CISC106 Fall 2011 Lab02

- This lab an all subsequent labs will be due Sunday at 11:55 PM EDT on Sakai.
- The preparation problems below are to develop your understanding without creating extra work for you or the TA; these problems will not be graded. Be sure to read and understand them - they will help with the problems you must submit for grading

Preparation (do not submit for grading)

The following preparation problems are a refresher of what we went over in IDLE in class last Wednesday.

1. Fire up IDLE so that you have a shell window. At the prompt[1] type in:

   ```
   >>> 10 * 8 + 55/5 - 2
   ```

   Note what gets *returned* (in this case, simply displayed) on the next line.

2. Start a new file by selecting File→New Window. Add the following line:

   ```
   print(10 * 8 + 55/5 - 2)
   ```

   Now run this file by selecting Run→Run Module. You'll be asked to save the file, give it a name like `atest` (it really doesn't matter what you name it.) The number printed out in the console should be the same as the number you got when you typed the expression directly into the console (the same number should be shown above and below the line with all the equals signs and RESTART.)

Programs (to be graded)

1. Give the *types* of the following expressions:

   (a) $1+1+1+1$

   (b) `'What type am I?'`

   (c) $99/11$

   (d) `True`

   (e) `'Hey ' + 'Now'`

2. Download the file `messed_up.py` from the labs folder on the course website. Without doing any of the following:

   - removing or adding any lines of code to the file
   - altering the last line (the one with the print statement)
   - changing any of the other lines to print statements

   fix all the errors so that the string `'Bluh Bluh Bluh'` is what's printed when you run the file. (Run `messed_up.py` selecting Run→Run Module from its window.)

[1] NB: from henceforth, I will preceed any input meant for the Python shell by the Python prompt, >>>.

3. Show the progression of the state diagram for the block of code in `state.py` (again, in the labs folder on the course website.) You should show the state after each line of code executes, separating the various states by a blank line.[2] For example, the state for the following block of code:

```
x = 10
y = 'Pigeons'
```

would be:

```
x -> 10

x -> 10
y -> 'Pigeons'
```

You should submit your corrected `messed_up.py`, the files containing your answers for parts 1 and 3, and any other docs required by your TA on Sakai.

[2] Copying and pasting will probably be very helpful here.