Lab 2 Feedback

- Getting a little tougher in grading
  - Paying more attention to style (e.g., variable names), efficiency, readability, good output
  - More strict on adhering to problem specification
  - Constants
    - Clear that not everyone was understanding why we use constants
  - Demonstrate program more than once if gets input from user or outcome changes when run again

Lab 2 Feedback: Common Issues

- Over **str**ing
  - If variable assigned value of raw_input, it is a string already
  - "\\" is a string
- Format specifiers
  - Use width when need columns
  - Otherwise, just precision is usually enough
    - "%.2f" -- exactly the width of your number, with two decimals of precision

Testing Programs with `if` Statements

```
clockspeed = input("Enter the clocked speed: ")
speedlimit = input("Enter the speed limit: ")

if clockspeed <= speedlimit:
    print "Continue safe driving practices"
else:
    diff = clockspeed - speedlimit
    fine = 50 + 5 * diff
    if clockspeed > 90:
        fine += 200
    print "Slow down!  You've been fined $" + str(fine) + "."
```
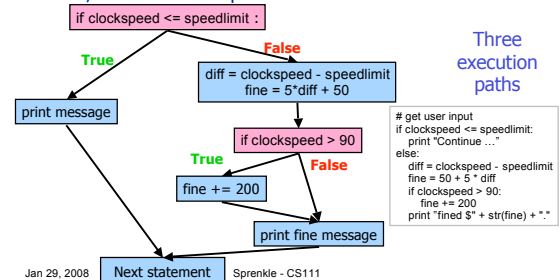
- What are good test cases for this program?

Testing with `if` Statements

- Make sure have test cases that execute each branch in control flow diagram
  - i.e., Each execution path is "covered"

Three execution paths

```
# get user input
if clockspeed <= speedlimit:
    print "Continue …"
else:
    diff = clockspeed - speedlimit
    fine = 50 + 5 * diff
    if clockspeed > 90:
        fine += 200
    print "fined $" + str(fine) + "."
```

Testing with `if` Statements

- Make sure have test cases that execute each branch in control flow diagram
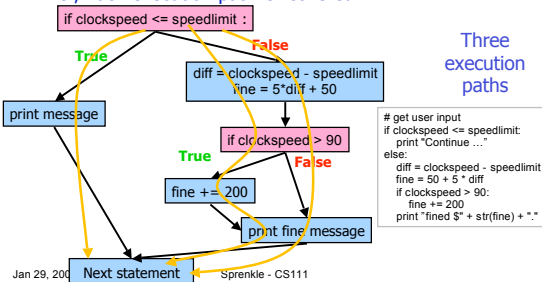 - i.e., Each execution path is "covered"

Three execution paths

```
# get user input
if clockspeed <= speedlimit:
    print "Continue …"
else:
    diff = clockspeed - speedlimit
    fine = 50 + 5 * diff
    if clockspeed > 90:
        fine += 200
    print "fined $" + str(fine) + "."
```

Lab 3 Overview

- Practice Python programming
  - Advanced For loops
  - Using random module
  - If statements
- More "tougher" word problems
  - Work out as much as possible, then move on and come back to problem later with a fresh mind