# Groups

**#**  **Members**

1 Shakhram, Hassan, Haseeb, Adina, Tian Long, Adnaan, Andrew C.

2 Huang, Derek, Mohammed, Angela, Rafid, Zakia, James

3 Huimin, Jiao Jiao, Naldy, Jinwei, Armando, Jiayi, Kamille

4 Kevin, Bo Hui, Xinjin, Jay, Stephen, Tianye, Jackson

5 Tahamina, Tenzin, Andrew L., David, Kristy

6 Fernando, Irina, Joseph, Muhammad, Raja, Rei, Youssef

**Overview:**

* Evaluate each project’s documentation, and source code. See following pages as example criteria to deepen your analysis/evaluation process.

# Suggested Criteria Documentation Review

* Project Description, i.e. a README
  + Description of features included with the packaged software, i.e. a CHANGELOG or RELEASE NOTES

I though the description of the project was adequate considering it was a small project.

* + Instructions on how to run the program

The instructions on running the project were clear.

* + Description of technologies used is provided

A short description of AWK was provided.

* Organized has clear File Management
  + Every file included in the repository has a sensible filename and extension.

I think the file names and folders are adequate. They would not be good for industry but they are good for college.

* + There are instructions on contributing to the project repository.

There are no instructions on contributing to the repository. Since this was a college project it would not make sense to include that.

* Source Code is readable.
  + Source code comments describe why each class/function/block exists, as well as expected input parameters and outputs.

The sone comment is readable because it is a simple project. You cannot really tell what the code is for when reading it but since the code is small and that info is the the README.MD I think it is fine.

* + Code is readable, such as has uniform indentation styling, sensible comments, sensible variable names, and is cleanly presented with an organization you can follow along while you read it.

The code is pretty readable.

* Sample Runs and/or Test Case(s) provided to show examples of using the program with expected outputs

No test cases are provided. I think this is fine as no user interaction is involved.

* If test suite provided - each test case has a description of what it is about, what inputs or outputs mean, etc.

No test cases provided.

# Suggested Criteria for Code Review

**System / Environment**

* The operating system you’re testing this on. Does the code/documentation tell you what operating system to use? If doesn’t, what should it state?

I used the College provided Linux servers.

The documentation does mention Linus as well as dependencies. A more explicit reference of where to run it would be nice.

**Software Design**

* Is it clear what this project is about? How do you know what the project is about? Is there any documentation or comments with code to understand what’s happening? Describe the purpose of the project in your own words.

From the code alone it is not entirely clear what the project is about. The documentation makes this more clear though. There is only one comment to describe what is going on but the README.MD does describe what the project does.

The purpose of the project is to read in data about cars from a car show. Find out how well each car did and output a list of the cars and their relevant info in order of most points to least points.

* Does the project allow you to input any parameters to change report settings? For example, can you change the reporting from monthly to weekly? Are there any instructions on setting parameters?

No there is no interactivity with the project. Since it is a simple report generator, I think this is fine.

* Describe in your own words how this program is to be used. For example, do you click to open the program? Or run a script?

This program is a script program that makes use of Linux and text files.

Personally, if I wanted to perform this, I would use a real programming language like Java or Python. The greater number of features that would be available as well as the larger amount of good documentation would make it easy to make the program and allow more easily to add different functionality.

**Software Implementation**

* What actions is the project author suggesting you take when you use code provided? If this information does not appear, describe where you looked for this kind of information.

The author asks that you run a simple make command. What the author wants you to do can be found in the README.MD file.

* + Is it clear what action you can take or should take next?

Yes, it is clear as there is only one command.

* + Describe what helps you figure out what you’re supposed to do next; or describe what could help you figure out what to do.

There is not really a second step to do. You run the code and look at the results.

* Does this project work as is or do you have to install anything additional to get the project working?

The project did not work for me as is. I had to add the data submodule separately. Perhaps this was an error on my part though due to my unfamiliarity with Git and Linux.

* + What did you have to install and how did you go about it? What instructions were provided for you to figure out project dependencies?

I had the install the data submodule. I found this out as when I ran the code it said the data did not exist. Instructions for dependencies were provided but not for adding the submodule. I do not take fault for that though as one would not think it would be an issue.

* + What language/tools/applications did you use to run the program provided? Describe any dependencies needed, and if they come with your machine.

I simply used the College Linux server.

**Data Analysis Evaluation**

* What report is supposed to be produced from this repository? Describe in your own words.

The report is to list relevant car data in descending order based on the car’s total score/

* Does the generated report meet the description in the previous bullet point? If not, explain what is missing.

Yes and no.

For the most part the report is in order but a strange thing happens towards the end of the report where it is no longer in proper order. Also, some of the scores differ from what I calculated.

I remember having a similar problem in my homework but I was able to fix it.

This error might be caused by the data changing and not the code being faulty though.

|  |  |  |
| --- | --- | --- |
| **Checklist Item** | **Included?** | **Notes** |
| README.md | Y | The README.MD explains how to run the program aswell as what it does and what is expected. |
| CHANGELOG or RELEASE NOTES | N | No Changelog or release notes are included. Since this was the first project I guess it is fine to do with out. |
| Instructions on how to run the program | Y | The instructions to run are simple which is just the make command followed by the specific pattern for this project.  It gets the job done. |
| Descriptions of technologies used and provided | Y | A short description of AWK is given. |
| Every file in the project has a sensible file name and extension | Y | Considering the homework nature of the code I beleive the files are appropriately named. |
| Source code includes comments. Comments describe why each class/function block of code exists as well as expected input parameters and outputs | Y | There is only one comment that explains the variable we are making is the total for each car. Some more comments could have been helpful. But considering how simple it is maybe we don’t need a lot of comments. Overdocumentation is also bad. |
| Code is readable, has uniform indentation styling, sensible variable names, and is cleanly presented. | Y | The code is indednted and sensible. |
| Test Case(s) are provided to show examples of using the program with expected outputs | N | No test cases were provided but because the code does not involve user input I beleive that is fine. |
| Each test case has a description of what it is about, what inputs or outputs mean, etc. | N | No test cases were provided |

**Other Comments**

**Issues**

* + When running the code directly after cloning the repo I was not able to get the code to run. For some reason the data submodule did not get cloned. To fix this I had to manually add the data submodule.
  + Some of the total score values differ from the ones that I got in my project
  + Towards the end of the list the sorted data is no longer sorted. There could be a few reasons for this. The first is that the code was made for an earlier version of the data and so it is no longer compatible. I remember having a similar issue with my code but I fixed it. I cannot remember what the problem was. I will see how my code handles it and if the same error occurs.
  + Some of the documentation is incorrect but this happened because it was originally written for a single student’s repo not 3 students.