**Part 2: GitHub Version Control (10 marks)**

**Question 1 (6 marks)**: Often, it makes sense to create a branch of a repository to work on that is separate from the original codebase. Write the commands into a text file or take screenshots as you perform the following actions to create a Git branch:

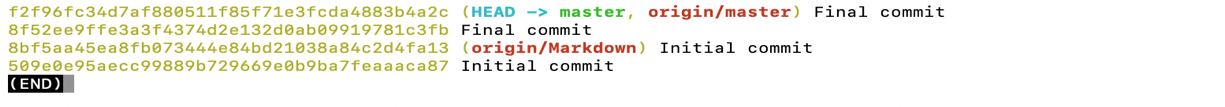
1. Clone your Lab 3 repository to your local machine (1 mark)
   * git clone <https://github.com/ubco-mds-2021-labs/data-541-lab-3-MayukhaB.git>
2. Navigate into the cloned repository folder
   * cd data-541-lab-3-MayukhaB
3. Create a new Git branch called **Markdown** (1 mark)
   * git branch Markdown
4. Use checkout to move to the **Markdown** branch. You may use the -b flag to create and checkout a branch at the same time. (1 mark)
   * git branch -b Markdown
5. Check the status of the current repository and confirm you are on the **Markdown** branch (1 mark)
   * git status
6. Add and commit your lab3\_q1.md and lab3\_q2.md files to the current branch. (1 mark)
   * cp /Users/mayukha/Downloads/lab3\_q1.md /Users/mayukha/data-541-lab-3-MayukhaB/lab3\_q1.md
   * cp /Users/mayukha/Downloads/lab3\_q2.md /Users/mayukha/data-541-lab-3-MayukhaB/lab3\_q2.md
   * git add -A
   * git commit -m "Initial commit"
7. Use the push command with the following flags -u origin Markdown to push the **Markdown** branch to GitHub. (1 mark)
   * git push -u origin Markdown
8. On GitHub, you should see the **Markdown** branch in the Branch dropdown

Graphical user interface, application

Description automatically generated

**Question 2 (4 marks)**: Once the desired work has been completed on a branch, you can then merge that branch into the **master** branch. Write the commands into the same text file as before, or take screenshots as you perform the following Git merge:

1. Use checkout to move back to the master branch of your Git repository. (1 mark)
   * git checkout master
2. Merge your **Markdown** branch into your **master** branch. The syntax for the merge command is git merge branchName. (1 mark)
   * git merge Markdown
3. Perform a push to push the merge up to GitHub.
   * git push
4. Delete your **Markdown** branch - now that it is merged, it is no longer necessary. (1 mark)
   * git branch -d Markdown
5. Use the git log function to print a short log of your actions. If the -oneline flag doesn't work, try using --pretty=oneline instead. Take a screenshot of the output. (1 mark)



1. Add and commit your text files with your commands and/or screenshot(s) to the **master** branch of your lab repo and push. Your lab repo **master** branch should now contain your Markdown files from Part 1 of this lab, plus your text files and screenshots from Part 2. Submit the link to your repo on Canvas.
   * git add -A
   * git commit -m “Final commit”
   * git push