# 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:**

* Each group will be provided projects to evaluate. Projects can be found at the link provided.  
  URL: <https://github.com/katychuang/cisc3140_lab3>
* Evaluate each project’s documentation, and source code. See following pages as example criteria to deepen your analysis/evaluation process.
* Modify each project with the proposed changes (see ‘New Features’ listed below). Each ‘new feature’ should apply to each sub project (proj1, proj2, etc).
* Prepare group presentation on your experience. Groups will present on 4/6, 4/11, 4/13 (in ascending order of the group numbers)
  + Suggested presentation length is 5 minutes per member in the group (i.e., roughly half hour per group)

**New features:**

1. In addition to providing a list of top three cars overall and also top three per make, find the top 3 ranked for new classes.
   1. The new classes: Racer, Engine, Body\_Frame, Mods, Mods\_Overall, Car\_Overall.
   2. List any of the top 3 ranks with rank numbers: 1, 2, 3.
   3. Ignore show cars.
2. Enable options during execution that would allow customize the report when running the provided scripts.
   1. Your team should come up with appropriate and/or useful options. For example this may be file format output, parsing configurations, customization of fields presented in the output(s), and other information.
3. Clean up the repository before submitting
   1. Update the build provisions (i.e., makefile) in the root folder of the repository and ensure that instructions are provided somewhere. If your team votes to use shell scripts instead, that is fine.
   2. Prepare a README file for the entire repository.
   3. Update each of the individual READMEs inside each subfolder as appropriate. Please use proper markdown formatting for markdown files.

**Submission Instructions:**

Each student should submit to the Google Form the pull request URL for their team. Pull request can be 1 per team (in comments of pull request, include group number and member names).

# Presentation Prompt Ideas

* Describe briefly how you or your team organized the review process. Did you rank the projects in a certain order before beginning your review process? What approach did you use to process the evaluation process (i.e. did you evaluate first come first out or alphabetized, etc.)

For the most part each group member decided which project they would work on and which aspect of the project they would work on. Some other members and myself thought it would be useful to do all that was asked for one project so that we could have experience with all the tasks. The evaluation process was mostly first come first served.

* Of the collection of projects you reviewed, which one would you say is the highest quality? Describe what elements to you indicate it is high quality. For example, was it well documented and/or organized? What made it seem well documented?

All of the code seemed to have issues and to be pretty similar but of them I would say that Project1 was probably the best even though it had some bugs. While not well commented I honestly think considering the small scale of the code given I think heavy documentation was not needed.

* Were there any bugs apparent with any of the projects you reviewed?

Yes, I encountered two bugs.

* + Briefly describe the nature of bugs. For example, was it operating system compatibility?

The first bug had to do with running the program. For some reason after cloning the repo and trying to run the code I was getting an error that the data csv file was not there. To fix this I had to manually add the data submodule. I am not sure why this bug occurred but it is certainly not the fault of the coder.

The second bug I encountered was that towards the end of the sorted list the data stops being sorted and we gt entries we should not expect to find.

This remined me of a bug I had in my own lab1 but I was able to fix. Unfortunately, I do not remember the fix that I used.

Also, the total scores differed from my total scores.

* + Was the program incomplete?

I would say that the program was complete for what the documentation says that it does aside for the bugs.

* + Were the instructions difficult to follow or missing?

The instructions were simple and easy to follow.

* Is there anything you saw with the projects you evaluated that you liked, didn’t use for yours and will adopt for your future version-controlled projects?

I thought the inclusion of downloading dependencies was a good addition. But I also think that it might not be needed and might overburden someone without knowledge of what to do. For example, for those of us using the College Linux servers we are not able to download dependencies and trying those instructions would lead to failure which might bring about confusion.

* What were preparatory steps you had to take before you can begin modifying source code? Was it immediately clear how to modify the code? What do you think aided your understanding? If not, why not and what could be improved?

Before changing the code, I had to look at the code, run the code, see the output and compare it to the expected output. I also needed to look at the data and how it has changed since the first lab. Since there was not a lot of code it was pretty clear what needed to be done.

* Was collaboration as a group going well or not? Why or why not? How did your team divide up the work? Did it meet your initial expectations, did it change over time? Why or why not?

I would say that our group’s collaboration went well. We seemed to have members motivated to actually try and understand the tasks being asked of us as opposed to pushing out the bare minimum. We divided the work individually with people saying what they would like to work on.

* + What tools or processes helped with collaboration? What didn’t?

Discord was certainly a useful tool. I did not find the classroom group meetings to be helpful.

* + Was it preferrable to you to work alone or pair program? Or something else? Why or why not?

Personally, I found it preferable to work alone. For relatively small projects I do think pair program can hamper work flow. Also considering the lack of designated roles and tasks made it easier to work alone.

**Thoughts on completed code**

While a good attempt I do think there is room for improvement in my work.

For example, it would be nice to create a printCarData function so that I only need to call a function to print the data instead of writing the whole thing out.

Also, more functionality could be added like a way to get user interaction without asking them to change the makefile.

Also, it would be interesting to see if there is a way to use regex on column names when getting the scores.

A better IDE would also help with making the code more presentable as opposed to using a text editor.

Code

https://github.com/HA-work/CISC3140Lab3PRoj1