

# CIS 6 :: Lab 4 - Strings and Lists

Student Name: **Naveed Yeganegi**

Semester: **Spring 2021**

## Task 1: Definitions & Concepts - Homework

Instructions: Answer the questions below.

1. What is a string data type?  
=> **A sequence of characters enclosed in quotations or apostrophes**
2. What type of data type the string is, primitive or class?:  
=> **Primitive**
3. What is ASCII character code?:  
=> **American Standard Code, an official encoding for most numbers and characters**
4. What is the difference between ASCII and Unicode?  
=> **Unicode is much bigger, supporting almost every character in every language. ASCII is a subset of Unicode.**
5. List two string methods and explain their use:  
=> **s.split() will split s into a list of strings. s.join(list) will combine a list of strings into one with s as the spacer**
6. What are some of the difference between strings and lists?:  
=> **Lists can be composed of multiple data types, and unlike strings lists are mutable.**
7. What are some of the similarities between strings and lists?:  
=> **They are both sequences of characters, and function with similar operators such as + and []**

## Task 2: Internet Research - Homework

Explain why public key encryption is more useful for securing communications on the Internet than private (shared) key encryption.

=> **Public key systems allow users to encrypt and upload communications while keeping the key for decryption separate.**

## Task 3: Understanding Strings

1. Given the initial statements:

s1 = "spam"

s2 = "ni!"

Show the result of evaluating each of the following string expressions.

(a) "The Knights who say, " + s2 => **The Knights who say, ni!**

(b) `3 * s1 + 2 * s2` => **spamspamspamni!ni!**

(c) `s1[1]` => **p**

(d) `s1[1:3]` => **pa**

(e) `s1[2] + s2[:2]` => **ani**

(f) `s1 + s2[-1]` => **spam!**

(g) `s1.upper()` => **SPAM**

(h) `s2.upper().ljust(4) * 3` => **NI! NI! NI!**

2. Given the same initial statements as in the previous problem, show a Python expression that could construct each of the following results by performing string operations on `s1` and `s2`.

(a) `"NI"` => **`s2[:2].upper()`**

(b) `"ni!spamni!"` => **`s2 + s1 + s2`**

(c) `"Spam Ni! Spam Ni! Spam Ni!"` => **`(s1.capitalize() + " " + s2.capitalize()) * 3`**

(d) `"spam"` => **`s1.lower()`**

(e) `["sp", "m"]` => **`s1.split("a")`**

(f) `"spm"` => **`s1.replace('a', '')`**

3. Show the string that would result from each of the following string formatting operations. If the operation is not legal, explain why.

(a) `"Looks like {1} and {0} for breakfast".format("eggs", "spam")` => **Looks like spam and eggs for breakfast**

(b) `"There is {0} {1} {2} {3}".format(1, "spam", 4, "you")` => **There is 1 spam 4 you**

(c) `"Hello {0}".format("Susan", "Computewell")` => **Hello Susan**

(d) `"{0:0.2f} {0:0.2f}".format(2.3, 2.3468)` => **2.30 2.30**

(e) `"{7.5f} {7.5f}".format(2.3, 2.3468)` => **Error, 7 is more spaces than is available for width**

