

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
 - Demonstrate program **more than once** if gets input from user or outcome changes when run again
 - Find errors before I do!

Feb 2, 2010

Sprenkle - CSCI111

1

Lab 2 Feedback: Common Issues

- Over **string**
 - If variable assigned value of raw_input, it is a string already
 - "\\\" is a string
- Only need to use **str**
 - In **print** when don't want the space that is added by the comma and have a number
 - In **input** statement because need to pass in *one string* as the prompt parameter

Feb 2, 2010

Sprenkle - CSCI111

2

Lab 2 Feedback: Doing Too Much

```
print "This molecule weighs", round(hyd_atm*H+cb_atm*  
+o_atm*O ,3), "g/mol"
```

- Instead:
- 1st: Compute the weight
 - 2nd: Round
 - 3rd: Display

```
cTemp = "%.2f" % ((5.0/9.0)*(fTemp-32.0))
```

- Instead:
- 1st: Compute the Celsius temperature
 - 2nd: Display

Why do I care?
Error prone; More difficult to debug, read

Feb 2, 2010

Sprenkle - CSCI111

3

Lab 2 Feedback: Common Issues

- Efficiency

```
i=4  
for j in xrange(1, 9):  
    result = i % j  
    print i, "%", j, "=", result
```

vs

```
for j in xrange(1, 9):  
    i=4  
    result = i % j  
    print i, "%", j, "=", result
```

Feb 2, 2010

Sprenkle - CSCI111

4

Lab 2 Feedback: Common Issues

- Format specifiers
 - Use whenever I say "displays X decimal places"
 - Use *width* when need columns
 - Otherwise, just precision is usually enough
 - "%.2f" -- exactly the width of your number, with two decimals of precision
- Problem 2: asked for **built-in function**
 - round

Feb 2, 2010

Sprenkle - CSCI111

5

Lab 2 Feedback: Formatting

- If I say "display X decimal places" → string formatting (format specifiers)
 - round does not guarantee X decimal places
- Don't mix tabs and spaces and format specifiers
 - Let format specifiers do the work
 - Instead of

```
print "\t   %.2f" % value
```

- Just use format specifier

```
print "%13.2f" % value
```

Easier to figure out output

Feb 2, 2010

Sprenkle - CSCI111

6

String Formatting

- What does this do?

```
print "%.2f deg F is %.2f deg C" % (degreesF, degreesC)
```

Feb 2, 2010

Sprenkle - CSCI111

7

String Formatting

- Equivalent Statements:

```
print "%.2f deg F is %.2f deg C" % (degreesF, degreesC)
```

```
print "%.2f" % degreesF, "deg F is %.2f deg C" % degreesC
```

```
print "%.2f" % degreesF, "deg F is", "%.2f" % degreesC, "deg C"
```

```
formattedFTemp = "%.2f" % degreesF
formattedCTemp = "%.2f" % degreesC
print formattedFTemp, "deg F is", formattedCTemp, "deg C"
```

Feb 2, 2010

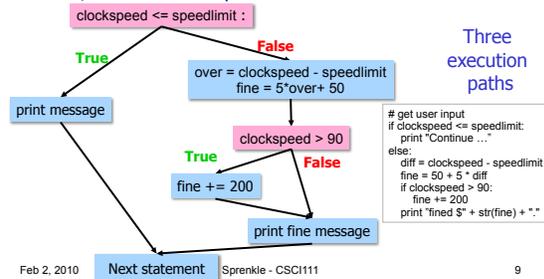
Sprenkle - CSCI111

8

Review: Testing with if Statements

- Make sure have test cases that execute each branch in control flow diagram

> i.e., Each execution path is "covered"



Feb 2, 2010

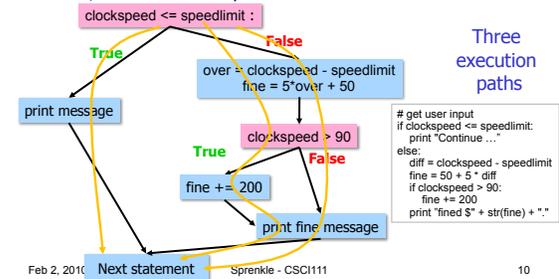
Sprenkle - CSCI111

9

Review: Testing with if Statements

- Make sure have test cases that execute each branch in control flow diagram

> i.e., Each execution path is "covered"



Feb 2, 2010

Sprenkle - CSCI111

10

Review: Efficiency of if statements

- Which is more efficient?

```
if x < 0:
    print x, "is negative"
if x >= 0:
    print x, "is 0 or positive"
```

```
if x < 0:
    print x, "is negative"
else:
    print x, "is 0 or positive"
```

Feb 2, 2010

Sprenkle - CSCI111

11

Review: Efficiency of if statements

- Which is more efficient?

```
if x < 0:
    print x, "is negative"
if x >= 0:
    print x, "is 0 or positive"
```

← Additional computation

```
if x < 0:
    print x, "is negative"
else:
    print x, "is 0 or positive"
```

Feb 2, 2010

Sprenkle - CSCI111

12

Lab 3 Overview

- Practice Python programming
 - Advanced For loops
 - If statements
 - Using random module
- More difficult word problems
 - If and Loops opened up world of new problems
 - Work out as much as possible, then move on and come back to problem later with a fresh mind

Break down problem into smaller pieces
Then, build up to solution!