

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 you receive.

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, **where it's suspected that a student is [representing AI output as their own work](#), they will 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/assessed in the *Azure Linux VM* provided at the beginning of semester. It is expected that you will develop and test your script in this Azure environment to ensure full compatibility.
- Your script **must** be a *bash* script and nothing else, i.e. the shebang line used is `#!/bin/bash`. Scripts of any other type, e.g. `#!/bin/sh` will receive a grade of zero (0).
- 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 *filter.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.
- *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 provided at the beginning of semester, only a **partial** mark may be awarded on a code-readthrough basis at the assessor's discretion. Your assessor will **not** fix non-functional, dysfunctional, or incompatible scripts.*

Task – Filter Records (20 marks)

Problem Description:

As you are considered the bash/shell script guru in your department at a tech sales firm, you have been asked to create a command line tool that will allow salespeople to quickly generate summaries of smartphone stock as listed in large .csv files sent out to them from a stock management system on a regular basis. Mostly, sales staff want to be able to quickly determine stock available based on the amount of RAM **and** storage capacity in-stock devices are equipped with. Presently, it is taking staff a very long time to scroll through these large .csv listings to identify those devices that are equipped with the RAM and storage capacity they're interested in.

Source File Example:

1	Smartphone	Brand	Model	RAM	Storage	Color	Final Price
2	Samsung Galaxy M23 5G	4/128GB	Azul Libre	Samsung	Galaxy M23	4,128,Blue	279.00
3	Motorola Moto G13	4/128GB	Azul Lavanda Libre	Motorola	Moto G13	4,128,Blue	179.01
4	Xiaomi Redmi Note 11S	6/128GB	Gris Libre	Xiaomi	Redmi Note 11S	8,128,Gray	279.99
5	Motorola Moto E32s	4/64GB	Gris Libre	Motorola	Moto E32s	4,64,Gray	148.52
6	Samsung Galaxy M23 5G	4/128GB	Verde Libre	Samsung	Galaxy M23	4,128,Green	279.00
7	Xiaomi Redmi Note 12 Pro	8/256GB	Gris Grafito Libre	Xiaomi	Redmi Note 12	8,256,Gray	329.99
8	Xiaomi Redmi Note 12 Pro	5G 8/256GB	Negro Libre	Xiaomi	Redmi Note 12	8,256,Black	382.99
9	Xiaomi Redmi Note 12 Pro	5G 8/256GB	Azul Libre	Xiaomi	Redmi Note 12	8,256,Blue	399.91
10	Xiaomi Redmi Note 12	4/128GB	Gris Libre	Xiaomi	Redmi Note 12	4,128,Gray	249.99
11	Apple iPhone 14	128GB	Medianoche Libre	Apple	iPhone 14	4,128,Black	1009.00
12	Xiaomi Redmi Note 12 Pro	8/256GB	Azul Glaciar Libre	Xiaomi	Redmi Note 12	8,256,Blue	249.26
13	Samsung Galaxy M13	4/64GB	Verde Libre	Samsung	Galaxy M13	4,64,Green	179.00
14	Xiaomi Redmi 10C	4/64GB	Gris Libre	Xiaomi	Redmi 10C	4,64,Gray	179.99
15	Xiaomi Redmi Note 11	4/128GB	Gris Grafito Libre	Xiaomi	Redmi Note 11	4,128,Gray	219.99
16	Apple iPhone 14 Pro Max	256GB	Negro Especial Libre	Apple	iPhone 14	8,256,Black	1599.00
17	Motorola Moto E20	2/32 GB	Gris Grafito Libre	Motorola	Moto E20	4,32,Gray	106.00

Important Points:

- The tool (script) you develop will be used by sales staff on the command line of their Linux workstations as this is the environment they spend most of their time in.
- Your script will be used throughout the organisation by many sales staff of varying skill and experience, so robust user input validation and informative user messaging are essential.
- Your script may well become someone else's responsibility to manage and extend upon in the future, therefore it must be properly and accurately documented to allow for this without any input from you as the original developer.
- The script must adhere to the organisation's coding guidelines (as addressed in the relevant lectures).
- Assume that the script will always be run in the same directory as the source .csv file the salesperson is using.

