

Scripting Languages

Assignment 2.1: Portfolio 1

ASSIGNMENT BRIEF

Important points before you begin:

- Read this brief very carefully in full, and at the earliest opportunity
- If any of this brief's stipulations are unclear to you, it is **your** responsibility to seek timely clarification from the lecturer/unit coordinator well before the assessment's due date
- Be aware that if you misinterpret this assignment brief, either in part or in full, this misinterpretation will not be accepted as grounds for an appeal of the result allocated

Overview

In this assignment you will be required to write a script that demonstrates the extent to which you have understood the shell script (bash) commands, programmatic techniques and concepts addressed in Modules 1 to 5 inclusive. This assignment is worth 20% (20 marks) of your unit grade and its completion will help you build a subset of the skills you will require to successfully complete *Portfolio 2* and *Assignment 3*.

Your Academic Integrity Obligations

Your tutor, lecturer, and unit coordinator all take Academic Integrity very seriously and it cannot be stressed strongly enough how important it is that you fully understand your academic integrity obligations as a student of the University. In regard to all of this unit's assessments, all suspected instances of academic misconduct will be reported for investigation, which may result in substantial academic penalties for those concerned. If you are unfamiliar with the University's Academic Integrity Policy as it applies to all assignments you submit for this unit, [you can familiarise yourself with it here](#). If you are unsure of anything, please contact the Unit Coordinator for clarification before submitting this assessment.

AI Tool Utilisation

Whereas AI tools can serve as a very useful tool in the software engineering workplace, they are **not** to be used to complete any of the assessments in this unit, either in-part or in totality. This is because the focus of this unit is the development of authentic knowledge of, and skill in applying, various scripting languages to achieve specific outcomes, and not the use of AI tools to act on a programmer's behalf. This acquisition of authentic knowledge extends beyond the outcomes of this unit and is important to a student's future ability to be successful in obtaining a Work Integrated Learning placement, and ultimately, a job in industry. Employers already use AI tools extensively and know what they can and cannot do – they see no value in hiring graduates whose demonstrable skills do not exceed that of existing AI tools. Additionally, students suspected of representing AI output as their own work may be called upon to demonstrate functional knowledge of this work, which, if not forthcoming, may in turn lead to allegations of academic misconduct.

Please read the checklist below and watch the [associated video](#) **before** submitting this assignment.

ACADEMIC INTEGRITY TICK-BEFORE-SUBMIT CHECKLIST

PLAGIARISM

- ✓ I have not copy and pasted from external sources without appropriate citation
- ✓ My in-text and end-text citations follow APA 7 guidelines
- ✓ I have not used my own or other student's previous assignment work



COLLUSION

- ✓ I have not worked with any other students on this assignment unless permitted
- ✓ My assignment is not based on or derived from the work of any other students
- ✓ I have not shown or provided other student(s) with my assignment at any point



CONTRACT CHEATING

- ✓ I have not asked or paid someone to do this assignment for me
- ✓ I have not used any content from a "study notes" or "tutoring" service / website
- ✓ I have not had a friend or family member assist me with this assignment



IF YOU ARE UNSURE ABOUT ANY OF THE ABOVE, DO NOT SUBMIT YOUR ASSIGNMENT BEFORE SPEAKING WITH YOUR UNIT COORDINATOR OR ECU LEARNING ADVISOR

General Assignment Requirements

- Your script will be run in the *Azure Linux VM* offered at the beginning of semester. You may of course develop your script using one of the alternative Linux environments recommended, e.g. WSL/Ubuntu on Windows, however, it is recommended that you still test your script in the Azure environment provided *before* submission to make sure everything functions as expected.
- Ensure the script you write is *fully self-contained* and is not configured to be dependent on external files, libraries or resources to run. Non-observance of this requirement may cause your script to run incorrectly or not at all.
- Carefully check your submission before uploading it to Canvas. **What you submit is what gets assessed!** If you make a submission error, e.g. submit a wrong file, an empty .zip archive etc, no further/subsequent submissions will be accepted, which may result in a substantial loss of marks, or even a zero (0) result in some cases.
- You must only submit a **single** shell script (.sh) file named *parsechars.sh* contained within a .zip file with the stipulated name in any *individual upload action*. Do **not** upload multiple files/zip archives in the same upload action as all will be considered invalid and will not be assessed. Also note that only the most recent individual submission made (as determined by the timestamp Canvas allocates) will be assessed.

Task – Parse Characters (20 marks)

Using **only** the commands, utilities and programmatic techniques addressed in lectures 1-5 inclusive, write a script that processes a list of strings contained in an external text file, and for each string in the list, parses out characters that are *allowed* characters from those characters that are *disallowed*, printing a report to the screen of *total [T]*, *allowed [A]* and *disallowed [D]* characters found, and their counts.

