CSC 110 – Lab 1
Using Python in the CSC Lab

Names: _____

Introduction
The purpose of this lab is to familiarize yourself with the Computer Science Department Lab and the Python environment in the CSC lab. You will also be given the opportunity to practice some basic concepts of Python so that we can build to developing algorithms in later labs.

Logging into the Lab Computers
Each student will be given a username and password to use in the lab each week. Although you will be working in pairs each week, it is a good idea for each student to log into a computer so you can both get the practice that the lab provides.

The computers in the lab can be booted to run either Mac OSX or Windows 7. For this lab we will use the Mac side, although Python can be run from either Mac or Windows depending on your preference.

- Log into the lab computer using your given username and password.

Using Python
Once you are logged in, you will run IDLE, the simple interface that we will use to run Python and to create simple Python programs.

- From the Applications folder, choose Python3.2, and then choose IDLE.
- You will see a window labeled Python Shell. This is the command window in which you can type Python commands and run Python programs. When you are finished using Python, you can quit by choosing the IDLE menu, and choosing Quit.

Interactive Mode
Python's interactive mode is an interpreter. That is, it evaluates and executes the commands that you type. The Python Shell window provides a place for you to enter commands to be interpreted.

- Type the following commands at the command line and indicate what results you see:

```
2 + 2                  _____________________

25 / 5                 _____________________

24 / 5                 _____________________

4 * 5                  _____________________

4 ** 2                 _____________________

4 * (8 – 5) ** 2       _____________________
```

Instructor Initials: _____

Variables

A variable is a named storage location. It is used to hold values that you will want to use later in your program. A variable can hold whatever type of value you assign to it. We use an assignment operator (=) to give a variable a value. Note that when using the assignment operator, we always have the following format:

```
<<variableName>> = <<expressionToEvaluate>>
```

The name of the variable is on the left side of the "=" operator, and the right side is a mathematical expression, made up of values, operators, and other variables that will be evaluated, and the final value will be stored in the variable on the left side.

- To practice using variables, type the following commands at the command line and describe the results. Whenever you get an error message, explain the reason for the error.

```
a                      _____________________


a = 3                  _____________________
```

```
a                    ____________________

b = a + 4            ____________________

b                    ____________________

7 = unluckyNumber    ____________________

a + b = c            ____________________

c = a + b            ____________________

c                    ____________________
```

- Now consider this set of statements:

```
x = 10
x = x + x
x = x - 5
```

What will be the value of x after you execute these statements?

- Check your results and explain.

Instructor Initials: _____

Exercise

- Write a program (a group of statements) to swap the values of two variables. Start with the following:

```
x = 200
y = 500
```

  Write code that exchanges their values: the value of x after your code runs must be 500 and the value of y after your code runs must be 200. Your program does not need to print any output. Test your code in the python command line and write the final result here:

Instructor Initials: _____

Lists

A list is a sequence of several variables, grouped together under a single name. Instead of writing a program with many variables `x, y, z, ...,` you can define a single variable `numList` and access its members `numList[0], numList[1], numList[2]`, etc. More importantly, you can put other expressions and variables inside the square brackets, like `numList[i]` and `numList[i+1]`. This allows us to deal with arbitrarily large data sets using just a single small piece of code.

One way to create a list is by enclosing several values, separated with commas, between square brackets:

```
myList = ["the first value in the list", 1999, 4.5]
```

This creates a list called `myList` of length 3. Each element of the list gets a number called its index: the initial element has index 0, the next element has index 1, and so forth. The individual variables comprising the list have the names

```
«listName»[«indexNumber»]
```

- In the example above, what is the value of:

`myList[0]` _____

`myList[2]` _____

- Create the following list in your python command line window:

  `numbers = ['one', 'two', 'three']`

  Now type:

  `numbers`

  What is displayed?

- Why do you need the single quotes around the values 'one', 'two' and 'three'? What would happen if you did not have them? If you are not sure, try it out and see.

