#19_Assignment_Calculator_Bonus_Features

01/06/202

Git Branching

- Allows you to amend your code without affecting the original code. Effective for when adding new features.

How to use git branches

- Type git status into Terminal

```
% git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
```

If the above does not occur, likely means need to add and commit

```
% git add .
% git commit -m 'insert your message here'
```

Creating a New Branch

02/06/2025

- Type git checkout -b development into the console
  - this creates a new branch named 'development' and switches to it.
- Any changes made to this branch will not affect the main branch.
- Can switch back to main by typing git checkout main
- Type git branch -d development to delete the development branch.

02/06/2025

Merging Changes back to main

- Merging incorporates any changes made in one branch to another branch.

Step 1

- Commit changes to the development branch.

```
% git add .
% git commit -m 'insert your commit message here'
```

Step 2

- Switch back to main branch

```
% git checkout main
```

Step 3

- Merge branches and incorporate new features from the development branch to the main branch.

```
% git merge development
Auto merging 'filename'
Merge made by the 'recursive' strategy
 filename | 4 +++
 1 file changed, 3 insertions (+), 1 deletion (-)
```

- Now main branch contains the same code as the development branch. Ready to push it to Github.

Step 4

- After merging development to main branch, you can delete development branch

```
% git branch -d development
```

02/06/2025

Conclusion

- Can open as many branches as you like
- Different branches for different features

Cheat Sheet

- git add . #Stage your files, preparing them for the next commit.
- git commit -m 'your message here' #Commit your changes, creating a new snapshot of your code.
- git checkout -b development # Creates a new development branch and switches to it. This is shorthand for the following two commands
- git branch development # Creates a new branch named development
- git checkout development # Switches you over to the development branch
- git checkout main # Switches you back to the main branch
- git merge development # Merge the changes from development to the current branch
- git branch -d development # Delete the development branch
- git branch -D development # Force delete the development branch
- git status # Shows you the status of the current branch.

A branch is not a copy of files but a reference to a commit

```
% git log --graph --oneline --all
  # allows you to view history
```