

CompSci 101 - Assignment 01

Due: 4:30pm, Tuesday 20th March 2017.

Worth: This assignment is marked out of 15 and is worth 2.5% of your final mark.

Topics covered:

- Using variables
- Arithmetic operators
- Printing output
- Manipulating string objects
- Generating random numbers
- Getting user input

NOTE: Each of your files must include a ***docstring*** at the top of the file containing your name, username, ID number and a description of the program. When solving these questions you must ***only*** use content covered in ***lectures 1 to 6***.

Submit the files containing your exercises using the Assignment Dropbox:

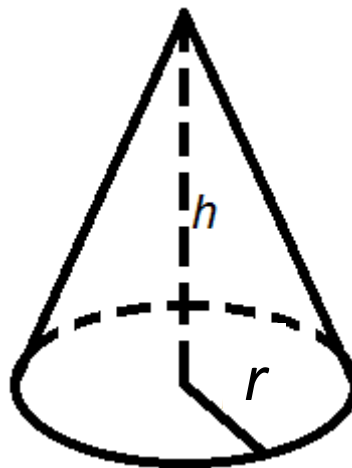
<https://adb.auckland.ac.nz/Home/>

QUESTION 1 (3 MARKS)

A right circular cone is a 3-dimensional shape that has 2 key dimensions:

1. The base radius – r
2. The height – h

An illustration of a right circular cone with these dimensions labelled is shown below:



The formula to calculate the surface area of a cone is:

$$\text{Surface Area} = \pi r \left(r + \sqrt{r^2 + h^2} \right)$$

The formula to calculate the volume of a cone is:

$$\text{Volume} = \frac{1}{3}\pi r^2 h$$

Write a program that calculates the surface area and volume of a right circular cone. Name your program file 'YourUserNameA1Q1.py', e.g. dazh001A1Q1.py

Your program will need to:

- Display a banner.
- Ask the user to enter values for the base radius and height. You can assume that the user will always enter integer values.
- Calculate the surface area and volume of the cone using the values for the radius and height entered by the user.
- Display these calculated values rounded to **4** decimal places.

Two example outputs using the completed program are shown below. Your program **must** give the output in the **same format** as the outputs in the two examples below.

<pre>##### Right Circular Cone Surface Area and Volume Calculator ##### Radius: 5 Height: 10 Surface Area: 254.1602 Volume: 261.7994</pre>	<pre>##### Right Circular Cone Surface Area and Volume Calculator ##### Radius: 25 Height: 13 Surface Area: 4176.5908 Volume: 8508.4801</pre>
--	---

QUESTION 2 (4 MARKS)

Write a program which encrypts a word using a Caesar cipher. Name your program 'YourUsernameA1Q2.py', e.g. dazh001A1Q2.py

A Caesar cipher is a simple and well known encryption technique, where each letter in the original message is replaced by a letter a certain number of positions down the alphabet. You will be using a Caesar cipher with a shift of 5. In other words the letter "a" would be replaced by the letter "f" while the letter "x" would be replaced by the letter "c"

Your program should use the following two variables initialized to the values specified below:

```
alphabet = "abcdefghijklmnopqrstuvwxyz"
shift = 5
```

The user will enter a 5 letter word to be encrypted by the Caesar cipher. You can assume that the word will always consist of 5 alphabetical characters only.

Two example outputs using the completed program follow. Your program **must** give the output in the **same format** as the outputs in these two examples.

```
*****
*Caesar Cipher Encryption*
*****
```

```
Please enter a 5 letter word: Damir
Encrypted word: ifrnw
```

```
*****
*Caesar Cipher Encryption*
*****
```

```
Please enter a 5 letter word: kazoo
Encrypted word: pfett
```

QUESTION 3 (4 MARKS)

Write a program that prompts a user to enter their name, and using that name, generates them a University of Auckland username. Name your program 'YourUserNameA1Q3.py', e.g. dazh001A1Q3.py

A University of Auckland username consists of the first letter of the user's first name (in lowercase), followed by the first three letters of their surname (also in lowercase), and followed finally by a randomly generated three digit number from 1 to 999.

You can assume that the user will always enter only a first name and a surname separated by a space as their name. You can also assume that their surname will have at least 3 letters.

Below are three example outputs using the completed program. Your program must give the output in the same format as the outputs in the three examples below.

```
*****
University of Auckland Username Generator
*****
```

```
Please enter your name: Damir Azhar
```

```
Your username is dazh007
```

```
*****
University of Auckland Username Generator
*****
```

```
Please enter your name: Adriana Ferraro
```

```
Your username is afer095
```

```
*****
University of Auckland Username Generator
*****

Please enter your name: Ann Cameron

Your username is acam988
```

QUESTION 4 (4 MARKS)

Write a program which, given a start time and a time to add, calculates the final time (in hours, minutes and seconds). Name your program 'YourUserNameA1Q4.py', e.g. dazh001A1Q4.py.

The times in this program use a 24 hour clock, e.g., 18:36:45 is 36 minutes and 45 seconds past 6pm. The first two lines of your program initialise the variables in the following way (you will need to change the values assigned to these variables when testing your program):

```
start_time = "13:05:25"
time_to_add = "01:14:54"
```

The final time should always be a time between 00:00:00 and 23:59:59. The start time, the time to be added and the final time always have the form "hh:mm:ss", i.e. a single digit hour, minute or second has a "0" in front of it. Look at the following example outputs using the completed program (using different variable values). Your program **must** give the correct output in the **same format** as the outputs in these examples. The lines of "=" symbols contain 35 characters.

```
=====
Start time: 09:59:12
Seconds to add: 79491 (22:04:51)
End time: 08:04:03
=====
```

```
=====
Start time: 09:59:12
Seconds to add: 18030 (05:00:30)
End time: 14:59:42
=====
```

```
=====
Start time: 23:32:59
Seconds to add: 1621 (00:27:01)
End time: 00:00:00
=====
```