**MCS 260 – Introduction to Computer Science – Fall 2020 – Emily Dumas**

# Week 4 Worksheet

**Instructions.** Unlike some previous worksheets, for this one you are encouraged to have a Python interpreter open at all times (or to have a code editor open, and to be ready to run your code).

**Language features.** In your solutions you can use (and may need) the following built-in functions/methods, even if they haven't been discussed in lecture:

- The function `str()` will convert a numeric type to a string, so for example, `str(123)` evaluates to `"123"`.

**Problems.**

(1) Suppose that `L` is a list whose elements are sequences. Generate a new list `M` where `M[i]` is equal to the length of `L[i]` (an integer). Thus if `L = [ [5,6], "Fuji" ]` then the result should be `M = [2,4]`; but your code should work for any `L` whose elements are sequences, not just in this one example.
   (a) Use a for loop to do this.
   (b) Use a list comprehension.

(2) Write a function `with_spaces()` that takes one parameter, an iterable of strings, and returns a list of the strings in this iterable that contain a space character. So, for example, if we set
```
L = [ "banana", "apple", "green pear", "guava", "red dragonfruit" ]
```
then `with_spaces(L)` should evaluate to
```
[ "green pear", red dragonfruit" ]
```

(3) In each part of this problem, write code that uses the following list of tuples:
```
coursedata = [ ("MCS",260,"Intro. to comp. sci."),
               ("MCS",275,"Prog. tools and file mgmt."),
               ("MATH",180,"Calculus I"),
               ("MATH",320,"Linear Algebra I"),
               ("MATH",549,"Differentiable Manifolds I"), ]
```
Thus you probably want to copy and paste this into a source file, or download it from
https://dumas.io/teaching/2020/fall/mcs260/samplecode/coursedata.py

   (a) Write a for loop that iterates over this list and prints all of the course numbers, i.e.
```
MCS 260
MCS 275
MATH 180
MATH 320
MATH 549
```
   (b) Write a list comprehension that iterates over `coursedata` and yields
```
[ "MCS 260", "MCS 275", "MATH 180", "MATH 320", "MATH 549" ]
```
   (c) Write a for loop that prints the data for each MCS course in the following format:
```
Course Number: MCS 260
Description: Intro. to comp. sci.
```
   (d) Write a list comprehension that is analogous to part (a), but yields a list of strings, one for each MCS course. For example the first string would be
```
"Course Number: MCS 260\nDescription: Intro. to comp. sci.\n"
```

(4) Using the list of lists below, write a for loop inside of a for loop that will print the even numbers that occur as elements of elements of `L`.

```
L = [ [3,1,2], [9,9,6], [3,0,4,1] ]
```

That is, the output should be:

```
2
6
0
4
```

(5) Write a function `opening(...)` to generate the first line of a letter or memo. It should take parameters `fullname`, `salutation`, and `greeting` (in that order). The parameter `salutation` should have default value `""`, and the parameter `greeting` should have default value `"Dear"`. This function should print the greeting, followed by a space, an optional salutation, the fullname, and a comma. Examples:

- `opening("Grace Hopper")` prints
  ```
  Dear Grace Hopper,
  ```
- `opening("Emily Dumas",greeting="Howdy")` prints
  ```
  Howdy Emily Dumas,
  ```
- `opening("Marie Curie",salutation="Dr.",greeting="To the esteemed")` prints
  ```
  To the esteemed Dr. Marie Curie,
  ```

Hint: How do you avoid printing two spaces between the greeting and the fullname if the salutation is the empty string?

(6) Write a function that takes a list of floats and returns their average (mean). (As a reminder, the mean of real numbers $x_1, x_2, \ldots, x_n$ is defined as $\frac{1}{n}(x_1 + x_2 + \cdots + x_n)$.)

(7) Suppose that `cmds` is an iterable of strings. Write a loop that will print the elements of `cmds` in order, stopping with the first one that is equal to `"stop"`, `"exit"`, or `"end"`.

(8) Write a function that applies an arbitrary linear function $f(x) = mx + b$ to a number $x$. It should take three parameters $m, b, x$. Then, use this to apply the function $3x+2$ to the integers $[0, 1, \ldots, 24]$ and the function $x - 7$ to the floats $[0, 0.5, 1, 1.5, 2, 2.5, 3]$.

**Extensions.**

- If `R` is a sequence in Python, the expression
  ```
  R[:5]
  ```
  does not necessarily evaulate to sequence of length 5. It can produce a sequence of a different length, without producing an error! How?
- These two sections of code do very similar things:
  ```
  y = []
  for x in range(5):
      y.append(x**3)
  ```
  and
  ```
  y = [ x**3 for x in range(5) ]
  ```
  But is there any identifiable difference between them? Specifically, if you know one of these was previously run in a REPL that is still open, is there any way (other than examining the previously run commands) you could tell which one it was?
- Write a list comprehension that produces all integers between 10000 and 20000 that are divisible by 23, have 5 as the last decimal digit, and also contain the digit 7.

**Revision history:**

- 2020-09-15 Modified wording in question 5
- 2020-09-13 Initial release