**Practical Test Preparation:**

There is a practical test on Session 7. Instructions will be given on the test paper. The practical test will be similar with the following tasks:

1. Create repository in GitHub, loaded the README.md, LICENSE.md & .gitignore files.
2. Set up the private and public key pairs.
3. Clone the remote repository. Load the VS project and develop using branches.
4. After committing to the local repository, use Git Syn to push the local changes to the remote repository.
5. Do further development based on a branch and at the end merge back to the local master branch.
6. Create software patch from one revision to another.

The above tasks are implemented as the following steps:

* Create repository in GitHub. Make sure that you change the GitHub with the master branch as the default branch. i.e. Click **setting** 🡪 enter **master** and click **update** button. It will be effective when you create a new repository.
* In general you need to add the README.md, LICENSE.md, & .gitignore files in the repository. At the time of creating a new repository, you can add an empty file or pick a selected file from the drop down list and later change the content to what you wish. ( in the practical test, you will be provided the content of these files).
* Use PuttyGen to generate the private key and public key pairs.
* Add public key to the GitHub.
* Use TortoiseGit, clone the remote repository to the local working folder.
* Initial loading of a VS project (use the **Ticketing** application in the student files) into the local repository and commit to the local master branch.
* Find out the hash index of the current version.
* Do Git push and syn the changes to the remote repository.
* Use the show log before and after the changes.
* Modify VS the application (e.g. change the ticket price 1st round) and commit the changes and push to the remote repository.
* Clone the remote repository to the local working folder.
* Create a branch for development based on the head revision of the master branch.
* Checkout the local branch. Modify the code (e.g. change the ticket price 2nd round) and commit the changes.
* Push the local changes to the remote repository.
* Create a new working folder and clone the remote repository. Check out the previous branch and do further development. Based on the previous branch to create a new branch. Checkout the new branch modify code (e.g. Give a further 10% discount to all ticket types) on the new branch. Commit the new branch to the local repository. Use Git Push to syn to the remote repository.
* Checkout the master branch. Choose Tortoise Merge - **from** the new development branch to the master branch.
* Do Git Sync and push to the remote repository.
* Use the show log to find out the revision numbers and **create patch serial** from one revision to another.
* You should be able to see the patch file(s).
* You may create another folder and checkout the version after you have committed the .gitignore file. Use the notepad to open the patch 0001\_xxx.patch. Read the from revision no. Record the revision no. In Show Log, locate the revision no.
* Try out apply the serial of patch files one by one.