COMP 2152 – Assignment 1 – Winter 2021

Important:

1. This is an individual assignment (complete the requirements on your own).
2. Do not share the assignment requirements with any former or future students in COMP 2152. Do not share this assignment requirements online in any format, anywhere.
3. You may choose to complete the all requirements or attempt to complete as much as possible.
4. Credit will be awarded for requirements completed correctly and entirely with the portion of program functioning as requested. No partial marks will be awarded.
5. You are allowed to make assumptions about application functionality not mentioned in the project idea (which is more of a general guideline).
6. At minimum your project must display the functionality described for the idea selected. Any functionality described that is not a part of your submission will result in grade penalties at the discretion of the instructor.

Submission:

1) Submit Python files only. No other format is accepted.
2) DO NOT submit zipped (compressed) files. Any compressed files will not be marked.
3) **Submit in GB Learn and Blackboard!! If you do not have access to GBLearn please contact support@GBLearn.com as soon as possible to obtain access. The GBLearn submission will be used for plagiarism checking and the Blackboard submission is where it will be graded.**

Get the user input and create a thematic list of strings. The program should ask for specific input theme. For example: name of the planets in the solar system, name of your favourite vegetables, list of programming languages you hope to learn, etc. The user should enter strings for the list until a condition is met such as entering a specific word or character. There should not be hard code list size.

Each user input should be one list element.

Then process the list as follows:

4) Count and display a list of the length of each element in the list with a relevant message.
5) Use the random function to generate a random number between 0 and the maximum length of the list. This number will be used as an index to the list elements. The following code generates a random integer between 1 and 10.

```
Import random
print(random.randint(1, 10))
```

6) Have the user enter a character of their choice and check if the character exists in the string with the index number just generated at random.
7) For example:
   a) The list is [“Mars”, “Venus”, “Earth”, “Saturn”, “Pluto”]
   b) The index number generated at random is 3
   c) The user enters character ‘A’
   d) The program checks to see if ‘A’ is in the list element with the index of 3.

e) The program should display a relevant message if the character is found in the string.

The search for the character should return true regardless if the user enters a lowercase ‘a’ or an uppercase ‘A’ . If the letter exists in the string either in uppercase or lowercase, the search should return true and a relevant message should be displayed.

Assignment 1 application requirements:

Requirements description	Points
1. Get the user input and create a thematic list of strings. The program should ask for specific input theme. For example: name of the planets in the solar system, name of your favourite vegetables, list of programming languages you hope to learn, etc. The user should enter strings for the list until a condition is met such as entering a specific word or character. There should not be hard code list size. Each user input should be one list element.	30
2. Then process the list as follows: a. Count and display a list of the length of each element in the list with a relevant message.	10

b. Use the random function to generate a random number between 0 and the maximum length of the list. This number will be used as an index to the list elements. The following code generates a random integer between 1 and 10.
`Import random`
`print(random.randint(1, 10))`
	10

c. Have the user enter a character of their choice and check if the character exists in the string with the index number just generated at random.	20
d. The search for the character should return true regardless if the user enters a lowercase ‘a’ or an uppercase ‘A’ . If the letter exists in the string either in uppercase or lowercase, the search should return true and a relevant message should be displayed. **For example:** • The list is [“Mars”, “Venus”, “Earth”, “Saturn”, “Pluto”] • The index number generated at random is 3 • The user enters character ‘A’ • The program checks to see if ‘A’ is in the list element with the index of 3. • The program should display a relevant message if the character is found in the string.	30
Total	**100**