

Laboratory Partner 1: Amethyst O'Connell

Laboratory Partner 2: \_\_\_\_\_

**CSCI 1523: Class Laboratory 2B**  
**Variable naming, operators, comments**  
**and**  
**Elementary Coding**

**Instructor: Warren W. Sheaffer**

August 20, 2016

## Storing data in in variables

Below find a series of questions which relate to numerics and literals used in the Python language. Students should attempt to complete these exercises without the use of the internet or your text. **Sometimes** guessing then correctly identifying an error can help a new user become familiar with a language more quickly so try to work without reference.

Following completion of the exercise uses the inactive Python prompt to test your answer:

1. `>>>0variable = 6.7`      Results and reason: SyntaxError: Invalid Syntax

Ya can't put a number in a variable name

2. `>>>newString = my new string`      Results and reason: SyntaxError: Invalid Syntax

Ya gotta use those single quotes for the strings

3. `>>>myfloat = 45.6`      Results and reason: sets number to variable myfloat

4. `>>>myFloat = 4.3e-25`      Results and reason: sets number to variable myFloat

myfloat remains 45.6 because variables are case sensitive

5. `>>>_myint = 25`      Results and reason: sets variable myint to 25, myint

has not been set, because variables can include underscores

6. `>>>myString = 'a little string"`      Results and reason: SyntaxError: EOL

while scanning string literal. Gotta use the same kind of quotes for ya strings.

7. `>>>$$myVar = 22.5`      Results and reason: SyntaxError: invalid syntax

Can't use dollar signs in your variables, this isn't CLI

Now some questions on *format*

8. `>>>format(7/8)`      Results and reason: Just turns it into a decimal, ya haven't specified a format for that decimal

9. `>>>format(7/8, '.1f')`      Results and reason: a nearly infinite amount of lines are created, it won't even let you run this without that parenthesis

10. `>>>format(25*3, '.1f')`      Results and reason: You get the result of the multiplication with a .0 at the end, b/c you've specified that you want one decimal place

Take care to answer each of the questions above collaboratively and to the best of your joint knowledge. Once you have completed each of the questions above use the interactive Python prompt on your workstation to test your solution.

We are using Python version 3.4.2 in this course so please be certain to check your prompt to insure you are testing for the correct answers.

Correct the solutions yourself and try to do a critical self-assessment of your knowledge of this topic.

## Manipulating integer data

Below find a series of questions which relate to integer operations used in the Python language. Students should attempt to complete these exercises without the use of the internet or your text. **Sometimes** guessing then correctly identifying an error can help a new user become familiar with a language more quickly so try to work without reference.

Assume the following variable assignment when completing your work on the problems below:

- AAA = 8
- BBB = 20
- CCC = 1
- DDD = 11
- EEE = 13
- FFF = 5

Following completion of the exercise uses the inactive Python prompt to test your answer:

1. `>>>A = BBB // AAA`  
`>>>print("The value of A is: ",A)` Results and reason: The value of A is: 2  
There are two spaces, also it's division without the modulus, so no decimals.
2. `>>>B = BBB % AAA`  
`>>>print("The value of B is: ",B)` Results and reason: The value of B is: 4  
Gives me the modulus cause that's what I asked for, also double spaces grrr.
3. `>>>C = DDD * 3 + FFF`  
`>>>print("The value of C is: ",C)` Results and reason: The value of C is: 38  
(11\*3)+5, it's python's order of operations! Also, double spaces.
4. `>>> D = EEE - FFF * 4`  
`>>>print("The value of D is: ",D)` Results and reason: The value of D is: -7  
13 - (5\*4), it's python's order of operations! Also, double spaces.

5. `>>> E = EEE // FFF + CCC`  
`>>> print("The value of E is: ", E)` Results and reason: The value of E is: 3  
(13//5)+1, it's python's order of operations, which still works despite weird division
6. `>>> F = EEE // (FFF + CCC)`  
`>>> print("The value of F is: ", F)` Results and reason: The value of F is: 2  
Parenthesises work in Python's order of operations. More double spaces.
7. `>>> G = FFF // 2 * 2`  
`>>> print("The value of G is: ", G)` Results and reason: The value of G is: 4  
(5//2)\*2, it's the python order of operations! More double spaces.

