**Week 5: Working with GIT HUB and Eclipse**

1. Evaluation of final SRS document.
2. Hands – on practice of GitHUB commands like, pull, push, fork, ssh, clone etc.
3. Installation/updation of eclipse to enterprise version, configuring tomcat9 to eclipse.
4. **Collaboration**

* **git remote:** Adds a new remote repository. This is useful when you want to link your local repository with a remote one, like GitHub.

**git remote add <name> <url>**

* **git push**: Used to transfer the commits or pushing the content from the local repository to the remote repository. The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

**git push –u origin <master|current directory>**

* **git clone**: used to create a local working copy of an existing remote repository. The command downloads the remote repository to the local computer.

**git clone <remote url>**

* **git fetch** : Downloads changes from a remote repository without applying them to your working directory. You can later merge these changes.

**git fetch <remote>**

* **git pull** : Fetches and integrates changes from a remote repository by rebasing instead of merging. This creates a linear history.

**git pull <remote> <branch>**

**Using SSH Key for Authentication in Git-Github:**

Using SSH keys in GitHub allows you to securely authenticate your GitHub account from your computer without needing to repeatedly enter your username and password. This method is more secure and convenient, especially for frequent interactions with GitHub repositories.

**Step 1: Set Up SSH**

1. **Check for Existing SSH Keys**
   * Before creating a new SSH key, check if you already have one:

**ls -al ~/.ssh**

Look for files named id\_rsa and id\_rsa.pub. If these files exist, you already have an SSH key pair.

1. **Generate SSH Key (if you haven't already):**
   * Open a terminal and run the following command to generate an SSH key:

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

* + Press Enter to accept the default file location and then enter a passphrase (optional).
  + Your SSH key will be saved to ~/.ssh/id\_rsa and ~/.ssh/id\_rsa.pub.

1. **Add SSH Key to GitHub:**
   * Copy the contents of your SSH key to your clipboard:

cat ~/.ssh/id\_rsa.pub

* + Copy the entire output.
  + Now, Go to GitHub, navigate to **Settings > SSH and GPG keys**, and click **New SSH key**.
  + Paste the SSH key into the "Key" field and give it a title, then click **Add SSH key**.

1. **Test SSH Connection:**
   * Run the following command to test the connection:

ssh -T [git@github.com](mailto:git@github.com)

* + You should see a message like "Hi username! You've successfully authenticated."

**Fork :**

In Git, a **"fork"** refers to a copy of a repository from one user's account to another user's account on a platform like GitHub. Forking is commonly used to contribute to open-source projects or collaborate on projects where you don't have direct write access to the original repository. Forking allows you to work independently.

1. **Fork a Repository:**
   * Go to a repository you want to contribute to on GitHub (e.g., <https://github.com/octocat/Spoon-Knife>).
   * Click the **Fork** button in the top-right corner to create a copy of the repository under your account.
2. **Clone the Forked Repository:**
   * On your GitHub account, navigate to your forked repository.
   * Click the **Code** button and copy the HTTPS or SSH URL (e.g., git@github.com:yourusername/Spoon-Knife.git).
   * Clone the repository to your local machine:

In Git Bash, type

git clone git@github.com:yourusername/Spoon-Knife.git

* + Navigate into the repository:

cd Spoon-Knife

**Step 3: Make Changes and Push**

1. **Create a New Branch:**
   * Create and switch to a new branch for your changes:

git checkout -b my-feature-branch

1. **Make Changes:**
   * Edit some files in the repository, or create new files.
   * After making changes, stage them for commit:

git add .

* + Commit the changes with a message:

git commit -m "Add my new feature"

1. **Push Changes to GitHub:**
   * Push the changes to your forked repository on GitHub:

git push origin my-feature-branch

**Step 4: Pull from Upstream Repository**

1. **Add Upstream Remote:**
   * Add the original repository as an upstream remote:

git remote add upstream git@github.com:octocat/Spoon-Knife.git

1. **Fetch and Pull Updates:**
   * Fetch the latest changes from the upstream repository:

git fetch upstream

* + Merge the changes into your local branch:

git pull upstream main

* + Resolve any conflicts if they arise.

