

# COMPSCI 101 - Assignment 01

**Due:** 4:30pm, Thursday 2<sup>nd</sup> April 2020.

**Worth:** This assignment is marked out of 20 and is worth 3% 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 (4 MARKS)

Pi ( $\pi$ ) is a mathematical constant that describes the ratio of a circle's circumference to its diameter. Pi is an irrational number and as such cannot be expressed exactly as a fraction. Numerous approximations of pi exist, to various levels of precision. Once such approximation, accurate to 7 decimal places, is shown below:

$$\pi \approx \left( \frac{802\phi - 801}{602\phi - 601} \right)^4$$

where  $\phi$  (phi) is the golden ratio:

$$\phi = \frac{1 + \sqrt{5}}{2}$$

Write a program that implements the above approximation of pi. Name your program file 'YourUsernameA1Q1.py', e.g. dazh001A1Q1.py. You must round your result to 7 decimal places. Your program must produce the following output:

```
Pi is equivalent to 3.1415926
```

## QUESTION 2 (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 'YourUsernameA1Q2.py', e.g. dazh001A1Q2.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), followed finally by a randomly generated three-digit number from 1 to 999 (**you must generate only 1 random number, not 3**).

You can assume that the user will always enter only a first name and a surname separated by a single space character 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. User input is shown below in red font.

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

Please enter your name: Damir Azhar

Your username is dazh007
```

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

Please enter your name: Ann Cameron

Your username is acam067
```

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

Please enter your name: Adriana Ferraro

Your username is afer276
```

### QUESTION 3 (4 MARKS)

Write a program which decrypts a four-letter word. Name your program 'YourUsernameA1Q3.py', e.g. dazh001A1Q3.py.

Your program will prompt the user to enter an encrypted string that is 4 characters long. This string has been encrypted using the Caesar cipher. With a Caesar cipher, each character in the original text is replaced by a character a certain number of places further along in the alphabet.

For example, if you used a Caesar cipher with a shift of 4, the character "a" in the unencrypted string would be replaced by the character "e" in the encrypted string, the character "b" in the unencrypted string would be replaced by the character "f" in the encrypted string, the character "c" by the character "g" and so on. Similarly, the character "w" in the unencrypted string would be replaced by the character "a" in the encrypted string, the character "x" in the unencrypted string by the character "b" in the encrypted string, and so on.

As your program decrypts an encrypted string, it works in the opposite direction. Again, assuming a shift of 4, the character "a" in the encrypted string would be replaced by the character "w" in the unencrypted string, the character "b" in the encrypted string would be replaced by the character "x" in the unencrypted string and so on.

Your program **must** use the following variables initialized to the values specified below:

```
alphabet = "abcdefghijklmnopqrstuvwxyz"  
shift = 6
```

You can assume that the user will always enter a string consisting of 4 lowercase alphabetical characters.

Three example outputs using the completed program follow. Your program **must** give the output in the **same format** as the outputs in these three examples. User input is shown below in red font.

```
Enter encrypted string: iuur  
Unencrypted string: cool
```

```
Enter encrypted string: futk  
Unencrypted string: zone
```

```
Enter encrypted string: cugn  
Unencrypted string: woah
```

#### QUESTION 4 (4 MARKS)

Write a program which solves the change-making problem for a particular currency. Name your program 'YourUserNameA1Q4.py', e.g dazh001A1Q4.py

The change-making problem involves selecting the **fewest** number of coins from a given set of coin denominations to represent a given value. The currency in question has the following coin denominations: 1, 5, 10, 25, 50 and 100. For example, given this currency and a value of 76 cents, the solution to the change making problem would be to select one 50 cent coin, one 25 cent coin, and one 1 cent coin.

The value to represent will be **entered by the user**. You can assume that the user will always enter a positive integer value. Your program **must** give the output in the **same format** as the outputs in the following three examples. The user input is shown below in red font. The lines of "=" symbols contain 19 characters.

```
Value: 76

You will need:
=====
 100s: 0
  50s: 1
  25s: 1
  10s: 0
   5s: 0
   1s: 1
=====
```

```
Value: 254

You will need:
=====
 100s: 2
  50s: 1
  25s: 0
  10s: 0
   5s: 0
   1s: 4
=====
```

```
Value: 83

You will need:
=====
 100s: 0
  50s: 1
  25s: 1
  10s: 0
   5s: 1
   1s: 3
=====
```

#### QUESTION 5 (4 MARKS)

Write a program which implements a dice game. Name your program file 'YourUserNameA1Q5.py', e.g. dazh001A1Q5.py.

The aim of the game is to reach a score as close as possible to 100 in three rounds. Each round consists of throwing five random dice. The user then chooses two of the dice values, where the two dice values chosen form a two digit score which is added to the user's current total. For example, if the user first chooses a dice with the value 3 and then a dice with the value 5, 35 is added to the user's total. The first dice chosen is therefore the tens digit and the second dice chosen is the units digit.

The five random dice are displayed with one space between each dice:

Your dice: 3 5 3 4 1

To choose the dice the user enters a number 1, 2, 3, 4, or 5 indicating which of the five dice they wish to choose (i.e. the position of the dice and **not** the value of the dice). This process is repeated three times. At the end of the program you will need to inform the user of their score and how far off they are from the goal of 100.

When writing your program, you **must** use the following statement to initialize the user's current score:

```
current_score = 0
```

Below are two example outputs using the completed program. The user input is shown in red font. You can assume the user will always enter valid input (i.e. an integer between 1 and 5). Your program **must** give the output in the **same format** as the outputs in the following two examples. The top string of "\*" symbols has a length of 45 and the bottom string of "\*" symbols has a length of 29.

```
*****
REACH 100 IN THREE ROUNDS!  Initial total: 0
*****

Round 1:
Your dice: 2 4 1 1 2
  Tens? 2
  Units? 3
Dice value: 41
Your current total: 41

Round 2:
Your dice: 6 1 2 1 6
  Tens? 3
  Units? 5
Dice value: 26
Your current total: 67

Round 3:
Your dice: 5 6 3 2 1
  Tens? 3
  Units? 4
Dice value: 32

*****
Your final score: 99
You are 1 away from the goal
*****
```

```
*****
REACH 100 IN THREE ROUNDS!  Initial total: 0
*****

Round 1:
Your dice: 5 2 1 6 2
  Tens? 1
  Units? 4
Dice value: 56
Your current total: 56

Round 2:
Your dice: 3 1 4 5 4
  Tens? 2
  Units? 3
Dice value: 14
Your current total: 70

Round 3:
Your dice: 5 6 1 4 1
  Tens? 4
  Units? 5
Dice value: 41

*****
Your final score: 111
You are 11 away from the goal
*****
```