# Department of Computing

# Class: SE-7B

# Lab 4: Strings

# Date: October 19, 2016

# Time: 2pm-5pm

# Instructor: Dr. Muddasir

# Lab 4: Strings and functions

**Tools/Software Requirement**

Python

#### Strings

Like Java, Python has a built in string type. The + operator is overloaded to do string concatenation on string values.

>>> 'artificial' + "intelligence"   
'artificialintelligence'

There are many built-in methods which allow you to manipulate strings.

>>> 'artificial'.upper()  
'ARTIFICIAL'  
>>> 'HELP'.lower()  
'help'  
>>> len('Help')  
4

Notice that we can use either single quotes ' ' or double quotes " " to surround string. This allows for easy nesting of strings.

We can also store expressions into variables.

>>> s = 'hello world'   
>>> print s   
hello world   
>>> s.upper()  
'HELLO WORLD'  
>>> len(s.upper())  
11  
>>> num = 8.0   
>>> num += 2.5   
>>> print num   
10.5

In Python, you do not have declare variables before you assign to them.

#### Exercise: Dir and Help

Learn about the methods Python provides for strings. To see what methods Python provides for a datatype, use the dir and help commands:

>>> s = 'abc'   
  
>>> dir(s)  
['\_\_add\_\_', '\_\_class\_\_', '\_\_contains\_\_', '\_\_delattr\_\_', '\_\_doc\_\_', '\_\_eq\_\_', '\_\_ge\_\_', '\_\_getattribute\_\_', '\_\_getitem\_\_', '\_\_getnewargs\_\_', '\_\_getslice\_\_', '\_\_gt\_\_', '\_\_hash\_\_', '\_\_init\_\_','\_\_le\_\_', '\_\_len\_\_', '\_\_lt\_\_', '\_\_mod\_\_', '\_\_mul\_\_', '\_\_ne\_\_', '\_\_new\_\_', '\_\_reduce\_\_', '\_\_reduce\_ex\_\_','\_\_repr\_\_', '\_\_rmod\_\_', '\_\_rmul\_\_', '\_\_setattr\_\_', '\_\_str\_\_', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'replace', 'rfind','rindex', 'rjust', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']  
  
>>> help(s.find)

Help on built-in function find:

find(...)

S.find(sub [,start [,end]]) -> int

Return the lowest index in S where substring sub is found,

such that sub is contained within s[start,end]. Optional

arguments start and end are interpreted as in slice notation.

Return -1 on failure.

>> s.find('b')  
1

Try out some of the string functions listed in dir (ignore those with underscores '\_' around the method name).

#### Lab Tasks:

Note: Use a main function to call the required functions. Take input from user if not mentioned in task.

1. Take two inputs from user and print them in same line alternatively thrice.
2. Write a program which outputs the length of any input string.
3. Write a program in which user will enter 2 numbers(First number will be the starting index from the string and second will be the ending index). That portion of statement should be uppercased in output and other statement will be omitted.

**For example**: ‘enter the statement’

User enters 0 and 4.

Output: ‘ENTER’

1. Write five different statements and number them. Ask user to press number from 1-5 and print that statement.

Hint: Use if statements. Don’t use elif.