

Name: \_\_\_\_\_ Section: 1 2 3 4

1 = Mutchler, 1<sup>st</sup>-2<sup>nd</sup> periods. 2 = Mutchler, 3<sup>rd</sup>-4<sup>th</sup> periods. 3 = Anderson, 7<sup>th</sup>-8<sup>th</sup> periods. 4 = Anderson, 9<sup>th</sup>-10<sup>th</sup> periods.

Use this quiz to help make sure you understand the videos/reading. **Answer all questions.** Make additional notes as desired. **Not sure of an answer?** Ask your instructor to explain in class and revise as needed then. **Please print two-sided.**

Throughout, where you are asked to “circle your choice”, you can underline or circle it (whichever you prefer).

Video: **Sequences** [8:43 minutes]

1. Sequences are powerful because:

2. What gets printed by the program shown below when *main* runs?

```
def main():
    list1 = [74, 34, 13, 30, 4004]
    string1 = 'Expert texpert!'

    examples_of_indexing(list1)
    examples_of_indexing(string1)

def examples_of_indexing(sequence):
    length = len(sequence)
    print('Length is', length)

    print('At indices 0, 1, 4:')
    print(sequence[0])
    print(sequence[1])
    print(sequence[4])

    print('List them all:')
    for k in range(len(sequence)):
        print(sequence[k])
    print()

    print('Cool stuff:')
    print(sequence[-1])

    print('Last item:')
    length = len(sequence)
    print(sequence[length - 1])
```

Just take a wild guess and ask in class!

**Output**  
(fill in the blanks)

Length is \_\_\_\_\_

At indices 0, 1, 4:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

List them all:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Cool stuff:

\_\_\_\_\_

Last item:

**(output continues)**

Length is \_\_\_\_\_

At indices 0, 1, 4:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

List them all:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

... [don't bother doing all of the letters, just skip to the last one]

\_\_\_\_\_

Cool stuff:

\_\_\_\_\_

Last item:

3. Implement (here, on paper) the following function, per its specification.

```
def count_negative(seq):  
    """  
    Returns the number of items in the given sequence of numbers  
    that are negative.  
  
    Precondition: The argument is a sequence of numbers.  
    """
```

4. Implement (here, on paper) the following function, per its specification.

```
def count_short_ones(seq_of_lists):  
    """  
    The argument is a sequence of lists. Returns the number of  
    items in that sequence whose length is less than 3.  
  
    For example, if the argument is:  
        [ [3, 5], [3, 9, 0, 4], [5], [5], [], [9, 8, 7], [5, 6] ]  
    then this function returns 5, since 5 of the 7 lists in the  
    above sequence have length less than 3  
    (I have displayed those 5 lists in red.)  
  
    Precondition: The argument is a sequence of lists.  
    """
```

Video: *The Last Item in a Sequence* [3:20 minutes]

5. Given a sequence called *seq* that contains 10 items, write expressions for:
  - The beginning item (element) in *seq*: \_\_\_\_\_
  - The last item (element) in *seq*: \_\_\_\_\_
6. Suppose that you have a *list* called *my\_list* that contains 10 items. What error message appears if you attempt to access *my\_list[10]*?
7. Suppose that you have a *string* called *my\_string* that contains 10 characters. What error message appears if you attempt to access *my\_string[10]*?
8. In Eclipse, when a run-time *exception* (error) occurs, how do you go straight to the line at which the exception occurred?
9. Is that line always the line that is wrong? **Yes** **No** (circle your choice)
10. What should you do if the exception occurred in the *zellegraphics* or *new\_create* modules?
11. In the two code fragments below, assume that the variable *s* has been assigned as its value some sequence. The code on the left (below) causes a run-time exception. The code on the right (below) does NOT cause a run-time exception. But both use the same expression **len(s)**. Explain why the code on the left breaks (and is wrong) but the code on the right does not break (and is correct).

```
... s[len(s)] ...
```

```
for k in range(len(s)):  
    ... s[k] ...
```

