**15110 PYTHON REFERENCE SHEET**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Arithmetic Operations: | \*\* | \* | / | // | % | + | - |
| Relational Operations: | == | != | < | <= | > | >= |  |
| Logical Operations: | and | or | not |  |  |  |  |

Variable Names: All variable names must start with a letter (lowercase recommended). The remainder of the variable name (if any) can consist of any combination of uppercase letters, lowercase letters, digits and underscores (\_). Variables are case sensitive.

Assignment Statement: *variable* = *expression*

Defining a function: def *functionname*( *parameterlist* ):

*function\_body*

A *parameterlist* may be empty or may include one or more variables representing data required for the

function, separated by commas.

Calling a function: *functionname* ( *argumentlist* )

An *argumentlist* may be empty or may include one or more expressions representing data required for the function to use, separated by commas.

Importing module: import *modulename*

Using module: *modulename*.*functionname*( *argumentlist* )

print(data) prints data to screen and moves cursor to next line print(data, end=" ") prints data to screen and keeps cursor on same line print() moves cursor to next line

return(data) returns data to instruction that called this function

for *v* in range(*x, y, z*): loops for *v* = *x* through *y*-1, inclusive in steps of *z loop\_body* (*y* is optional, default 0. *z* is optional, default 1.)

while *condition*: loops while *condition* is True

*loop\_body*

if *condition1*: executes *instruction1* set once if *condition1* is True

*instruction1\_set*

elif *condition2*: otherwise executes *instruction2* set once if *condition2 instruction2\_set* is True. This part is optional, can be repeated.

else: otherwise executes *instruction3* set once if all

*instruction3\_set* previous conditions tested as False. Optional.

Lists: *listname* = [] An empty list.

*listname* = [ *item0* , *item1* , … , *itemn-1*] A list of n items, n >= 1.

*listname*[*i*] Evaluates to the ith element of the list

len(*listname*) returns the number of items in the list

*item* in *listname* returns True if the item is in the list, False otherwise.

*listname*[*i*:*j*] returns a sublist of list from index *i* to *j*-1 *listname* =[ *item*] \* *n* creates a list with *n* copies of the item *listname*.append(*item*) appends item to end of the list *listname*.remove(*item*) removes the first occurrence of the item in the list

*listname*.insert(*pos, item*) insert the *item* to the item in so that its index will be equal to *pos*

for *item* in *listname*: performs instructions once for each item in list, no index is available

*loop\_body* (*item* can be referenced in loop body)