JC2002 Java Programming - Practical 2 (Day 2)

This practical has two parts: Java programming part (tasks 1-2), and Git part (tasks 3-5). The goal of the first part is to get familiar with the basic Java syntax and structures in practice.

1. Write a simple Java program that takes two integers as user input (**start** and **end**). Then, write a loop that runs from **start** to **end**, and computes the sum of all the integers from **start** to **end**. For example, if user gives value 5 for **start** and value 10 for **end**, the result would be 5+6+7+8+9+10=45. After running the loop, the program should print the result in the console. Try to use the program with different inputs to make sure it works correctly.
2. Add a conditional statement in your code to check if **end** is larger or equal to **start**. If **start** is larger than **end**, the program should display an error message instead of computing the result. Try to use the program with different inputs to make sure it works correctly.

The goal of the second part is to learn to use Git, both on your own and with a partner, so that you know how to work together on a project with Git. The Git part is divided into three tasks. Do task 3 on your own, and then find a partner for the tasks 4 and 5.

3. Do the practicals at https://learngitbranching.js.org on your own. These online interactive exercises guide you through the basics of Git so that you're comfortable using Git on the command line. They also have good graphics to clarify what is happening in the background with your code. do not spend more than 20-30 minutes max, you can come back later to finish the exercises on your own time.
4. Next, find a partner for the following tasks. Start a repository in GitHub with a **'Readme.md'** file with your name, and a Java application in another file. You can use the java file you created in tasks 1-2. After you commit this to GitHub, then add your partner to the repository as a collaborator so that they can clone it and push it to the local repository.

   Now, you and your partner should each add a branch and modify your Java code (for example to add some new feature), and then merge these changes into the main branch. As you each do your work, add notes to the **'Readme.md'** file outlining the features you are adding.
5. Commit a change to a file and push it to the repository. Then, ask your partner to change the same line of code in their local version of that file (do not pull the first changes) and try to commit. You should see a warning that there is a conflict. You need to open the file with a conflict in a text editor and resolve the conflict manually by deleting the conflict markers. Have a look here what to look for: https://imagej.net/develop/git/conflicts.