Review of Comp Sci: Cristafalk
1) Framework w/in which c.s. algorithms + Data structures fit   3/15/17
2) Python programming language   A2

WHAT IS COMPUTER SCIENCE?
*= problem solving

▷ Study of problems, prob solving, solutions, & *p.s. process

▷ Goal: Develop an algorithm - step by step list of instruction for solving a problem

▷ ! Solely computer work (computers = tools for p.s.)

▷ Study of computation (problems that == & != computable) study of existing/non-existent algorithms)

Drive car, but don't know how engine works (nor do you NEED to) →

* Abstraction - seperates logical & physical perspectives

* Interface - Functions provided by maker for client to understand/use for purpose despite abstraction

▷ procedural abstraction - like import math → math.sqrt()
  - don't know calculation details, but know def/usage

WHAT IS PROGRAMMING?

▷ programming - process of taking an algorithm & encoding it into notation (programming lang) so it can be executed by a computer

Control Structures -
- iteration (while, for)
- selection (if, else, elif)
all can be nested

▷ Languages provide control constructs/data-types to represent algorithm & manipulate data

▷ Comps store data - strings of binary digits

▷ Data types - provide strings w/ meaning

  ▷ low-level built in data types primitive D.T.

WHY STUDY DATA STRUCTURES/ABSTRACT D.T.?

List Comprehension
for x in range (1,10)
[nothing to x w. append]
w = [? for x in range]
ALSO CONT. STR.
w = [x+1 for x in [] if x%2==0]
what data reps
& how it is eventually constructed.

▷ Abstraction (better def) - a process that hides details of a particular function to allow user to view it at a very high level

▷ Data Abstraction: Implementation-independent

pwr/ing level of abstraction to encapsulate data

  ▷ Abstract Data Type (ADT) - logical description of how we view data + operations that are allowed w/o regard to how they will be implemented

  ▷ encapsulation - hiding details from users view, called information hiding

  ▷ Data structure - implementation of a ADT
    - phys view of data w/ programming constructs & primitive D.T.