Scripting Languages: Assignment 2.2- Portfolio Task 2 Brief

Overview

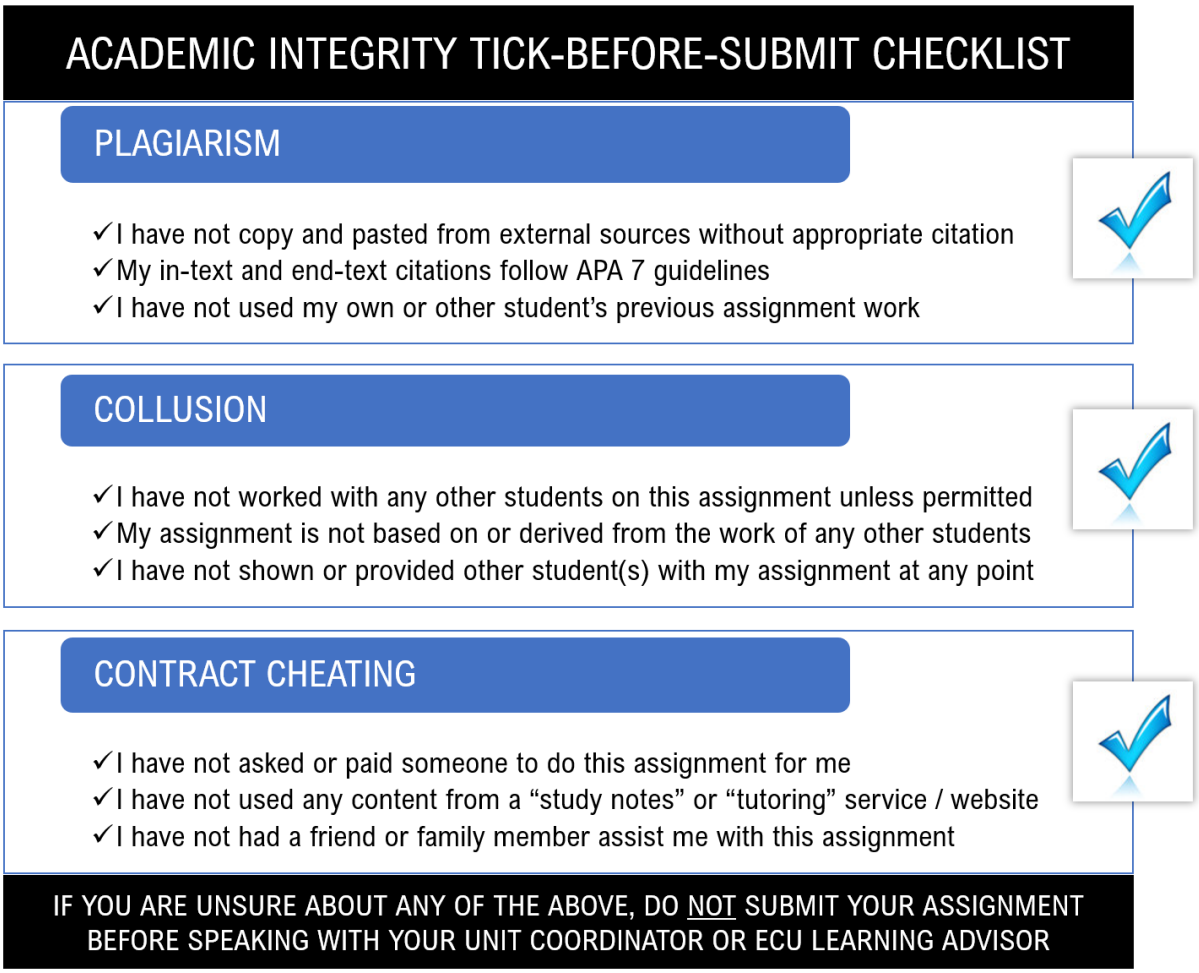
In this assignment you are required to write a script that demonstrates the extent to which you have understood the *shell script* concepts and practices addressed in Modules 1 to 6 inclusive. This assignment is worth **20%** (*20 marks*) of your unit grade and its completion will help you build skills required for the final assignment.

GENERAL REQUIREMENTS

- Your script will be marked on a standard Ubuntu Linux for desktop installation using the **bash** shell.
- Refrain from using *non-core* commands, tools and utilities in your bash shell scripts. Non-standard bash commands, tools and utilities will **not** be downloaded and installed by your tutor.
- Ensure each script you write is *fully self-contained* and is not configured to be dependent on external files, libraries or resources to run.
- Do **not** use the *trap* command in any of your scripts.
- All scripts you submit must contain your full name and student number at the top as comments.

IMPORTANT NOTES:

- This is an **individual** assignment only and must **not** be completed in collaboration with other students.
- You may **not** work with others, acquire code from others, or provide code to others.
- Further, you may **not** post any of the assignment tasks below to a code-development community of any kind seeking solutions or advice.
- Nor may you copy and paste, or otherwise reproduce, any shell script verbatim from sources external to the unit materials into your own scripts. You may examine (read) how external sources have approached particular shell scripting tasks with a view to generating ideas about how you might approach a particular task yourself as long as their code does **not** become your code.
- Where it is found that any of these restrictions have been ignored, academic misconduct proceedings may be initiated.
- Please read the checklist below and watch the associated video **BEFORE** submitting your assignment