The following characters are *allowed*:

- Uppercase letters
- Numbers
- The following subset of special characters:
*#!._-

All other characters are *disallowed*.

Required Script Functionality

- When your script is run, it is to prompt the user for the name of a file in their *current working directory*.
- If the file name provided does not exist in the user's current working directory, or if it does exist but contains no data (is empty), the script is to terminate immediately with an appropriate exit code, and the user being advised as to the nature of the error in the form of a terminal message.
- If the file name entered by the user does exist and does contain data, your script is to process each of the strings it contains and parse out its *allowed* and *disallowed* characters, with the results reported to the screen precisely as shown in the screenshot below.

Please note: Not all of the script's required functionality is demonstrated in the screenshot below.

```
$ ./parsechars.sh
Enter the name of the candidate password file (including ext): strlist1.txt
7PL;H^n+WZ91u] [T: 14] [A: 19ZWHL7 (8)] [D: ]u+n^; (6)]
q.UihJE6IBj||,y [T: 15] [A: BI6EJIU. (8)] [D: y,||jhg (7)]
Rvgr>^_a"$Zs; [T: 13] [A: Z_R (3)] [D: ;s$a^>rgv (10)]
sp&ZEeG&R40#A+[ [T: 15] [A: A#04RGEZ (8)] [D: [+&e&ps (7)]
xQaGz%J4nx6X:* [T: 14] [A: *X64JGQ (7)] [D: :xn%zax (7)]
6{44UZ(qo=0&$bY [T: 15] [A: Y0ZU446 (7)] [D: b$&=oq({ (8)]
OY'9@?d<t.-40 [T: 13] [A: 04-.9YO (7)] [D: t<d?@' (6)]
Ge+X3#Zx26NbyT$ [T: 15] [A: TN62Z#3XG (9)] [D: $ybx+e (6)]
0: ^A0n;-z0y4k+y [T: 15] [A: 40-0A0 (6)] [D: y+kyz;n^: (9)]
V>]N#'P5#|0GJd [T: 14] [A: JG0#5P#NV (9)] [D: d|'> (5)]
```

Other Compulsory Requirements

- Call the script `parsechars.sh`.
- Your full name and student number **must** be placed at the top of your script (as comments) immediately after the *shebang* line. *Note: There will be a one (1) mark deduction from your achieved score if either or both of these items are absent.*
- Immediately after your name and student number, provide a description of no more than 220 words that outlines **how** you developed your script to solve for the stipulated task. Be sure to address the main commands, utilities and programmatic techniques you employed from Lectures 1-5 inclusive, and the role each serves in the *context* of the script's functionality. This statement is of course to take the form of *comments*. Please split the statement over multiple lines for easy readability.
- To construct your script, use any combination of commands, utilities and programmatic techniques covered throughout the unit's lecture slides (*Modules 1-5 inclusive*). Do **not** use *bash/shell script commands, utilities and programmatic techniques not addressed in the unit's lecture slides (Modules 1-5 inclusive)*.
- Your script **must** contain concise 'in-situ' comments that accurately explain **all** of the code elements it contains. *Note: Comments that are not relevant and accurate in regard to the code they describe, or a complete lack of comments, will not only cost marks, but may be viewed as suggestive of possible academic misconduct.*
- Any *temporary* files and/or folders created by your script in the course of its execution are to be removed from the assessor's system when the script terminates. *Note: There will be a one (1) mark deduction from your achieved score if any temporary files and/or folders created by your script are left behind on the assessor's machine after your script terminates.*
- The efficiency and correctness with which the commands, utilities and programmatic techniques within your script have been utilised will also form part of your mark, so please pay close attention to this aspect of your code as well. For example, your `parsechars.sh` script is expected to make use of appropriately selected and correctly applied commands/options, conditional testing and loop structures, IFS management, regular expressions, and piping. The total lines-of-code your script employs will also be considered, with deductions applicable should this significantly exceed the total deemed necessary to achieve the stipulated outcomes.
- To assist you in the development and testing of your script, you have been provided with three (3) test files named `strlist1.txt`, `strlist2.txt`, and `strlist3.txt`, each of which contains a list of random strings of varying lengths and character combinations. *Important Note:* Your assessor will **not** use any of these text files for the marking process so do not hard code any of them into your script. Your assessor will use a *different* set of text files (.txt) files, with different names and containing different random string lists.

Important Note: If your script does not run for any reason, e.g. hard-coding of files/directories/paths, use of a development environment not compatible with the Azure Linux VM offered at the beginning of semester, only a partial mark will be awarded on a code-readthrough basis (at the assessor's discretion). Your assessor will **not** fix non-functional, dysfunctional or incompatible scripts.

