CS 146 Fall 2025 – Quiz 3 "Cheat Sheet"

Numeric Operators

+, -, /, *, **: Addition, subtraction, division, multiplication, exponentiation
//: Floor division: Round division result down to nearest whole number
%: Modulus: Evaluate to remainder of division

Precedence (equal precedence evaluated left-to-right):

parentheses > ** > negate > *,/,//,% > +,-

Indexing Operator

seq[idx]: Item of `seq` at index `idx`
seq[start:stop(:step)]: Copy of subsequence of `seq` from inclusive `start` to exclusive `stop` by `step`

Range

range(stop): Equivalent to range(0, stop, 1)
range(start, stop[, step]): Create sequence of integers from inclusive **start** to exclusive **stop** by **step**

Strings

- The following functions are built-in
  len(string): Returns the number of characters in the string
  int(string), float(string): Converts numeric string to int or float
  str(object): Converts object, e.g. int or float to a string
  sorted(string): Returns the characters of the string as a list in sorted order
- String object methods
  upper(), lower(), capitalize(), swapcase(): Returns a new upper or lower-cased, or 1st letter upper-cased, or case reversed (lower ↔ upper) string
  index(item): Returns the index of the first occurrence of `item` in the string or error otherwise
  find(some_string): Returns the first index that **some_string** occurs at in the string or -1 if not found
  find(some_string, index): Same as above, but starts searching at index
  replace(old, new): Return a copy of the string with all occurrences of old substituted with new
  startswith(prefix): Returns **True** if the string starts with prefix, **False** otherwise
  endswith(suffix): Returns **True** if the string ends with suffix, **False** otherwise
  strip(): Returns a copy of the string with only the leading and trailing whitespace removed
  split(): Return a list of the words in the string using whitespace as the delimiter
- String operators
  string1 + string2: Returns a new string that is the concatenation of string1 and string2
  string * int: Returns a new string that is string repeated int times
  substr in string: Returns **True** if substr is a substring of string, **False** otherwise

Lists

- Creating new lists
  [] creates empty list
  [object1, object2, ...] creates list containing objects
  list(iterable) creates a list from any iterable object (e.g., range, set, string)
- The following functions are built-in and answer questions about lists
  len(list): Returns the number of elements in `list`
  sum(list), min(list), max(list): Returns the sum, min, or max of elements in `list`
  sorted(list): Returns a new copy of the list in sorted order
- List object methods
  append(x): Adds x to the end of the list

`extend(other_list):` Adds all elements of `other_list` the end of the list
`index(item):` Returns the index of the first occurrence of `item` in the list or error otherwise
`insert(index, x):` Insert `x` at `index` in the list
`pop():` Removes the item at the end of the list and returns it
`pop(index):` Removes item at `index` from the list and returns it
`reverse():` Reverses the elements in the list
`sort():` sorts the elements in the list in place

- List operators
  `list1 + list2:` Returns a new list that contains the elements of `list1` followed by the elements of `list2`
  `list * int:` Returns a new list that contains the items in `list` repeated `int` times
  `item in list:` Returns True if `item` is an element of `list`, False otherwise

Modules

- **`random`** module
  `randint(a, b):` Return a random integer `N` such that `a ≤ N ≤ b`
  `uniform(a, b):` Return a random floating point number `N` such that `a ≤ N ≤ b`
- **`math`** module
  `sqrt(num):` Return the square root of `num`
  `pow(x, y):` Return `x` to the power of `y` (i.e. `x ** y`)