Lab 3 Feedback

- More difficult problems
- Took off for insufficient testing
- Took off for poor naming, style, indentation, comments

Common Issue

- Using loops for statements that should only be executed once

What is the difference between executing this in the loop and just executing once outside the loop?

Result is the same
But in loop, executes two more times, not changing extra, unnecessary assignments

Common Issue

- Using same variable for different purposes

> Will cause problems/confusion later

Beyond that we don’t need the game variable...

The purpose of x has changed:
- x is the counter for the loop
- x is the random number for determining who won

Instead of

- Instead of

This is better:

Beatles Lyrics Problem

- Using two for loops, a variable with value “She loves you,” and another variable with value “yeah”, print out the Beatles lyrics...

```python
sly = "She Loves You,"
yeah = "yeah"
# prints the line three times
for line in xrange(3):
    print sly,
    # print 2 yeahs, with commas
    for ycount in xrange(2):
        print yeah + ",",
    print yeah # closes the fence post
print "Yee-aahh!"
```

```python
# prints the line three times
for line in xrange(3):
    sly = "She Loves You,"
```

What is the difference between executing this in the loop and just executing once outside the loop?

Result is the same

Problem 3, 4 Efficiency

3
num1 > num2
1 wins
num2 > num1
2 wins
num2 == num1
Tie
End

4
num1 > num2
1 wins
num2 > num1
2 wins
num2 == num1
Tie
End

• How many conditions evaluated?
• Consider case num1=5, num2=3

Problem 4 Efficiency

Which tends to be more efficient?
How many conditions to evaluate?

• num1 > num2
  1 wins
  num2 > num1
  2 wins
  num2 == num1
  Tie
  End

• num2 > num1
  1 wins
  num2 > num1
  2 wins
  num2 == num1
  Tie
  End

Object-Oriented Programming

• Objects combine data and methods together
  Provides interface (the methods)
  that users interact with

  Hides internal data structures, implementation

  Optionally may return something back

  Use an Application Programming Interface (API)
  to interact with a set of classes

OO Terminology Summary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>A data type. Defines the data and operations for members of the class</td>
<td>string, TV, GraphWin</td>
</tr>
<tr>
<td>Object</td>
<td>An instance of a specific class</td>
<td>animal, myTV, window</td>
</tr>
<tr>
<td>Method</td>
<td>Operations you can call on an object</td>
<td>setBackground(&lt;color&gt;), getWidth()</td>
</tr>
<tr>
<td>Constructor</td>
<td>Special method to create an object of a certain type/class</td>
<td>GraphWin(), str(1234)</td>
</tr>
</tbody>
</table>

Always need to create/construct an object before using

OO Review

• How do we create an instance of a Rectangle?
  > rect = Rectangle(Point(10,10), Point(30, 40))

• Draw the rectangle?
  > win = GraphWin()
  > rect.draw(win)

• Shift the instance of the Rectangle class to the right 10 pixels
  > rect.move(10, 0)

• What are the x- and y- coordinates of the upper-left corner of the Rectangle now?
  > upperleft = rect.getPt()
  > upperleft.getX(), upperleft.getY()

upperleft is a Point object
OO Hints

- Use API to do work for you
  - Call appropriate methods for the type/class of object
- Keep in mind the type of the object that you're dealing with

<table>
<thead>
<tr>
<th>Object Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>rect</td>
<td>Rectangle</td>
</tr>
<tr>
<td>win</td>
<td>GraphWin</td>
</tr>
<tr>
<td>upper_left</td>
<td>Point</td>
</tr>
</tbody>
</table>

Lab 4 Overview

- Practice Python programming
  - Advanced for loop
  - Use sys module
  - Indefinite loops (while)
  - Object-oriented programming using a third-party library
    - Creates a .pyc file
    - Causes printing havoc
- Due on Friday before class