(f) `"Time left {0:02}:{1:05.2f}".format(1, 37.374)` => **Time left 01:37.37**

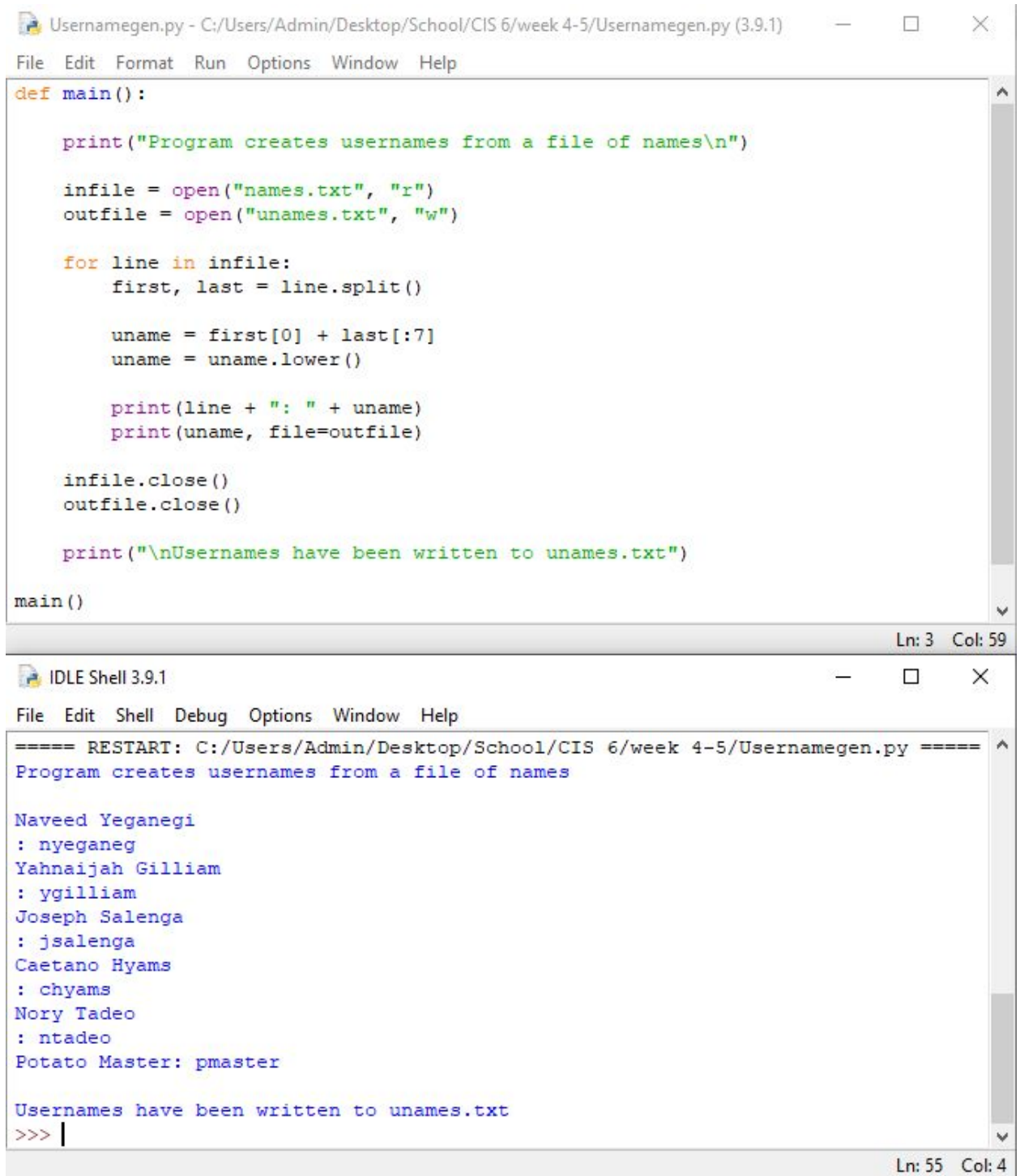
(g) `"{1:3}".format("14")` => **Error, 1 is out of range for index arguments**

## Task 4: Programming Exercises

**Instructions:** Use Python IDLE to write and execute below exercises from the book chapter 3. Attach Snipping photos of your source code and executions of the code in Python shell. Make sure to create separate files for each exercise

