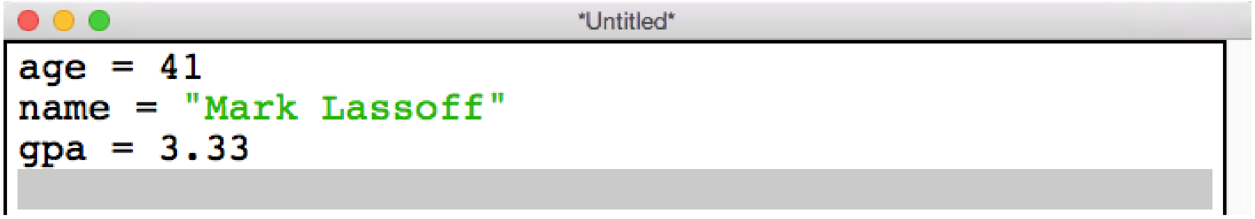
learntoprogram.tv

Activity 10.6 IO and Variables

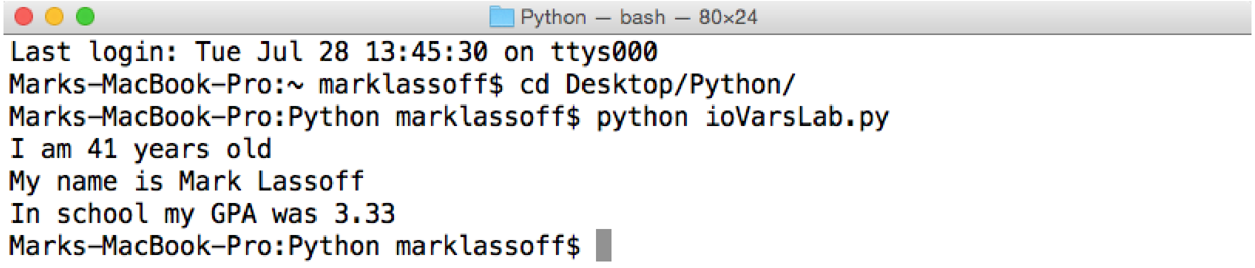
In this activity you'll write a Python program according to the guidelines provided. Throughout the industry deliverables for applications are known as **requirements**. A common error in enterprise-level programming is for an application to fail to meet some of the requirements listed. Another common error is for developers to create additional requirements as they are coding. This is known as **scope creep**. As you work through the balance of the exercises in this course pay special attention to the requirements listed and make sure your solution meets all of them.

1) Using your Python development environment, IDLE, create a new file. In this file declare three variables: age, name, and gpa. Assign them unique values. See the example below:



```
age = 41
name = "Mark Lassoff"
gpa = 3.33
```

2) Using **print** statements and the **format()** function, write Python code print out the values of your variables with the strings in the output below. Don't forget to save your code with the .py extension before testing it in your command line tool.



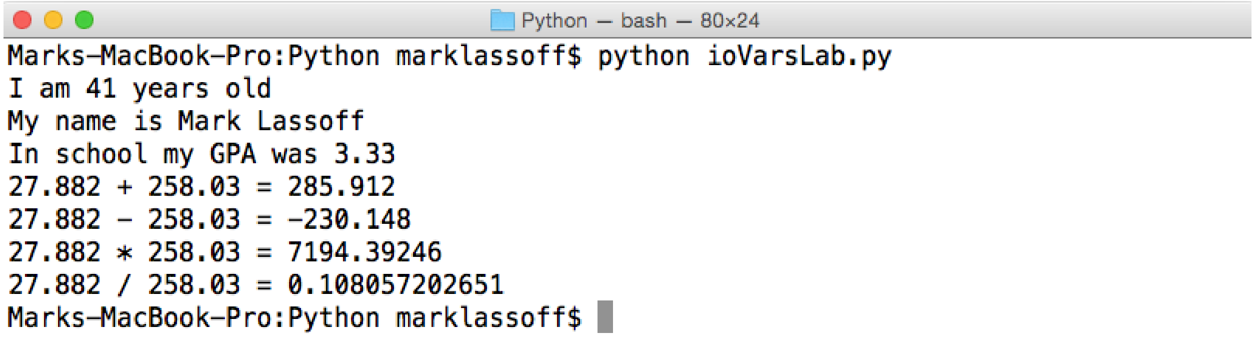
```
Last login: Tue Jul 28 13:45:30 on ttys000
Marks-MacBook-Pro:~ marklassoff$ cd Desktop/Python/
Marks-MacBook-Pro:Python marklassoff$ python ioVarsLab.py
I am 41 years old
My name is Mark Lassoff
In school my GPA was 3.33
Marks-MacBook-Pro:Python marklassoff$ 
```

