Assessment Sheet 2

This is the second of eight assessment sheets that you will be given over the duration of this unit. You will need to complete the tasks as outlined below and then document them in a word document. At minimum, you should provide screen shots of:

- Your code
- The output that your code generates

In instances where your code could give different outputs depending on what values it is given; you should provide multiple screen shots of the console screen showing different outputs to demonstrate that the code is working correctly.

This assessment sheet will focus around **loops**

To achieve a D grade

Successful completion of this task (documented in your portfolio) will allow you to achieve up to a D1 grade. Please follow all instructions:

In this task you will write a small program that will mark an 'X' on a line of 20 dashes. This program will use a loop and concepts from the previous weeks.
Take this program one step at a time. Here is a break down to help you:

- Firstly, display a single dash on the screen
- Then use a loop to make that dash print out 20 times
- Now modify the program so that it asks the user for a number between 1 and 20. This number will represent the location of the X
- Finally, modify the loop so that when it reaches the users chosen location, it will print out an 'X' instead of a dash

Here is an example of what this program should output:

```
Please enter a number between 1-20:
3
--X-----------------
```

Note that the user entered 3, thus, what would have been the 3rd dash is an X instead.

Complete the above task and then document it in your portfolio.

Note, to achieve a D

- The write up must demonstrate that the code works
- For a D grade, you only need to provide screen shots of your code and some of the different outcomes it can give - At this stage you do not need to do any writing

- Your word document should have appropriate headings to ensure that this task can easily be identified alongside your work from other assessment sheets.

To achieve a C grade

Modify the above program so it now behaves as follows:

- In the event that the user enters a number that is not between 1 and 20, the program should not display any dashes. It should instead display an error message saying "Sorry, the number needs to be between 1 and 20"
- In the event the user enters a correct number, then the program should work as stated in the D grade tasks

Add the modifications to your portfolio

Furthermore, to achieve a C

- The above task should be completed and documented in the portfolio. The portfolio should demonstrate that the code works
- ALL tasks must be accompanied by written descriptions or annotations. These must show some basic understanding of what the code is doing

To achieve a B grade

Modify the above program (C grade section) so it now behaves as follows:
Please note, this task will require some thought and problem solving!

- Once the user has entered where they want the 'X' to appear, they should then be asked to enter a second number. This will represent the location of a second 'X' - specifically, it will represent the distance the second 'X' will be from the first one.
  For example, if the user entered 3 for the first X and then 5 for the distance. The first X will be printed in location 3 and the second X will be printed 5 dashes away from it.

Here is an example of the output expected from this:

```
Please enter a number between 1-20:
3
Please enter the distance between the two X's
5
--X----X------------
```

Notice that the user entered 5 for the distance and therefore the second 'X' is 5 spaces away from the first one.

Please note, there are a number of ways you can potentially achieve this, and I will accept many of them, but the following rules should be observed:

- Only one loop is required for this program
- Any printf's that are responsible for printing out the line (- or X) should only print 1 character at a time

To achieve a B:

- The above task should be completed and documented in the portfolio. The portfolio should demonstrate that the code works
- All tasks must be accompanied by written **explanations** - All new concepts, as they are encountered should be explained. For example, I would expect loops to be explained. This is in addition to the C grade tasks where you describe what your code is doing
- Any concepts that have been explained in work covered by previous task sheets do **not** need to be explained again (i.e. variables)

To achieve an A grade

- Your written explanations from the B task should be very detailed

As before, the A grade task will require some independent research.

We have only looked in to while loops this week, however there are other ways of looping. You will need to investigate alternative approaches, demonstrate them and document them in your portfolio.

For this week, I will tell you exactly what to look into:

Many of the tasks you have attempted this week involved some form of 'counting.' Where counting is involved, while loops are not actually the best option to use.
A better loop in these situations would be the **For Loop**

- Repeat one of the tasks above, but this time use a for loop.
- Document this in your portfolio and provide a detailed explanation of for loops along-side
- Please give this section an appropriate heading to ensure it is easy to find.

Note, from here on in, you are free to use for loops in this unit as and when you deem appropriate.
While loops will continue to be what is taught, however.