

Bronx Community College
Department of Mathematics and
Computer Science
Fall 2018 CSI31
Review for Final Exam

This review lists only the topics covered in class after the midterm exam. There will be topics on the exam from the first part of the semester. Use the midterm exam review for those topics.

Chapter 6: Defining Functions

What is a function?

Reasons for using functions:
modularity - make program more understandable,
code reuse - reduce duplication of code

Function call, function definition

function parameter, formal parameter, actual parameter, argument

mechanism of function call and parameter passing, mutable parameters

variable scope
return value

function changing its parameters

define new functions in Python code

Chapter 7: Decision Structures

understand conditional execution

simple decision pattern: if statement

two-way decision pattern: if-else statement

multiway decision pattern, if-elif-else statement

write programs using these structures

max of three problem and some approaches

Boolean expressions, relational operators

Chapter 8: Loop Structures and Booleans

while loop: body is executed as long as loop condition remains true

definite loops using for
indefinite loops using while

interactive loop pattern, sentinel loop pattern, end-of-file loop pattern, nested loops

Boolean algebra, Boolean operators and or, not

Chapter 9: Simulation and design

computer simulation, Monte Carlo methods

pseudo-random numbers, python random module, functions random, randrange

top-down design
bottom-up implementation, unit testing

Chapter 11: Data Collections 11.1, 11.2, 11.7

python lists and their functions/methods and operations:

operations on sequences, including lists:
+ : concatenation, * int : repetition,
[] : indexing, len() : length, [:]: slicing

for <var> in <seq> : iteration,
<expr> in <seq> : Boolean membership
check
List methods:
<list>.append(x): attach the item x to the
list in last position, changes the list
<list>.sort(): sort the list in increasing
order, changes the list
<list>.reverse(): reverse the list, changes
the list
<list>.index(x): if x is in the list, returns the
index of the first, location
<list>.insert(i, x): inserts s into the list at
index i, changes the list
<list>.count(x): returns the count of the
number of times x appears in the list
<list>.remove(x): removes the item x from
the list, changes the list
<list>.pop(i): removes the item at index i,
returns the item, changes the list

Lists are mutable, dynamic,
heterogeneous

write programs using lists
statistics functions using lists:
mean, median, standard
deviation

python dictionary is a built-in Python data
type for storing key-value ordered pairs.
Also called mapping or function table

dictionary methods: see slides for 12/3
create a dictionary: move= {'bird':'flies',
'fish':'swims', 'human':'walks'}
find the value associated with a key":
move['fish']
change the value associated with a key:
move['human'] = "takes a taxi"
add a new pair: move['dog'] = 'walks'
test whether a key is in the dictionary:
'human' in move, returns True or False
Return the keys as list: move.keys()
Return the values as list: move.values()

Return the key-value pairs as set of tuples:
move.items()
Delete pair with key: del move['human']
Delete all: move.clear()

write simple programs using dictionaries