Assessment Rubric

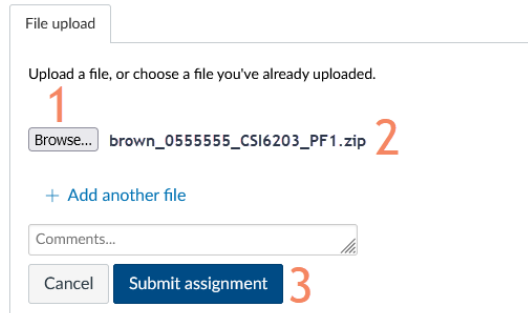
Your assignment will be marked using an online rubric, the assessment criteria and weightings of which are outlined below:

SCRIPT DOCUMENTATION [4 marks]		
Unsatisfactory: (0-1 marks)	Satisfactory: (2-3 marks)	Excellent: (4 marks)
<ul style="list-style-type: none"> Numerous discrepancies or deficiencies exist in the description of "how" the script has been designed and developed to solve for the stipulated requirements and expectations and/or For <i>much/all</i> of the script's code elements, commenting was not accurate and/or complete in explaining their purpose and functionality 	<ul style="list-style-type: none"> Some <i>minor</i> discrepancies or deficiencies exist in the description of "how" the script has been designed and developed to solve for the stipulated requirements and expectations and/or For <i>some</i> of the script's code elements, commenting was not entirely accurate and/or complete in explaining their purpose and functionality 	<ul style="list-style-type: none"> Description comprehensively and accurately explains <u>how</u> the script was designed and developed to solve for its stipulated requirements and expectations and Script contains accurate and complete commenting that explains the purpose and functionality of <u>all</u> its code elements
FUNCTIONALITY & IMPLEMENTATION [10 Marks]		
Unsatisfactory (0-4 marks)	Satisfactory (5-9 marks)	Excellent (10 marks)
<ul style="list-style-type: none"> Some to much of the required functionality and/or compulsory requirements have not been implemented to a satisfactory level, and/or not precisely as described in the assignment brief, and/or Numerous major errors, omissions, and/or oversights evident 	<ul style="list-style-type: none"> All required functionality and compulsory requirements have been implemented to a satisfactory level, but not precisely as described in the assignment brief in some to several cases, and/or with some minor errors, omissions, and/or oversights evident 	<ul style="list-style-type: none"> All required functionality and compulsory requirements have been implemented in full, precisely as described in the assignment brief, and are entirely free of error, omission, or oversight
CONVENTION, EFFICIENCY & COMPLIANCE [6 marks]		
Unsatisfactory (0-2 marks)	Satisfactory (3-5 marks)	Excellent (6 marks)
<ul style="list-style-type: none"> For the most part, the script's code is <u>not</u> properly structured and/or In many cases, the selection of commands, utilities and programmatic techniques was poor resulting in significant script inefficiencies and errors, and/or Significant utilisation of commands, utilities and programmatic techniques <u>not</u> addressed in the unit's lecture slides (Modules 1-5 inclusive) evident 	<ul style="list-style-type: none"> Script's code is properly structured at most, but not all stages, and/or In some, but not all cases, the script utilises appropriately selected commands, utilities and programmatic techniques applied correctly, efficiently, and in an error-free manner, and/or Very minor utilisation of commands, utilities and programmatic techniques <u>not</u> addressed in the unit's lecture slides (Modules 1-5 inclusive) evident 	<ul style="list-style-type: none"> Script's code is properly structured at all stages, and Script fully utilises appropriately selected commands, utilities and programmatic techniques applied correctly, efficiently, and in an entirely error-free manner, and Script only employs commands, utilities and programmatic techniques addressed in the unit's lecture slides (Modules 1-5 inclusive)

How to submit your portfolio to Canvas

Submit a **single** shell script (.sh) file named *parsechars.sh* contained within a **.zip** archive to Canvas with the following naming format (use *your* surname/student number):

[surname]_[student-ID]_CSI6203_PF1.zip



The screenshot shows the 'File upload' section of a Canvas assignment submission page. It includes a text prompt 'Upload a file, or choose a file you've already uploaded.', a 'Browse...' button (labeled with a red '1'), a file name 'brown_0555555_CSI6203_PF1.zip' (labeled with a red '2'), a '+ Add another file' link, a 'Comments...' text area, and 'Cancel' and 'Submit assignment' buttons (labeled with a red '3').

Do **not** submit any files other than that stipulated above. Further, even though there is no restriction on how many times you make individual submissions (each of which gets its own unique timestamp in Canvas), do **not** upload multiple files/zip archives in the *same upload action* as all will be considered invalid and not be marked.

END OF ASSIGNMENT BRIEF