**Exam 2: Review Lecture**

*(The list is not complete)*

**Sequences: String, List, Tuple**

1. Strings: have seven operations
2. Indexing

Var[idx], idx = 0, 1, 2, 3, .. or … -3, -2, -1

IndexError: if idx does not exist

1. Slicing: var[start:end], colon part is necessary,
   1. does not through IndexError
   2. creates a brand new object (regardless mutable or immutable)
   3. creates same data structure type (list of lists – list of lists)
2. for loop

for item in sequence: when sequence is a string then item is a char

1. len:

for idx in range(len(sequence)):

You can get value from idx, but converse is not always possible. For example, sequence contains duplicate items

1. in operator:

substring match

list or item exists

1. + operator: concatenation

“Hello” + “ World” = “Hello World”

[1, 2, 3] + [4, 5, 6] = [1, 2, 3, 4, 5, 6]

Creates a brand new object instance

Write on your cheat-sheet: when a new object is created-copying lecture

1. \* some integer

“Hello”\*2 = “HelloHello”

[1, 2, 3]\*2 = [1, 2, 3, 1, 2, 3]

1. Comparison

Lower case vs upper case

Digits: “11” < “2”

Prefixes: “bat” < “batman”

1. Methods (produce new object)

upper/lower

startswith/endswith

strip/lstrip/rstrip

Replace

find → -1 (if not found)

format → replaces { }

split—> list of strings

join→ “,”.join(L)

Invocation of methods: some\_var.method():

For example, “hello”.upper() → “HELLO”

str.upper(some\_var)

B. List/csv

1. L = [1, 2, 3], emptyLIST [ ] or list()
2. Methods
   1. append()
   2. extend()
   3. pop-last index
   4. index: IndextError (different from find in string)
   5. sort
      1. increasing (default)
      2. Reverse (Ture→ decreasing)
      3. Key (function object reference)
   6. sorted function (creates a brand new object instance)
3. Process\_csv: list of lists

C. Dictionary

1. immutable value as key
2. value can be anything
3. {}, dict()
4. Method:

Pop

keys-> a kind of list

values->a kind of list

items() creates a list of tuples

D. json: useful for the data structures other than list of lists

know the difference between json and python

E. Objects

assignment operation & argument passing => reference copy

stack (reference) vs Heap (object)

F. namedtuple (custom -> immutable)

know how constructor function to create named tuple

G. Copying:

Import copy

Shallow (copy.copy→ depth level 1)

Deep (copy.deepcopy→ all depth levels only for immutable objects)

H. Recursion:

Base case

Recursive case

RecursionError (Infinite recursion or wrong arguments)

I: Function Object references

Sorting [(fname, lname), ]

def extract(item)

item[1]

lambda item: item[1]:

J: Compressions

[ <expression> <for> <if>]

[ <expression1> if condition else <expression2> <for> ]

[ key: value <for> <if>]

K: Error Handling

-assert

-try except

-raise

L: File and Directories

f= open()

Read:

1. f.read()
2. list(f)
3. for line in f

Write: “w”

f.write(“\n”)

f..close

M. os module

os.mkdir

os.listdir

os.path.join

os.path.exists

os.path.isfile

os.path.dir