**Configuration**

**In Visual Studio Code:**

1. Check that Git is enabled
   1. From Side Bar, open **Manage🡪Settings** or type **Ctrl+,**
   2. Type “git enable” in Search settings and press Enter
   3. Make sure that Git:Enabled is checked
2. Check username and email configuration in Git
   1. Open the **Terminal** and issue the following commands:git config user.name  
      git config user.email
   2. If either or both are missing/incorrect, issue the following commands:  
      git config --global user.name “Your Name”  
      git config --global user.email “Your Email”
3. Check that you are logged into GitHub
   1. From Side Bar, open **Accounts**
   2. If you do not see your GitHub userid,   
      select **Turn on Cloud Changes…**then, in the popup menu, select **Sign in with GitHub**

**Group Development with Branches**

**Scenario:**

* Group is comprised of 3 members: A, B, and C
* B is nominated to create the group's Git repository and publish it to GitHub
* Group members each work on their designated branch
* When ready, group member branches merged into the main branch

**On B's computer:**

1. Create a new folder for the group project (using VS Code or File Explorer)

**In Visual Studio Code:**

1. Choose **File🡪Open Folder…** and select the folder that was just created. This empty folder is now B’s workspace
2. From Side Bar, select **Explorer**
3. Create your initial Python files (e.g. containing your Patient class and patient management application)
4. From Side Bar, select **Source Control** and click on **Initialize Repository**
5. Add the files to Git staging by clicking on the **'+'** icon beside the file names – each file’s status will change from U to A (untracked to added)
6. **Type in a brief message** and click the **Commit** button to commit staged changes to the Git repository
7. Now click the **Publish Branch** button to publish the repository to B's GitHub account. Choose **'Publish to GitHub private repository'** when prompted to choose between a private or public repository.

**In GitHub (signed on to B's account):**

1. Select/open the newly published group repository
2. Click on **Settings**, then **Collaborators** (underneath the **Access** section)
3. Click the **Add people** button, search for A's GitHub username or email, and select the correct match to be added as a collaborator on the group repository
4. Repeat the previous step to add C as a collaborator, then move on to the next step
5. Navigate back to the main page of the group repository and click on **'1 branch'** to view branch details
6. Click **New branch** (green button) and **type in a name** for the branch that A will work on
7. Click **New branch** again and **type in a name** for the branch that B will work on
8. Click **New branch** again and **type in a name** for the branch that C will work on
9. Return to the main page of the group repository and click on the green **<> Code** drop-down menu/button
10. Now click the **copy icon** to copy the HTTPS repository URL to the clipboard
11. Send this URL to A and C (it should look something like [https://github.com/<B>/<repo-name>.git](https://github.com/%3cB%3e/%3crepo-name%3e.git)

**On A's and C's computers:**

**In Visual Studio Code:**

1. Close any open folder(s) using **File🡪Close Folder**
2. From Side Bar, select **Explorer** or **Source Control**
3. Click the **Clone Repository** button
4. In the “Provide repository URL” input box, enter the **group repository URL** and press Enter
5. A dialog will now open requesting you to select the parent folder under which your project/group repository folder will be created (e.g. you could select C:\CPRG216S)
6. Open the newly cloned repository
7. Current branch will initially be set to the default branch of master (or main)
8. From Side Bar, select **Source Control**
9. Click on **...** (top right of the navigation window) and choose **Checkout to...**
10. Select the remote branch that was created for this group member (A / C)
11. Group member can now develop on their specific branch. Changes should periodically be committed to their local repository and pushed to GitHub by clicking on **Sync Changes.**

**On B's computer:**

**In Visual Studio Code:**

1. Ensure that the project/group repository folder is open
2. From Side Bar, select **Source Control**
3. Click on **...** (top right of the navigation window) and choose **Fetch**
4. Current branch will likely be set to the default branch of master (or main)
5. Click on **...** again and choose **Checkout to...**
6. Select the remote branch that was created for group member B
7. Group member B can now develop on their specific branch. Changes should periodically be committed to their local repository and pushed to GitHub by clicking on **Sync Changes.**

**On any group member's computer:**

To **merge** the work done on the group member’s branch into the master (or main) branch

**In Visual Studio Code:**

1. Ensure that the project/group repository folder is open
2. From Side Bar, select **Source Control**
3. Check that you are in the group member’s specific branch (see very bottom left of screen). If not, choose **Checkout to...** and select the correct branch
4. Add any new/updated files to Git staging by clicking on the **'+'** icon beside the file names
5. **Type in a brief message** and click the **Commit** button to commit staged changes to the Git repository
6. Click on **...** and choose **Checkout to...**
7. Select the master (or main) branch
8. Click on **...** again and choose **Branch🡪Merge Branch…**
9. Select the group member’s branch that you want to merge into master (or main)
10. If there are any conflicts, you will need to resolve them before going to the next step
11. Click on **Sync Changes** to push the changes to GitHub