Instructor Initials: _____

- Let's access the different elements of this list. Type the following on the command line and describe and explain the results:

```
numbers[0]
```
_____

```
numbers[3]
```
_____

- You can change the value of list elements by assigning a new value to an individual element. Try these assignments and then display the entire list.

```
numbers[0] = 0
```

```
numbers[2] = 2
```

What are the contents of the list now? (*Hint:* type `numbers` on the command line to show the values in the list).

Instructor Initials: _____

List Operations: We can use some built-in functions to work with lists. To get the length of a list, we use the `len( )` function.

- On the command line, type the following and write the result:

  `len(numbers)` _____

We can also add values to lists, and add lists together using what look like mathematical operators. To merge two lists together (an operation we call concatenation) we use the '+' operator.

- On the command line, create the following list:

  ```
  words = ['this', 'is', 'my', 'list']
  ```

  Now let's merge this list with the one we created above:

  ```
  WordsAndNumbers = words + numbers
  ```

  What are the contents of the list `WordsAndNumbers`?

  What would be the result if you change the order in your expression and instead typed:

  ```
  WordsAndNumbers = numbers + words
  ```

  Try it out and see if you are right:

We can also use this concatenation operator to add elements to a list.

- Type the following command and then display the contents of the list:

```
numbers = numbers + [3]
```

What are the contents of the list now?

Instructor Initials: _____

- Create a list containing the following ten midterm grades for a class of students, call it `midtermGrades`:

```
85
95
84
83
99
100
75
77
89
90
```

Show the command you will use here:

- Now suppose that the student who received an 84 came to you and found an error in your grading. What command would you use to change the grade of 84 to an 86?

- One more student took the midterm late because she was sick when it was originally given. You have graded it and computed a grade of 87 for this student. Add this new grade to the end of the list, and then display the contents of the list again. What commands did you use to do this?

- You want to make sure that you have a grade for all of the students in the class. What command would you use to find out how many grades are in the list?

Instructor Initials: _____

Input and Output

Most algorithms that we will develop this semester will require that we request some input from the user (input), and that we display a result at the end (output). In Python, there are several ways to provide input and output. We will consider the simplest methods for now.

Input: The `input()` statement is used to accept input from the keyboard and save it in a variable. It defaults to assume that the data you are entering is a string of characters. If you want to enter a number, you have to explicitly evaluate the data as a number.

- Type the following statements on the command line and write the results that you see. Enter different types of data for each to see how they can be entered. When you see the command line doing nothing but blinking, it is waiting for you to enter data from the keyboard. Type in a value and hit the <return> key.

```
myString = input()
```

The computer will be waiting for you to enter a string. Type in any string and hit return. Now type the name of the variable to see what it contains.

```
myString
```

Now let's do some user input with numbers. Type the following command:

```
number = int(input())
```

The computer will be waiting for you to enter a number. Type in any number and hit return. Then type the name of the variable to see what it contains.

```
number
```

You can also use the `input()` command to prompt the user for the data you are looking for.

- Type the following statements. Note that instead of a blank line, you will see the prompt for the data you are asking for.

```
name = input("Enter your name: ")
```

```
gradYear = int(input("Enter your expected year of graduation: "))
```

Output: Using the command line interpreter as we have been doing so far, we can display the contents of any variable just by typing the name of the variable. However, there is a more versatile way to display values using the `print` statement.

Using the `print` statement, you can print numbers, strings, values of variables, and evaluations of expressions.

- Try typing each of the following and write down the results:

```
print("Hello World")
```

```
print("The grades for the midterm are: ", midtermGrades)
```

```
i = 4
print("Student ", i, "received a grade of ", midtermGrades[i])
```

```
print("The average of the numbers 89, 99 and 75 is: ", (89 + 99 +
75)/3)
```

```
name = "Donald Duck"
age = 16
print("The age of ", name, " is ", age)
```

Instructor Initials: _____

Exercise

- Write a short program (set of instructions) that asks the user to enter his/her name and age. Then compute the year the person was born and print out a statement indicating the result. Show the program to your instructor and show that it works.

Instructor Initials: _____