Required Script Functionality

- When your script is run, the user is to provide the required input data as command line arguments only. The user is **not** to be *prompted* for input at any point in the script's execution.
- The arguments provided by the user on the command line are to be fully validated **before** the script's main logic is allowed to execute. If any of the input arguments are invalid, the user is to be accurately *informed* of what the error is and how to correct it when they next attempt to run the script. The script must always terminate with an appropriate error code when an error is encountered.
- If all inputs pass the script's validation procedures, the script is to then provide filtered results structured and formatted **precisely** as shown in the example screenshot below.

```
$ ./filter.sh 4 64 source.csv
```

13 devices were found with 4GB RAM and 64GB Storage from:

Apple (1)
Motorola (2)
OPPO (1)
Samsung (2)
Xiaomi (7)

BRAND	MODEL	COLOUR	PRICE
Apple	iPhone 11	Black	\$589.00
Motorola	Moto E13	Black	\$119.00
Motorola	Moto E32s	Gray	\$148.52
OPPO	A17	Black	\$179.00
Samsung	Galaxy M13	Blue	\$179.00
Samsung	Galaxy M13	Green	\$179.00
Xiaomi	Redmi 10A	Gray	\$159.00
Xiaomi	Redmi 10C	Gray	\$179.99
Xiaomi	Redmi 12C	Gray	\$169.00
Xiaomi	Redmi 12C	Green	\$169.00
Xiaomi	Redmi 9C	Green	\$159.00
Xiaomi	Redmi Note 12	Blue	\$229.00
Xiaomi	Redmi Note 12	Gray	\$229.00

- Perhaps needless to say, the results **must** be fully *correct* and *complete* in the context of the command line arguments provided.

Other Compulsory Requirements

- Call the script `filter.sh`.
- Your full name and student number **must** be placed at the top of your script (as comments) immediately after the *shebang* line.
- You may **only** use commands, utilities and programmatic techniques addressed in **Lectures 1-5 inclusive** with the exception of `awk`, `sed` and `functions`. `Awk`, `sed` and `functions` are **not** to be used until the Portfolio 2 assignment onwards.
- Do **not** use bash/shell script commands, utilities and programmatic techniques not addressed in Lectures 1-5 inclusive or excluded above as this will result in a substantially lower mark being awarded (see rubric for details).
- Your script **must** be properly documented, accurately explaining **all** of the code elements it contains in your own words. *Note: Documentation (comments) that is not relevant and accurate in regard to the code they describe, or a complete lack of documentation, will not only attract a lower mark, but may be viewed as suggestive of possible academic misconduct.*
- **AI-generated documentation (comments) will receive a mark of zero (0) in this section of the rubric**
- Any *temporary* files and/or folders created by your script in the course of its execution **must** be programmatically removed from the assessor's system when the 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 *filter.sh* script is expected to make use of **appropriately selected** and **correctly applied** commands/options from *Lectures 1-5* inclusive (except those specifically excluded above). 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 a file named *source.csv*. Examine this file carefully. *Important Note:* Your assessor will assess your script using a .csv file that has precisely the same structure as *source.csv*, but that will contain values taken from an entirely different section of the master data set from which those in *source.csv* were drawn. The assessor's .csv file will also not be named *source.csv*, but rather, something else. Hence, do not hard code the file's values or name into your script.

ASSESSMENT RUBRIC

Go to *Assignment 2.1: Portfolio 1* on Canvas to see the applicable rubric for this assessment.

How to submit your portfolio to Canvas

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

[surname]_[student-ID]_CSP2101_Pf1.zip

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