Git Lab – Branching & Merging

Objective

The objective for this exercise is to become familiar with branching and merging operations with the git command line tool.

Steps

1. Create an empty local Git empty repository

git init

2. Create some simple text files (e.g. file1.txt, file2.txt, file3.txt) and commit them to the repository (on the main branch)

    e.g. use notepad or VsCode to create your txt files

    git status

    git add .

    git status

    git commit -m "your comment"

    git status

3. Create a new branch named req1, representing changes you need to make to address some imaginary requirements (req1)

    git checkout -b req1

    git status

4. Now you should be on the req1 branch, though the files should currently be the same as they were on the main branch.
5. Make some changes to some of the files and commit these changes. These changes will be added to the req1 branch

    e.g. using notepad or VsCode make some changes to your files

    git add .

    git commit -m "your comment"

    git status

6. Revert to the main branch, and verify the files in your working directory have reverted back to the original versions prior to your most recent changes.

    i.e. Check that the main branch still has the old versions of the files.

7. Now make some other changes to the files and commit these changes. These changes will be added to the main branch

    e.g. using notepad or VsCode make some changes to your files

    git add .

    git commit -m "your comment"

    git status

8. Now try switching back and forth between the branches with the files open in VsCode.

    Notice how the contents of the files changes to the current branch.

    git checkout main

    < take a look at your files >

    git checkout req1

    < take a look at your files again >

9. Now try manually "merging" the req1 branch into the master

    git checkout master

    git merge req1

10. If you get any "CONFLICT" messages then open the files in a text editor and fix them.

    To do this, look for any lines containing:

    '<<<<<<<<<<<<<<', '===============' or '>>>>>>>>>>>>>'.

    Fix these lines so they contain the content you want.

11. Once you're happy with the contents of your files you need to add and commit them before the merge is complete.

    git add .

    git commit -m "Merged req1 branch into master"

    git status

12. Notice how all the commits are all now on the main branch, including those made on the req1 branch.

    git log