Name: Shravan Sharath Shenoy ~~(as it would appear on official course roster)~~	
UCSB email address: shravansharathshenoy	**@ucsb.edu**
Lab Section: Monday 9:00 AM	
Optional: name you wish to be called if different from above	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	

h02: Python Variable Types and Operations

Assigned: Tuesday, April 9th, 2019
Due: Tuesday, April 16th, 2019
Points: 100

- You may collaborate on this homework with AT MOST one person, an optional "homework buddy". MAY ONLY BE TURNED IN THE LECTURE LISTED ABOVE AS THE DUE DATE. There is NO MAKEUP for missed assignments; in place of that, we drop the single lowest score (if you a zero, that is the lowest score.)
- IMPORTANT: When submitting this homework:
  - **DO NOT USE STAPLES**
  - WRITE YOUR **NAME ON EACH PAGE** IN THE SPACE PROVIDED
  - USE DARK INK PENS – PLEASE **DO NOT USE PENCIL**
  - PRINT THIS HOMEWORK **DOUBLE-SIDED** PLEASE!

READING ASSIGNMENT: Read Chapter 2.1 thru 2.5 in Perkovic, review your lecture slides/notes. Then complete these problems.

1. (20 pts) Section 2.1 describes how several operators and built in functions in Python work. What would be the result of entering the following at the Python interactive shell prompt (i.e. using IDLE)? (Note: You are encouraged to check your answers at the Python prompt before turning in your work, but try this on paper first, just by reading the text and trying to predict what will happen. Then try typing in the results at the Python prompt. Change your answers if they were mistaken, but even more important, try to figure out why you were incorrect.)
   Be very precise. Note that **True** is not the same in Python as **true**; upper vs. lower case matters. You will not get full credit for answers that are not precisely correct. 2 points for every entry.

Expression	Result	Expression	Result
2 + 3 * 5	17	True and False	False
4 < 3	False	True or False	True
19 % 3	1	7 // 2	3
1 + 2 == 3	True	9 // 2	4
5 ** 2	25	5 != 10//2	False

2. (5 pts) As described in section 2.1, a Python assignment statement contains the assignment operator, an expression and a variable, but not in that order. What is the correct order for these three parts, reading from left to right?

   variable, assignment operator, expression

Name: Shravan Sharath Shenoy
(as it would appear on official course roster)

3. (5 pts) Section 2.1 contains a list of thirty-three *reserved words* in Python that may not be used as the name of a variable. You don't need to memorize this list, but you do need to know where to find it, either in the book, or online. So, to be sure you can find it, list all of the Python reserved words that start with the letter c or f.

starts with "F" → False

Finally for from

Starts with "C" → class, continue

Starts with "f"

4. (20 pts) Assume the following variable settings and tell me the value of each of the following expressions in parts **a** thru **j**. If there is NO value because of a *syntax error*, please state so as well. 2 points for every part.

```
school = "UCSB"
course = "CMPSC8"
food = 'sushi'
qtr = "S19"
```

a. `school * 2`
'UCSBUCSB'

b. `qtr[0:3]`
'S19'

c. `course[1:3]`
'MP'

d. `'D' in school`
False

e. `'8' not in course`
False

f. `school > food`
False

g. `qtr < school`
True

h. `len(food) > 5`
False

i. `school[-1]`
'B'

j. `food[0]`
's'

5. (5 pts) Explain what, *if anything*, is wrong with this code?

```
num = 6
int = 8
print(int + num)
```

We should avoid using 'int' as a variable name because it is a python reserved keyword.

Name: Shravan Sharath Shenoy
(as it would appear on official course roster)

6. (5 pts) Assume that `courseNum` is an integer that represents the numeric part of a course number (e.g. 3, 8, 130, 16, 24). Write a Python expression that converts `courseNum` to a string (i.e. <class 'str'> in Python).

```
str(courseNum)
```

* In this case, assume a numerical value is in the same place as courseNum

7. (5 pts) Write the **full definition of a function** that takes 3 (and only 3) float type of arguments as inputs, x, y, and z, and returns their sum, their average, and the square-root of their sum of squares. When called, for example, as `myFunction(2.0, 3.0, 4.0)`, it should print out the following (it does not have to *return* anything!):

```
The sum is: 9.0
The average is: 3.0
The square-root of their sum of squares is: 5.38516480713
```

```
def SumAvgSqrt(x,y,z)
    a = (x+y+z)
    b = ((x+y+z)/3)
    c = ((x**2 + y**2 + z**2)**.5)
    print("The sum is: ", a)
    print("The average is: ", b)
    print("The square-root of their sum of squares is: ", c)
```

8. (5 pts) Write out, in the box on the right, what the following code prints out exactly?

```
L = [1, 2, 3, "skidoo"]
print(len(L))
print(L[2])
print(L[3]*2)
L[3] = 0
print(L)
L[0] = 3
print(L)
```

```
4
3
skidooskidooskidoo
[1, 2, 3, 0]
[3, 2, 3, 0]  # Final Result
```

Name: Shravan Sharath Shenoy
(as it would appear on official course roster)

9. (10 pts) As discussed in class and in Section 2.4, the **type()** function returns the type of a Python value. When you pass a variable such as **x**, **type(x)** returns the type of the value that the variable **x** currently refers to. Assume that the following assignment statement has been executed:

```
schools=["UCSB","Stanford","UCSD","Cal Poly"]
```

What will each of the expressions below evaluate to? As a reminder, strictly speaking, Python will print types in the format `<class 'int'>`, `<class 'float'>`, `<class 'str'>`, etc. so please use exactly that format for full credit. 2 points for every part.

a. `type(1+2.5)`

<class 'float'>

b. `type(2 * "3")`

<class 'str'>

c. `type((3,3))`

<class 'tuple'>

d. `type(schools)`

<class 'list'>

e. `type(schools[1])`

<class 'str'>

10. (10 pts) If you want to check whether x is greater than 10, and y is greater than 5, you can write the Python expression:

```
(x > 10) and (y > 5)
```

If x has the value 20, and y has the value 17, this evaluates to **True**.
But what if we accidentally wrote it as:

```
(x > 10) + (y > 5)
```

What would this expression evaluate to, assuming the same values for x and y?
(The answer requires you to read the sections in the textbook carefully, and then apply what you have learned. I suggest you try that first before trying it at the Python command line.)

The expression would evaluate to:
2