Team #:_____ Name:_____
Absent:

1. Python numeric classes (integer, long integer, and floating point) use the operations:

Operations: ** (exponentiation)
*, /, % (remainder)
+, -

Parenthesis can be used to override the default precedence. "Normal" mixed mode rules apply (i.e., at least one floating point operand is needed to generate a floating point result). For the following Python arithmatic expression predict the result:

Expression	Predicted Result	Actual Result
4 + 3 * 2		
7 + 8 / 2		
(7 + 8) / 2		
10 ** 20		
(7.0 + 8) / 2		

,./;;;../;'

2. A Python list is an ordered collection of comma-separated values of any data type enclosed in square brackets ('[', ']'). Operations on lists (**or any sequence collection**) include:

Operation	**Operator**	**Explanation**	**Example** myList=[5,6,7,8] ListB=[8,9]	**Result of Example**
Indexing	[<index>]	Access the element specified by the index	myList[2]	7
Slicing	[:]	Extract a part of the list	myList[1:3]	[6, 7]
Concatenation	+	Combine lists together	myList + ListB	[5, 6, 7, 8, 8, 9]
Repetition	*	Concatenate a repeatd number of times	ListB * 3	[8, 9, 8, 9, 8, 9]
Membership	in	Ask whether an item is in a list	3 in myList	False
Length	len(list)	How many items are in the list?	len(myList)	4

For the following lists, predict the results:

cheer = [2, 4, 6, 8, 'who', 'do', 'we', 'appreciate']
rhyme = [1, 2, 'buckle', 'your', 'shoe']

Expression	Predicted Result	Actual Result
cheer[4]		
cheer[2:6]		
rhyme[:4]		
cheer[1:4] + rhyme[3:]		
cheer[:2] * 3		
6 in cheer		
len(cheer)		
[cheer[2:4]*4]		

Team #:_____ Name:_____
Absent:

3. The *range* function is frequently used to generate a list of integers which are equally spaced. The syntax is: range([start,] stop [, step]). For example,

range(1, 100, 10) returns the list [1, 11, 21, 31, 41, 51, 61, 71, 81, 91], and
range(10) returns the list [0, 1, 2, 3, 4, 5, 6, 7, 8, 9].

For the following range calls, predict the resulting lists:

Expression	Predicted Result	Actual Result
range(5)		
range(0, 50, 10)		
range(100, 0, -10)		
range(2, 8)		
range(4, 4)		

4. Lists in Python are mutable, i.e., you can assign individual elements or slices new values.

For the following lists, predict the resulting lists:

Initial List Value	Expression	Predicted Result	Actual Result
temp = [0, 1, 2, 3, 4]	temp[1] = 99		
temp = [0, 1, 2, 3, 4]	temp[1] = 'cat'		
temp = [0, 1, 2, 3, 4]	temp[1] = ['cat', 'dog']		
temp = [0, 1, 2, 3, 4]	temp[1:3]=[6, 7, 8, 9]		
temp = [0, 1, 2, 3, 4]	temp[1,2] = 5		

5. In Python, the rules for identifiers (names) are (*letter* | "_") (*letter* | *digit* | "_"), plus:

- they are case sensitive
- they can be of any length (no excuse for nondescriptive names)

Variables are created when first used on the left-hand side of a assignment statement. Variables hold references to pieces of data and not the data itself. Variables can refer to any type of data.

a) Circle the validate variable names: foo_bar _local_time 3_test test_3_ r2d2_$now

b) What is the values of each of the variables after each code segment?

myList = range (5) otherList = myList otherList[0] = 99	a = 5 b = a b = 7	aL = 5L bL = aL bL = 7L