## Book Chapter 5: Write and execute below program from the book

### 1. username.py (from the lecture videos)



```
Usernameegen.py - C:/Users/Admin/Desktop/School/CIS 6/week 4-5/Usernameegen.py (3.9.1)
File Edit Format Run Options Window Help

def main():

    print("Program creates usernames from a file of names\n")

    infile = open("names.txt", "r")
    outfile = open("unames.txt", "w")

    for line in infile:
        first, last = line.split()

        uname = first[0] + last[:7]
        uname = uname.lower()

        print(line + ": " + uname)
        print(uname, file=outfile)

    infile.close()
    outfile.close()

    print("\nUsernames have been written to unames.txt")

main()
Ln: 3 Col: 59

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help

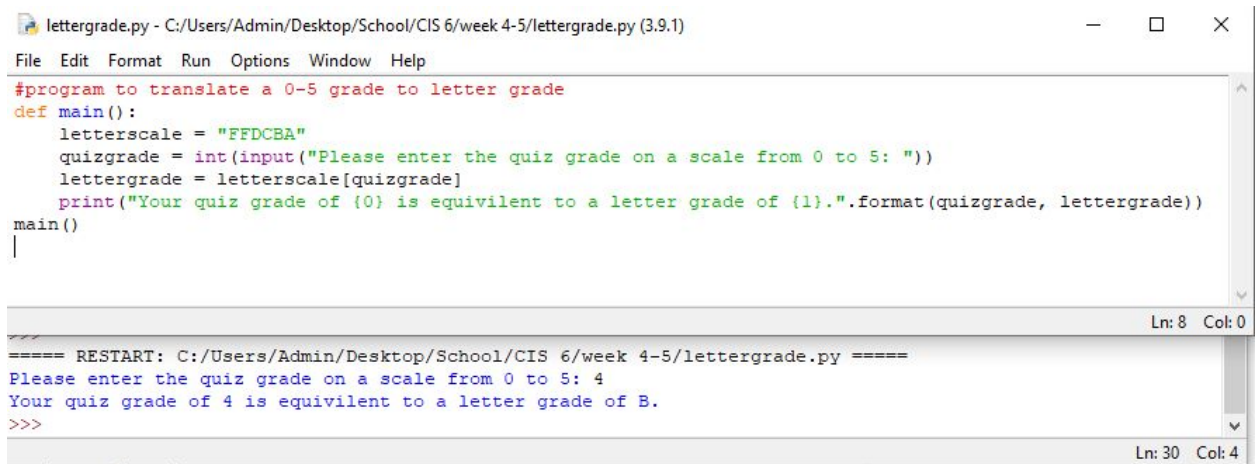
===== RESTART: C:/Users/Admin/Desktop/School/CIS 6/week 4-5/Usernameegen.py =====
Program creates usernames from a file of names

Naveed Yeganegi
: nyeganeg
Yahnaijah Gilliam
: ygilliam
Joseph Salenga
: jsalenga
Caetano Hyams
: chyams
Nory Tadeo
: ntadeo
Potato Master: pmaster

Usernames have been written to unames.txt
>>> |
Ln: 55 Col: 4
```

## Chapter 5 - Programming Exercises (Page 171-174).

### Exercise 2: 5-point grades



```
lettergrade.py - C:/Users/Admin/Desktop/School/CIS 6/week 4-5/lettergrade.py (3.9.1)
File Edit Format Run Options Window Help
#program to translate a 0-5 grade to letter grade
def main():
    letterscale = "FFDCBA"
    quizgrade = int(input("Please enter the quiz grade on a scale from 0 to 5: "))
    lettergrade = letterscale[quizgrade]
    print("Your quiz grade of {0} is equivalent to a letter grade of {1}.".format(quizgrade, lettergrade))
main()
|

Ln: 8 Col: 0

===== RESTART: C:/Users/Admin/Desktop/School/CIS 6/week 4-5/lettergrade.py =====
Please enter the quiz grade on a scale from 0 to 5: 4
Your quiz grade of 4 is equivalent to a letter grade of B.
>>>

Ln: 30 Col: 4
```

### Exercise 3: 100-points grades

### Exercise 4: Acronyms

### Exercise 9: Word count

### Exercise 10: Average word length

### Exercise 11: Formatted caos.py

### Exercise 12: Formatted futval.py

## Week 2:

### Exercise 1: Date conversion with format method

The image shows a screenshot of a Python IDE with two windows. The top window, titled 'datedisplay.py - C:\Users\Admin\Desktop\School\CIS 6\week 4-5\datedisplay.py (3.9.1)', contains the following Python code:

```
def main():
    print("Displays month and day of entered date")
    datenum = input("Please enter the date in mm/dd/yyyy format: ")
    datemonth, dateday, dateyear = datenum.split("/")

    months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun",
              "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
    monthnum = months[int(datemonth) - 1]

    print("The date is:", monthnum, dateday + ", " + dateyear)
main()
```

The bottom window, titled 'IDLE Shell 3.9.1', shows the execution of the script. It displays the following output:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Desktop\School\CIS 6\week 4-5\datedisplay.py =====
Displays month and day of entered date
Please enter the date in mm/dd/yyyy format: 02/28/2021
The date is: Feb 28, 2021
>>>
```

## Exercise 7: Caesar Cypher

## Exercise 13: File Input/Output