ECEN 489: Data Management and Cloud Services
Task 2 – Git and Master Repository

1 Git

Git is a popular source code management system. Every Git working directory is a full-fledged repository, with complete history and full version tracking capabilities, not dependent on network access or a central server. In this sense, it trades off space for speed. The JGit implementation of Git is a pure Java software library, designed to be embedded in any Java application. It is used in EGit, a Git client for the Eclipse IDE. Pertinent documentation about EGit can be found on the Eclipse website.

- `http://www.eclipse.org/egit/documentation/`

Action Items

- **Read:** About Git.
  `http://gitscm.com/`
- **Download and Install:** EGit.
  `http://www.eclipse.org/egit/`

GitHub

GitHub is a web-based hosting service for source code management (SCM). It is built on the Git revision control system and offers free accounts for open source projects. According to the terms of service, if bandwidth usage of an account significantly exceeds the average of other GitHub customers, the associated file hosting service may be immediately disabled.

To facilitate integration of our projects, we will employ a Git archive. Our master Git repository is hosted on GitHub.

`https://github.com/chmbrlnd/ECEN489-Spring2014.git`

Active participants in this course will be given access to our master repository. You will find below a short list of frequently used commands.

Common Actions

- **Init:** The `init` command creates a new local repository.
- **Clone:** Use `clone` to instantiate a working copy from a master repository. This is usually the first command employed to establish a local working hierarchy under this paradigm.

- **Add:** The `add` command is used to add one or more files to staging. Only add pertinent files to the repository.
- **Commit:** The `commit` command incorporates changes to your working copy of the repository.
- **Push:** The `push` command sends changes to the master branch, typically a remote repository.
- **Pull:** The `pull` command fetches and merges changes on the remote server to the local working directory.
- **Mergetool:** Sometimes, there may be a discrepancy between the latest version of a file and its working copy on a given host. In such cases, the developer may need to take action to resolve these issues. This can be achieve through normal editing, followed by the Git `add` command. Alternatively, one can use the `mergetool` command, which initiates a visual tool.
- **Status:** The `status` command lists the status of working files and directories.

Action Items

- **Account:** Go to `https://github.com` and create a developer account. Email your GitID to your instructor via your TAMU account.
- **Clone:** Once you have been added to the list of collaborators, use `EGit` within `Eclipse` to clone the master repository.
- **Directory:** At the root of your working repository, make a directory named `<NetID>`. This location is where you will commit all your individual work. Within this directory, create a file labeled `CompletedTasks.txt` that contains a one-line header.

  ```
  = Completed Tasks =
  ```

  Add and commit your modifications to your working repository, and push the information on the master repository.