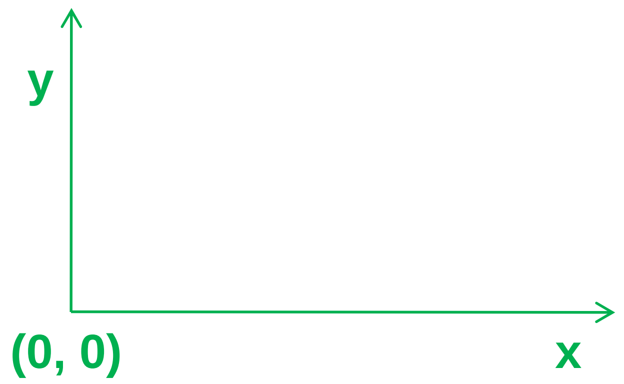
Coordinate systems

- An important use of graphics is to represent **data** visually
  - Example: a bar chart
- We really want (0,0) to be in the lower-left corner





Desired coordinate system



- `win.setCoords(x1, y1, x2, y2)` method from `GraphWin` class
  - Sets the coordinates of the window to run from (x1,y1) in the lower-left corner to (x2,y2) in the upper-right corner.