

### **Lab 3: Source Code Control**

**Devayan Mandal, Ikhlas Ahmed**

**Ques 1:** Even if working independently on a software project, describe briefly why a source code control system would be helpful to manage development. Consider both multiple development environments (eg, a portable computer as well as an office desktop computer, both used in development) and the need for revision history.

**Ans:**

Source code control systems prove to be very helpful while editing text, code, documents or a program.

1. The source code control system helps keep track of changes made to the file. The changes can be imported or exported to other environments, such as a remote desktop or portable device as desired.
2. Source code control systems provide a versioning system. This provides multiple functionality:
  - a. Allows for retrieval of the file in the situation of a system crash.
  - b. Allows for retrieval of the previous version of the file if the present version gets corrupted.
  - c. Allows for retrieval of specific files if they were deleted accidentally.
3. The source code control system allows for effective file sharing and co-ordination if several individuals are working on the same file or document.

**Ques 2:** Investigate and briefly describe the purpose and functionality of each

**Ans:**

1. Watch:  
Watching a repository, notifies us about:
  - New pull requests made.
  - Issues created in the repository (even those which did not have to do with us directly).
2. Star:  
This options allows us to keep track of repositories we find interesting.
  - The Stars page displays all repositories we are current watching.
  - We can star/unstar a repository directly from the navigation pane.
3. Fork:  
This option allows us to make a copy of the file, document or project into our repository. Forking allows to edit the original copy with affecting the original version of the repository.

**Ques 3:** Based on the contents of this file, briefly summarize the purpose of the project.

Ans:

The project chronicles information about Bootstrap which is a front-end framework for website development. Bootstrap was the initial blueprint for Twitter. It was created by Mark Otto and Jacob Thornton. Ever since its initial release, Bootstrap has gone through several version revisions and changes. It serves as a style guide for internal website tools development.

**Ques 4:** How many changed files are involved? How many additions and deletions were committed?

Ans:

1. Changed files: 1
2. Additions: 7
3. Deletions: 2

**Ques 5:** What color coding is used to indicate additions and deletions to the files. (4 points)

Ans:

Red gems or '+': insertions

Green gems or '-': deletions

(I could be wrong about the colors because I'm color-blind. I'll rely more on the +/- symbols)

**Ques 6:** Briefly describe the purpose of issues and pull requests, and how they differ. (4 points)

Ans:

Issues represent the questions, concerns and comments regarding a given repository. Issues can be posted by anyone. The content of an issue is controlled by the repositories owners.

Pull requests differ from issues. They are proposed by users and these proposals can be either accepted or rejected by the repository's owners.

Every pull request is an issue but not vice-versa. A 'pull\_request' statement is visible if an issue is a pull request.

**Ques 7:** Describe the information shown under "Pulse".

Ans:

The Pulse option helps users monitor activity of a repository or a project. It allows us to observe new and merged pull requests, issues which are open or closed for discussion, issues which are unresolved, additions, deletions and commits which have been pushed by the repository's moderators.

**Ques 8:** Give the URL of your repository.

Ans: <https://github.com/DevayanMandal/cs507s16lab3.git>

**Ques 9:** Issue the command git status. What does this print?

Ans:

```
8. git status
[mandald427@mogul cs507s16lab3]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#    modified:   Hello.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#    Hello.java~
no changes added to commit (use "git add" and/or "git commit -a")
```

**Ques 10:** Issue the command git status again. What does this print now?

Ans:

```
[mandald427@mogul cs507s16lab3]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#    modified:   Hello.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#    Hello.java~
```

**Ques 11:** Issue the command git status again. What does this print now?

Ans:

```
[mandald427@mogul cs507s16lab3]$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#    Hello.java~
nothing added to commit but untracked files present (use "git add" to track)
```

**Ques 12:** Issue the command git status again. What does this print now?

Ans:

```
[mandald427@mogul cs507s16lab3]$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#    Hello.java~
nothing added to commit but untracked files present (use "git add" to track)
```

**Ques 13:** A successful update to your Hello.java by each member is required. (28 points)  
There is much more git can do, but hopefully this gives you the basics of a typical workflow.

Ans:

Devayan: ✓

Ikhlas: ✓