Objectives:
- Understand the selection statements: if, if-else, and if-elif-else
- Understand the count-controlled loop: for statement
- Understand the conditional iteration loop: while statement

Part A: if-elif-else statements
Write a Python program that prompts the user for a package weight and prints the total shipping charge for the package. Use the following table to determine the rate per pound:

<table>
<thead>
<tr>
<th>Weight of Package</th>
<th>Rate per Pound</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 pounds or less</td>
<td>$1.10</td>
</tr>
<tr>
<td>Over 2 pounds, but not more than 6 pounds</td>
<td>$2.20</td>
</tr>
<tr>
<td>Over 6 pounds, but not more than 10 pounds</td>
<td>$3.70</td>
</tr>
<tr>
<td>Over 10 points</td>
<td>$3.80</td>
</tr>
</tbody>
</table>

Test your program thoroughly to make sure that it works correctly. Record the package weights you used to test your program and their corresponding shipping charges in the following table:

<table>
<thead>
<tr>
<th>Weight of Package (input)</th>
<th>Shipping Charge (output)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After you have your program working correctly (“debugged your program”), raise your hand and demonstrate your program.

Part B: for-loop
A “count-controlled” for loop using the xrange function to average the scores for a class is given below and in the file PartC_average_with_for.py:

```python
# Calculates the average score using a counter-controlled for-loop
def calculate_average(number_of_scores):
    total = 0.0
    for counter in xrange(number_of_scores):
        score = input("Enter score #" + str(counter+1) + ": ")
        total = total + score
    if number_of_scores > 0:
        print "The average score is", total / number_of_scores
```

Write a program that calculates the amount of money a person would earn over a user-specified number of days if their salary is one penny the first day, two pennies the second day, and continues to double each day. The program should prompt the user for the number of days, and then displays a (well formatted) table showing the day #, the salary for that day, and total salary paid from day 1 through that day.

After you have your program working correctly (“debugged your program”), raise your hand and demonstrate your program.
Part C: while loops

“Count-controlled” for loops work great if we know how many times we’d like to loop before we start the loop, but often we don’t. Instead, we want to use a while-loop to continue looping until some stopping condition. A while statement repeatedly executes the body of the loop as long as the condition evaluate to True. The syntax of a while statement is:

```python
while <condition>:
    statement_1
    statement_2
    statement_3
```

Write a program similar to Part C that calculates the amount of money a person would earn over a period of time if their salary is one penny the first day, two pennies the second day, and continues to double each day. This version of the program should determine how many days a person would need to work before they received a million dollars per day in salary.

After you have your program working correctly (“debugged your program”), raise your hand and demonstrate your program.

If you complete all parts of the lab, nothing needs to be turned in for this lab. If you do not get done today, then show me the completed lab in next week’s lab period.

Make sure that you take your USB-drive out of the computer and log off before you leave.