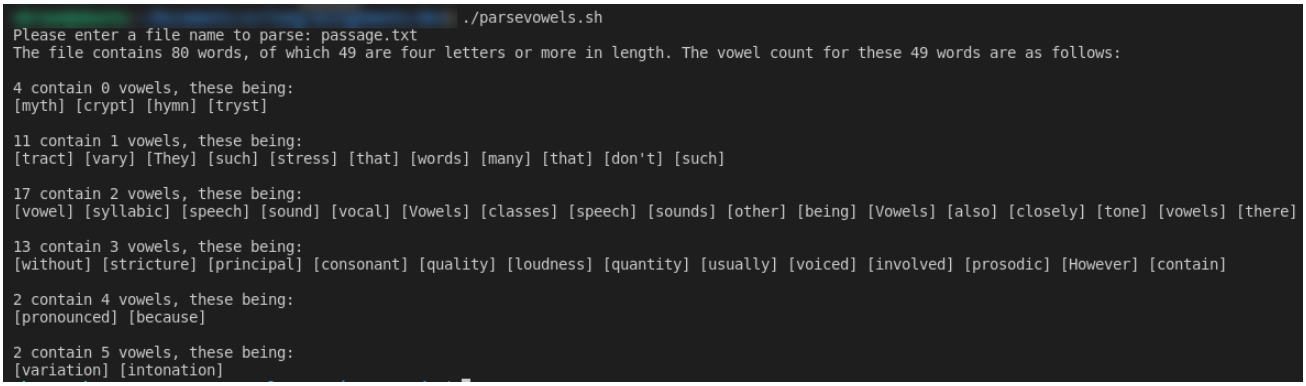
Watch this video before submitting your assignment

Task – Parse Vowels (20 marks)

Write a function that will accept a file name provided by the user and scan it word-by-word to determine how many vowels are in each word and output a report of the findings to the terminal.

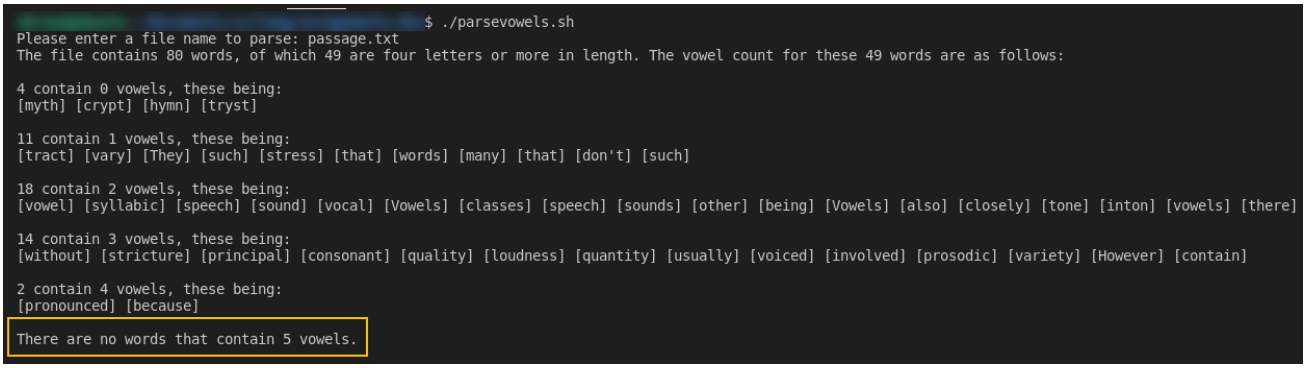
Requirements:

- Call the script *parsevowels.sh*, and its resident function *parseVowels()*.
- Ensure that **all** vowel parsing and results display logic is contained with the *parseVowels()* function.
- Only the code that a) prompts the user for a file name, b) validates the existence of this file, and c) passes the validated file name to the *parseVowels()* function is to reside within the script's main body.
- Only words within the nominated file that are four (4) or more characters in length are to have their resident vowels counted. Words of three (3) characters or less are to be excluded by your function's logic and not appear in the on-screen vowel count report.
- Your function is to output the results of the vowel count process as shown in the screenshot below:



Screenshot Task 2 - 1

- In the event that a particular vowel count has no words associated with it, inform the user as per the screenshot below:



Screenshot Task 2 - 2

- To assist you in writing this script, the *passage.txt* file that features in the above screenshot has been supplied. Please note however that your tutor will use a text (*.txt*) file with a different name and which contains a different passage of text. Do not therefore hard-code the name of, or text within, the *passage.txt* file.

- When your tutor marks your script, the text file (*.txt*) they use will be located in the same directory as your script when run. There is **no** need therefore to prompt for the location of this text file, but rather, just its name.
- To construct your function, use any combination of commands, utilities, control structures etc covered through *Modules 1-6* inclusive.

Watch the **Parse Vowels** demonstration video that accompanies this assignment brief on Canvas to see this script in action. This will give you a good idea of what your own script needs to do.

Marking Key

Marks will be awarded as follows:
[*Not Achieved / Doesn't Work* - 0.0]
[*Partially Achieved / Works, but not as stipulated* – ½ of available mark(s)]
[*Fully Achieved / Works exactly as stipulated* – all of available mark(s)]

ITEM#	ITEM	VALUE
1	All vowel parsing and results display logic is contained with the parseVowels() function	3
2	Script prompts user for a valid file name until one is provided and passes it to the parseVowels() function	3
3	Only words within the nominated file that are four (4) or more characters in length are subject to the vowel count logic	3
4	The script's function output **correct** results of the vowel count process in the format shown in Screenshot Task 2 - 1	8
5	In the event a particular vowel count has no words associated with it, user is informed as per Screenshot Task 2 - 2	3
	TOTAL	**/20**

How to submit your portfolio to Canvas

Submit the single shell script (bash) file you have created in a **zipped folder** with the following naming format:

`[surname]_[student-ID]_CSP2101_PF2.zip`

END OF ASSIGNMENT BRIEF