PIC 16, Winter 2018 – Preparation 2W

Assigned 1/12/2018. To be completed before class 1/17/2018.

Intended Learning Outcomes

By the end of this preparatory assignment, students should be able to:

- choose the most appropriate sequence data type (list, set, tuple, dictionary, etc…) based on the specific needs of the problem.
- conveniently interact with the contents of a list using list methods,
- define a list or dictionary compactly using list/dict comprehension, and
- remove elements from a list using `del`.

Tasks

- ☐ Read 5.1 (not 5.1.x) and 5.2
- ☐ Predict and check the output of the following:
  ```
  x = [1,2,3]; print x
  x.append(4); print x
  x.extend([6, 7, 8]); print x
  x.insert(2,5); print x
  x.remove(3); print x
  a = x.pop(); print a; print x;
  b = x.index(6); print b
  c = x.count(1); print c
  x.reverse(); print x
  x.sort(); print x
  del x[1:-1]; print x
  y = x; y.append(9); print y; print x
  y += [10, 11, 12]; print y; print x
  y  = y + [13, 14, 15]; print y; print x
  ```
- ☐ Read 5.3 and http://stackoverflow.com/questions/1708510/python-list-vs-tuple-when-to-use-each
- ☐ Given the following:
  ```
  months_tuple = ('January', 'February', 'March', 'April', 'May', 'June',
  'July', 'August', 'September', 'October', 'November', 'December')
  ```
  Write a lambda function `num_to_month` that converts an integer into the name of the month of the year, e.g. `num_to_month` (2) returns `February` (not `March`). This isn't the most appropriate use of a tuple, considering the first post on stackoverflow, although perhaps this function would be the slightest bit faster using the tuple instead of a list.
- ☐ Read 5.4. Don't worry about "set comprehension". (You can read that part again after 5.1.4.)
- ☐ Determine the number of unique letters in "supercalifragilisticexpialidocious" using a set.
- ☐ Read 5.5. This is like a map data structure in Java or C++. Hold off on "dict comprehension".
- ☐ Manually modify `months_tuple` from before to create a dictionary `months_dict` that can be used to convert month name to number, like `months_dict['February']` returns 2
- ☐ Read 5.1.4, "set comprehension" from 5.4, and "dict comprehension" from 5.5
- ☐ Watch https://youtu.be/DHEXfdXvtt0 for an example of list comprehension.
- ☐ Recreate `months_dict` from `months_tuple` using dict comprehension (instead of manually).