Take care to answer each of the questions above collaboratively and to the best of your joint knowledge. Once you have completed each of the questions above use the interactive Python prompt on your workstation to test your solution.

We are using Python version 3.4.2 in this course so please be certain to check your prompt to insure you are testing for the correct answers.

Correct the solutions yourself and try to do a critical self-assessment of your knowledge of this topic.

## Mixed type operations on numeric data

Below find a series of questions which relate to assignment operations using mixed types in the Python language. Students should attempt to complete these exercises without the use of the internet or your text. **Sometimes** guessing then correctly identifying an error can help a new user become familiar with a language more quickly so try to work without reference.

Assume the following variable assignment when completing your work on the problems below:

- AAA = 9
- BBB = 13
- XXX = 17.9
- YYY = 1.2e+3
- ZZZ = 5.625e-12

Please note we have excluded the use of the `>>>` symbol here as we are assuming it's use.

Following completion of the exercise uses the inactive Python prompt to test your answer:

1. `A = int(XXX)`  
`print("The value of A is: ", A)` Results and reason: The value of A is: 17  
Converts that floaty point into an integer by just chopping off the end like monsters

2. `B = int(XXX + 4.8)`  
`print("The value of B is: ", B)` Results and reason: The value of B is: 22

Order of ops meets chopping off the end of a floating point like a monster

---

3. `C = int(XXX-AAA//2)`  
`print("The value of C is: ", C)` Results and reason: The value of C is: 13

Order of ops meets chopping off the end of a floating point like a monster

---

4. `D = int(ZZZ)`  
`print("The value of D is: ", D)` Results and reason: The value of D is: 0

See the obvious problem with chopping the end of a floaty point? It ate the whole #

---

5. `E = int(YYY - ZZZ)`  
`print("The value of E is: ", E)` Results and reason: The value of E is: 1199

More problems with chopping, 1199.9999999999943 is the number w/out chopping

---

6. `X = float(XXX)`  
`print("The value of X is: ", X)` Results and reason: The value of X is: 17.9

It turns a floaty point into a floaty point... nothing really happens.

---

7. `Y = float(BBB // 2)`  
`print("The value of Y is: ", Y)` Results and reason: The value of Y is: 6.0

The .0 is to indicate that this whole number is indeed a floaty point.

---

Take care to answer each of the questions above collaboratively and to the best of your joint knowledge. Once you have completed each of the questions above use the interactive Python prompt on your workstation to test your solution.

We are using Python version 3.4.2 in this course so please be certain to check your prompt to insure you are testing for the correct answers.

Correct the solutions yourself and try to do a critical self-assessment of your knowledge of this topic.

## Programming application - Quadratic Equation

Consider the quadratic equation:

$$Ax^2 + Bx + C = 0$$

and its solution:

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

where x is the unknown and A, B and C are constants (with A not equal to 0). A quadratic equation has two solutions (called roots), which may not be distinct values and which may not be real values.

Develop a program to compute the two roots of a quadratic equation.

Test your program on the following values:

1. **A=4, B=0, C= -36**

Root 1: \_\_\_\_\_

Root 2: \_\_\_\_\_

---

2. **A=1, B=5, C= -36**

Root 1: \_\_\_\_\_

Root 2: \_\_\_\_\_

---

3. **A=2, B=7.5, C= 6**

Root 1: \_\_\_\_\_

Root 2: \_\_\_\_\_

---

4. **A=0, B=3.5, C= 8**

Root 1: \_\_\_\_\_

Root 2: \_\_\_\_\_

---

5. **A=5, B=0, C= 6.5**

Root 1: \_\_\_\_\_

Root 2: \_\_\_\_\_

---

Demonstrate your completed program to your Instructor.