

1. What is the difference between the `/` and `//` operators? What are the types returned, and why would you use one over the other? Provide an example where the value would change depending on the operator used.

The `/` is classic division and will return the Data type that is inputted,  
 The `//` is true division and will always return an integer.  
 Examples  $15/6 = 2.5$   $15//6 = 2$  > rounds down to nearest integer

2. Consider the code below:

```
1 firstWord = "CSCI128"
2 secondWord = "Rocks!"
3 print(firstWord + secondWord)
```

- (a) What is printed in the code above?

**CSCI128Rocks!**

- (b) Provide two ways a space could be added between the printed words by modifying line 3.

You can add a space in the quotations of firstWord or secondWord. You can also add quotations to the print statement, `Print(firstWord + " " + secondWord)`

- (c) If secondWord was assigned the integer value 1 on line 2, what would be printed to the terminal? If this causes an error, how could we fix it?

It prints

**CSCI1281**

3. Complete the table below that lays out syntax and use rules for String and List de-referencing.

Rule	Correct Example	Incorrect Example
A single value or character from a list or string can be retrieved using square brackets around an index number	<code>my_list[3]</code>	<code>my-list(a)</code>
To access a list/string, your index number must be an integer.	<code>my_list[1]</code>	<code>my_list[1.2]</code>
Indices used must correspond to a valid location in the list or string	<code>my_list = [1,2,3]</code> <code>my_list = [1]</code>	<code>my_list = [1,2,3]</code> <code>my_list[6]</code>
A list/string must have square brackets around the data in the list.	<code>my_list = [1,2,3]</code> <code>my_list[1]</code>	<code>my_num = 123</code> <code>my_num[1] # error</code>

4. Analyze the string stored in variable x below. For each of the questions about indices and slicing, what value is accessed?

x = "Hello World"  
 0 1 2 3 4 5 6 7 8 9 10

(a) x[0] H

(b) x[6] W

(c) x[0:6] Hello\_ space

(d) x[-2] l

(e) x[0::2] Hllowrd

(f) x[-5::3] w|

- (g) Provide the required slicing indices to access the substring "HlWl"

`Print(x[0:10:3])`

- (h) Provide the required slicing indices to reverse the entire string: "dlroW olleH"

`Print(x[::-1])`

5. What is the difference between immutable and mutable variables? Give an example of each.

Mutable; Can be modified after being created

↓  
List

Immutable; Cannot be modified after creation

↓  
integer

6. The code below does not do what the programmer expects. What is wrong with their code? How would they fix it?

```
1 listOfInts = [1,2,3,4] # Create list with numbers 1-4
2 listOfInts[1] = 3//2   # Update the 1st number of list to be 1.5
3 listOfInts[4] = 1 ** 2 # Update the 4th number of list to be 2
4 print(listOfInts)      # Print the list
```

This code doesn't work because line 3 goes out of range with [4]. This can be fixed by properly accessing the 4<sup>th</sup> number with [3].

7. Two armies of  $n$  ants walk towards each other on a horizontal pole of length  $L$  cm at a speed of 1 cm/s. When two ants would collide, they immediately reverse direction; that is, turn back and walk in the opposite direction (taking 0 seconds to do so). When an ant reaches the end of the pole, it falls off. Given  $n$ ,  $L$ , and the starting position of each ant, at what time will the **first** and **last** ants fall off the pole?
- (a) This problem appears to be very challenging, but there is one key idea that greatly simplifies it. Use problem solving strategies such as considering simple examples, working backwards from the solution, etc to determine this key idea, and write 1-2 sentences describing it.

- (b) Now that you know the key idea, how can you more easily determine when the first and last ants will fall off the pole?

- (c) We project the pole onto the x-axis and consider the center of the pole to be at  $x = 0$  and give the locations of the ants in terms of this. Write pseudocode for a program that takes in the length of the pole and the positions of the ants and outputs the time at which the first and last ants fall off the pole. The first line of input provided to your program is the length of the pole  $L$ . The second line of input is the locations of the ants, from left to right, separated by spaces (hint: `.split()` could be useful here). Assume ants in a negative position are initially moving right, while ants in a positive position are moving left.

An example input could be:

```
6.2
-1.9 -0.5 1 2.5
```

with an output of:

```
first: 3.6s
last: 5.6s
```

After the studio for this week opens, submit your working program to the “Ants” studio problem for testing and credit.

- (b) Now that you know the key idea, how can you more easily determine when the first and last ants will fall off the pole?

- (c) We project the pole onto the x-axis and consider the center of the pole to be at  $x = 0$  and give the locations of the ants in terms of this. Write pseudocode for a program that takes in the length of the pole and the positions of the ants and outputs the time at which the first and last ants fall off the pole. The first line of input provided to your program is the length of the pole  $L$ . The second line of input is the locations of the ants, from left to right, separated by spaces (hint: `.split()` could be useful here). Assume ants in a negative position are initially moving right, while ants in a positive position are moving left.

An example input could be:

```
6.2
-1.9 -0.5 1 2.5
```

with an output of:

```
first: 3.6s
last: 5.6s
```

After the studio for this week opens, submit your working program to the “Ants” studio problem for testing and credit.