3) At the bottom of the code you have written so far, define the follow variables:

operand1 = 27.882
operand2 = 258.03

4) Creating the necessary expressions, create the following output (screenshot below). **Note**: Do not “hard code” the values of the operand variables into strings. Use the replacement operator {} and format() function. Also note that you can use multiple sets of replacement operators in a print command. For example the following would produce the output “**2 + 2 = 4**”

```
print “{} + {} = “.format(2,2,2+2)
```

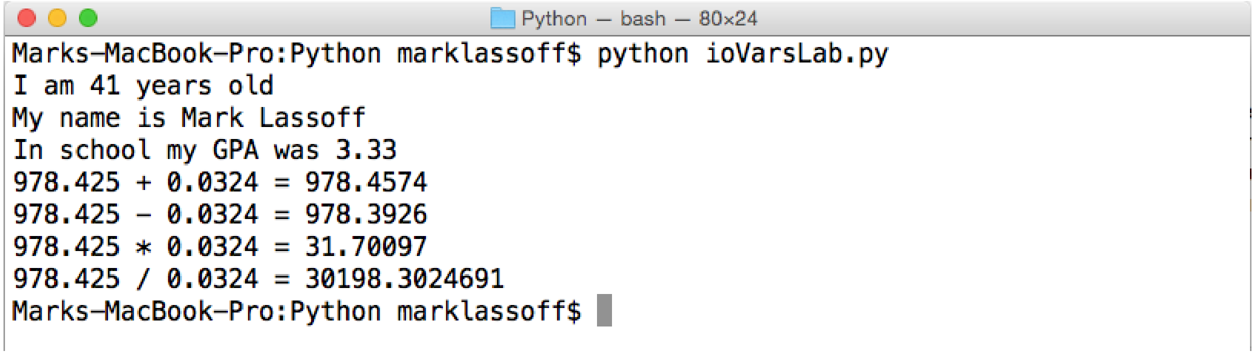


```
Marks-MacBook-Pro:Python marklassoff$ python ioVarsLab.py
I am 41 years old
My name is Mark Lassoff
In school my GPA was 3.33
27.882 + 258.03 = 285.912
27.882 - 258.03 = -230.148
27.882 * 258.03 = 7194.39246
27.882 / 258.03 = 0.108057202651
Marks-MacBook-Pro:Python marklassoff$ █
```

5) Edit the code where your operands are declared to appear as follows:

```
operand1 = 978.425
operand2 = 0.0324
```

6) Run your code again. Without any further changes to the code, you should get the result below:



```
Marks-MacBook-Pro:Python marklassoff$ python ioVarsLab.py
I am 41 years old
My name is Mark Lassoff
In school my GPA was 3.33
978.425 + 0.0324 = 978.4574
978.425 - 0.0324 = 978.3926
978.425 * 0.0324 = 31.70097
978.425 / 0.0324 = 30198.3024691
Marks-MacBook-Pro:Python marklassoff$ █
```

Note that changing the value of the variables changed the result for all four lines you printed out. Had you “hard coded” the numbers into the print output you would have had to make eight separate changes to the code— much increasing your chances of making an error. This concept is known as **indirection** and is critical in computer science. The simplest definition of indirection is **“**the ability to reference something using a name, reference, or container instead of the value itself.”

Any time you will be referencing something multiple times you want to make use of indirection. This way if the value you are referencing changes you only have to make the change in one place.

7) Aside from the common mathematical operators that you have used so far, there is another operator that is frequently used in programming called modulus (or “mod”, for short). The modulus operator provides the remainder after division. After you last line of code add the lines of code below:

```
print "{} % {} = {}".format(9,3,9 % 3)
print "{} % {} = {}".format(112,10, 112 % 10)
```

