DOC1 Exercise(s) 1

Exercise 1.1 – Git (local, using the command line)

For this exercise, we are assuming you have installed Git Bash or you are on a system with a Bash-like command shell. So, the 3 steps below should be carried out using the Git bash shell:

a) Create a new directory and change to that directory.
b) Initialize the directory as a Git repository using the git init command.
c) Create the first commit by adding some (at least 3) files (use something like VS Code or Notepad(++) to create some simple text files in the directory) to the project. For each file you create during this exercise use this sequence of commands to see how your local Git repository changes
   a. git status
   b. git add <name of text file>
   c. git commit
   d. git status

Exercise 1.2 – Git (local, using IntelliJ)

For this exercise, you'll create a new Java project with a Git repository.

a) Create a new Java project in IntelliJ (it could be a Spring boot Project or just a simple project with a single Main class containing psvm). Remember to create a Git repository for the new project when you create it. To have a few files to work with you can create a simple dummy class with a single private member and accompanying getter/setter.
b) Once your little project is compiling/running in step a, check-in your initial code by clicking on the "Commit"-icon to the left and take a careful look at the files IntelliJ suggests you add to the Git repo. What does the two files .gitignore and .idea/.gitignore do? Why do you think IntelliJ creates those files automatically?

Exercise 1.3 – Git Branching and Merging

For this exercise, you'll rename and create a new branch in the project we created in exercise 1.2.

a) Rename branch master -> main (if it is not already called main).
b) Create 3 small classes (feel free to create 3 controller classes or whatever). For each class:
   i. Create a new branch called dev1/dev2/dev3, respectively
   ii. Create a single class
   iii. Make a happy test of the class (in psvm)
   iv. Add/commit changes to current branch (both via IntelliJ and commandline)
   v. Do some poking around via Powershell/Bash shell to see Git status, view git log
   vi. Merge new branch into main
   vii. Delete the dev[1,2,3] branch

Exercise 1.4 – Extracting the project's Git log

You need to export the Git log for your project and submit it to the itsLearning assignment called "*Upload the Git log for exercise 1*". Follow these steps:

a) Open Powershell or Git Bash in the root folder of your project.
b) On the command line write: **`git log > gitlog.txt`**
c) Inspect, using VS Code/Notepad(++) for that, the file (gitlog.txt) and upload it.