

# CS 111 – Introduction to Computer Science – Fall 2017

## Lab Assignment #3

### *Strings and Formatting \* (20pts)*

---

**Due Date:** at 11:59pm on Saturday, February 24.

---

The purpose of this lab is to provide practice working with strings and producing formatted output. You will be modifying a program from the previous lab and writing two programs from scratch.



Before getting started with the lab, create a folder named `lab3` within your `U:\cs111` folder and use it to save your work from this lab.

### Random Character



You are to write a program in which the user enters a text string and the program randomly picks a character from the string and replicates it 25 times. For example, if the user enters the text

`Elephants.`

and the program randomly chooses the letter `e` at index position 2, it would print

`eeeeeeeeeeeeeeeeeeeeeeeeeeee`

The algorithm required to solve this problem is as follows:

1. Prompt and read a text string from the user.
2. Randomly choose a character from the user-entered string. Think about the following questions to solve this problem:
  - How do you determine the length of a string?
  - What is the first index in a string and what is the last index?
  - How do you ask Python for a random number?
  - How you get a character from a string via its index?
3. Replicate the character 25 times.
4. Display the replicated string.

Your program must be written to the following specifications:

- Use `randomChar.py` as the name of your program.

---

\*Based on the labs of Dr. Rance Necaise

- Use a meaningful prompt and assume the user will enter at least one character.
- The only required output is the string containing the replicated character.
- Include a file prolog that includes the name of the file, your name and today's date and a brief description of what the program does.
- Use meaningful variable names and include descriptive comments for each step or block of code.

You should fully verify that your program works correctly by testing multiple text strings. After completing the program, be sure the file is saved.



## Formatted Output

To provide practice formatting strings, you are going to create a new version of the cylinder properties program to have it produce nicely formatted output.



Load your `cylinder2.py` program from your `U:\cs111\lab2` folder into *Wing*. Save a new version of the file named `cylinder3.py` into your `U:\cs111\lab3`.



You are to modify the program such that the resulting output is formatted as shown below (where the “C” in `Cylinder` is at the left edge of the output window)

```
Cylinder dimensions:
height = 12.000 radius = 1.500
  area          =    113.097
  volume        =     84.823
  circumference =     9.425
```

The output of the program has the same basic output as before, but with all numerical values displayed to 3 decimal places and the three computed values aligned along the decimal point. **Note** that your output should look *exactly* as shown above when the user enters 12.0 for the height and 1.5 for the radius.

Be sure to fully test your program to verify it works for different input values. After which, make sure to save the file.



## Email Address

In the previous problems, you were provided with the algorithm required to solve the given problem. For the final part of the lab, you will be designing an algorithm for a simple problem and then implementing that algorithm as a Python program.



You are to design and write a program that constructs the username portion of an email address based on a person's full name – first, middle, and last. The format of the username will be

`lastnameXY`

where X and Y represents the initials of the first and middle names. For example, the username for John Tyler Smith would be

`smithjt`

The name, which can be comprised of mixed case letters, should be entered at three separate prompts. After which, a correctly formed email address, including the username, should be printed to the terminal (note the address part will always be `ezmail.org`):

`lastnameXY@ezmail.org`

To begin, you are to design an algorithm that correctly solves this problem. Remember, an algorithm specifies the specific individual steps required to solve a given problem. For your algorithm, you are to specify the steps as a numbered list in a similar fashion as used with the algorithm for the random character problem earlier in the lab. Your algorithm should be included as part of the file prolog, immediately below the file description.

After completing the algorithm, you can then implement the algorithm by creating a Python program. Your program must be written to the following specifications:

- Use `emailAddress.py` as the name of your program.
- Use meaningful prompts and assume the user will enter text for each prompt.
- The only output should be the resulting email address.
- There should be no spaces in the email address.
- Your solution will require the use of the `lower` string method to ensure that the resulting email address is comprised of only lowercase letters.
- Include a file prolog that includes the name of the file, your name and today's date and a brief description of what the program does. Include the algorithmic steps in the file prolog, immediately following the description.
- Use meaningful variable names and include descriptive comments for each step or block of code.

You should fully verify that your program works correctly by testing multiple text strings. After completing the program, be sure the file is saved.



## Finishing Up

When you are finished with the lab, you need to show me that your code runs and correctly computes the solution for each part of the lab. Also, you need to submit the source files for grading. To submit the files, find the lab assignment on Canvas and upload the three files:

- `randomChar.py`
- `cylinder3.py`
- `emailAddress.py`