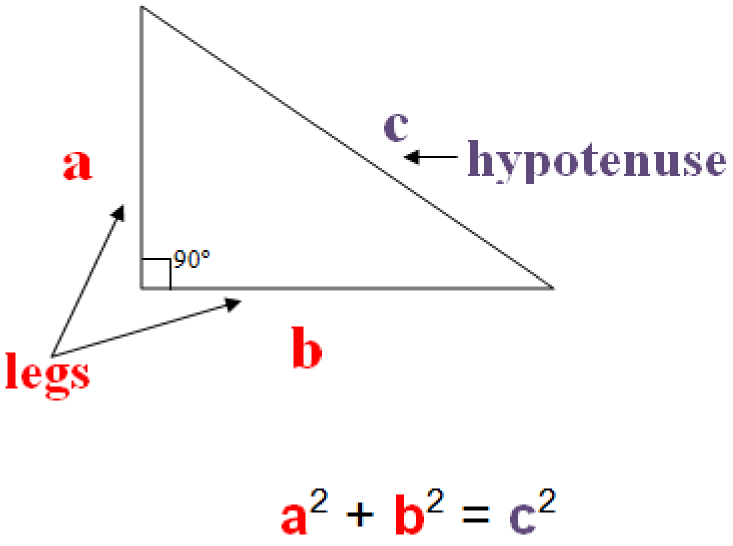
Since the % operator provides the remainder after edition, what do you imagine the result will be when this code is executed?

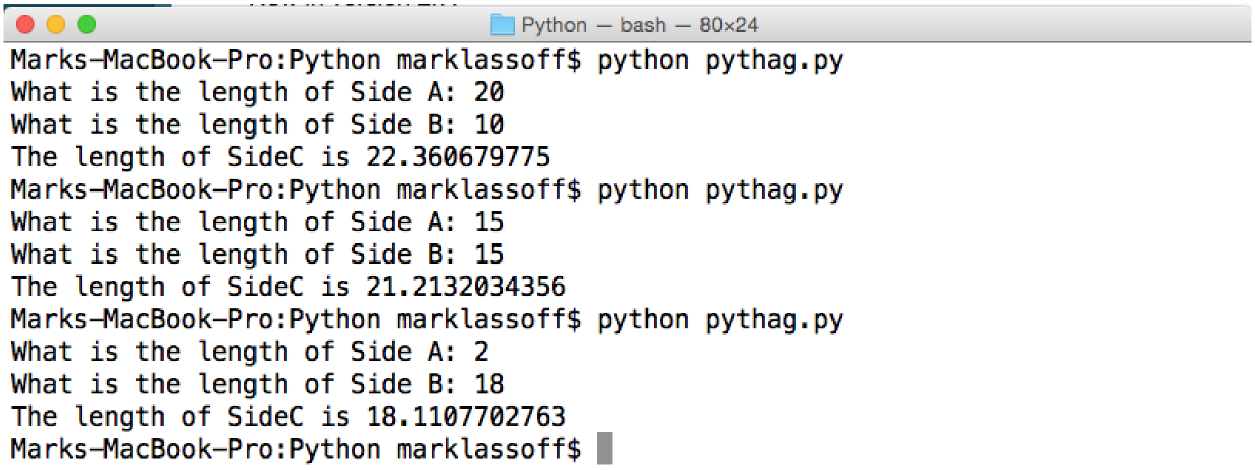
Part II

1) Create a new Python file in IDLE called “pythag.py.” Study the output below and imagine the code you’d have to write to produce this output. The program is using the Pythagorean theorem to calculate the length of the third side of a right angle triangle. You’ll recall that the Pythagorean theorem states the following:

$c^2 = a^2 + b^2$

The illustration below shows that if you have the length of side a and the length of side b, you can determine the length of side c. Adding a^2 (a * a) to b^2 (b * b) will yield c^2. If you then obtain the square root of side c, you’ll have its length.





```
Marks-MacBook-Pro:Python marklassoff$ python pythag.py
What is the length of Side A: 20
What is the length of Side B: 10
The length of SideC is 22.360679775
Marks-MacBook-Pro:Python marklassoff$ python pythag.py
What is the length of Side A: 15
What is the length of Side B: 15
The length of SideC is 21.2132034356
Marks-MacBook-Pro:Python marklassoff$ python pythag.py
What is the length of Side A: 2
What is the length of Side B: 18
The length of SideC is 18.1107702763
Marks-MacBook-Pro:Python marklassoff$ 
```

2) To get the square root of a number, you'll need Python's math.sqrt() function. The function will give you the square root of any number (or variable) you place inside the parenthesis. For example, math.sqrt(9) would produce 3.

   To use the math functions in Python, the first line of code must import Python's math library. On the first line of your new file type: **import math**.

3) Now write code using the input() function, the print command and format() function to calculate the length of sideC and produce output according to the screenshot above. Test your program when done and insure the values you are getting are correct. Debug as necessary.