**Exercise – 1**

1. Open Spring Tool Suite
2. Choose Window-Preferences-version Control -> git->Configuration
3. Select User Setting Tab
4. Select Add Entry
5. Add user.name
6. Add user.email

**Exercise – 2**

1. Create a Java Project
2. Create Some Classes and Save it
3. Right Click of the Project Select Team->Share Project
4. Choose Create Button and Enter the Name for Repository

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Click on finish
2. In the Project Explorer -look for Symbol “>” and project Name appended with [repositoryName]
3. Select Window->ShowView -> Other-Git -> Git Repository
4. Look At the repository and explore the various options

**Exercise – 3**

1. Right Click on the Project -Select Team Commit
2. Open a Git Staging window
3. Click on the ++ Button in the unstaged Area
4. Add a Commit Message
5. Click on the commit button
6. Open the repository view and look for the Master Branch

**Exercise – 4**

1. Select Window->Show View->Version Control ->History
2. In the History View Choose the Filter Icon
3. Choose All changes in the Repository
4. Look at the various History links provided and understand

Graphical user interface, text, application

Description automatically generated

1. Select Window-Show View -other->Git->Git Ref Log and view the same

Graphical user interface, text, application

Description automatically generated

**Exercise – 5**

1. Open the Repository : <https://github.com/vatsanTraining/gitexercise>
2. Click on Green Button with the text => Code use drop down
3. Copy the Https URL it will be copied
4. Choose File->Import->Git-> Project from git With Smart Import
5. Select CloneURI
6. Values in the text field will be pre-populated click on next

Graphical user interface, text, application

Description automatically generated

1. Choose the Branch typically Master Branch

Graphical user interface, text, application

Description automatically generated

1. Accept default option and Finish
2. Project comes into local repository and close into Eclipse Workspace

**Exercise – 6**

1. Create a GitHub Account , If you don’t have one
2. Create a Repository with a name
3. Copy the https URL
4. Right Click on the Project Select Team>Push-Remote give the remote project name
5. In the GitHub portal choose settings below your profile
6. Select Developer Settings -> personal access token
7. Click Generate token -> give a name and select repo -> generate token
8. Copy the generated token and save it in a text file
9. Push the Project to the repository by adding the required details and pasting the generated password

Exercise -7

1. Access the following GitHub Repository <https://github.com/vatsanTraining/training_sept_2021>
2. Click on the Fork Button
3. After Successful cloning
4. Import the project to Workspace

**Exercise -8**

1. Right Click Select Team->Branch->New Branch
2. Give a name for the branch
3. A new branch is created, and you move to that branch
4. Team->Advanced ->Delete Branch => Delete the branch
5. Repeat step 2 and create a new branch
6. Add some code to the project and commit the changes
7. Right Click Project and 7Switch to master Branch
8. Right Click Select Team -> Merge choose the branch observe the code is merged
9. Right Click on the Project Select Team -Push Confirm Configured Remote Repository
10. Click Next Button

Graphical user interface, text, application

Description automatically generated

1. Select Add Spec choose the Branch and Select Force Update Finish

**Exercise -9**

1. **Login to GitHub Account**
2. Navigate to the main page of the repository.
3. Under repository name, select   **Settings**.
4. Select Manage Access Add the users
5. **Login with other user and clone the project to workspace**
6. **Make the changes and commit and push the changes**
7. **Make a Pull Request from github account and merge the changes**

**Exercise:10**

**Person 1 will Host a Project in github**

**Person2 Let the other user fork the repo**

**Make the changes in the code**

**Make a Pull request**

**Person1 will review the code change and merge the changes to original**

**Merge pull requests**

Graphical user interface, application

Description automatically generated

1. **Add the People by Name**
2. **Invitation will be sent to them by email after they accept**
3. **Let them clone the project**
4. **The Collaborator will create a branch**
5. **Add /Modify the files in the workspace**
6. **Commit and Push the changes**
7. **Navigate to Github and make pull request and merge the changes**
8. **Look at the updated repository with the new changes**