1. **Push Merged Changes:**
   * After resolving conflicts and merging, push the changes to your forked repository:

git push origin my-feature-branch

**Step 5: Submit a Pull Request**

1. **Create a Pull Request:**
   * Go to your forked repository on GitHub.
   * You should see a banner suggesting to compare and create a pull request. Click on it.
   * Review your changes and create the pull request to propose your changes to the original repository.
2. **Installing Eclipse**

To install Eclipse IDE for Enterprise Java Developers (commonly known as Eclipse Enterprise) on your Windows 10 system, follow these steps:

**Step 1: Download Eclipse Installer**

1. Visit the Eclipse Downloads Page:
   * Go to the official Eclipse Downloads page.
2. Download the Eclipse Installer:
   * Scroll down to find the "Eclipse IDE for Enterprise Java and Web Developers" option.
   * Click on the "Download x86\_64" link to download the installer for Windows.

**Step 2: Run the Eclipse Installer**

1. Launch the Installer:
   * After the download is complete, locate the installer file (eclipse-inst-jre-win64.exe) in your Downloads folder and double-click it to run.
2. Choose the IDE Package:
   * In the Eclipse Installer, you will see various package options. Select "Eclipse IDE for Enterprise Java and Web Developers."
3. Select Installation Folder:
   * Choose a destination folder where you want to install Eclipse. The default location is typically fine, but you can change it if needed.
4. JDK Selection:
   * The installer should automatically detect your installed JDK. Ensure that it points to the correct JDK version. If not, you can manually browse and select the correct JDK path.
5. Start the Installation:
   * Click on the "INSTALL" button to start the installation process.
6. Accept the License Agreement:
   * Read and accept the license agreement to proceed with the installation.

**Step 3: Launch Eclipse**

1. Finish Installation:
   * Once the installation is complete, click on the "LAUNCH" button to start Eclipse.
2. Workspace Selection:
   * The first time you launch Eclipse, it will ask you to select a workspace. This is the folder where all your projects and settings will be stored. You can either accept the default location or choose a new one.
3. Start Using Eclipse:

* After selecting the workspace, Eclipse will start up, and you can begin creating or importing projects.

**Download and install Tomcat :**

Apache Tomcat, commonly referred to as Tomcat, is an open-source web server and servlet container developed by the Apache Software Foundation. It is designed to serve Java applications and is widely used to deploy Java-based web applications and services.

Installing Apache Tomcat 9.0 on Windows 10 involves downloading the software, setting up the environment, and configuring Tomcat.

**1. Download Apache Tomcat 9.0**

1. Go to the **Apache Tomcat 9.0 download page**.
2. Under the **Binary Distributions** section, find the "Core" section and download the **32-bit/64-bit Windows Service Installer** (.exe file).

**2. Run the Installer**

1. Locate the downloaded .exe file and double-click to start the installation.
2. During the installation, follow these steps:
   * **Welcome Screen**: Click Next.
   * **License Agreement**: Accept the agreement and click Next.
   * **Choose Components**: Select the components you need (leave the defaults selected for a basic setup) and click Next.
   * **Configuration**:
     + Set the **Tomcat Server** ports:
       - HTTP/1.1 Connector Port (default is 8080).
       - If you want to run multiple instances or have another service using 8080, you can change this port.
     + **Tomcat Administrator Login**: Enter a username and password (this is optional).
     + Click Next.
   * **Choose Java Virtual Machine**:
     + Ensure the correct path to your JDK (Java Development Kit) is set. If you haven't installed the JDK, you'll need to install it first.
   * **Choose Install Location**:
     + Choose the directory where you want to install Tomcat or leave the default location.
     + Click Next, then click Install to begin the installation.

**3. Complete the Installation**

1. After the installation, you’ll have the option to start Tomcat immediately. If selected, the Tomcat service will start.
2. Click Finish to exit the installer.

**4. Verify Installation**

1. Open a web browser and go to http://localhost:8080 (or the port you specified during installation).
2. If Tomcat is running correctly, you should see the Tomcat homepage.

**5. Manage Tomcat as a Service**

* **Starting/Stopping Tomcat**:
  + You can manage Tomcat from the Windows Services management console. Open it by typing services.msc in the Start menu.
  + Find Apache Tomcat 9.0 in the list, and you can start, stop, or restart the service from there.
* **Configuring Tomcat**:
  + Tomcat's configuration files are located in the conf directory of your Tomcat installation (e.g., C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf).
  + Common files to configure include:
    - server.xml: Configures the main settings of the server.
    - web.xml: Configures web application defaults.

**Configure Tomcat Web-Server on Eclipse**

* **Eclipse IDE**: Ensure you have Eclipse IDE for Java EE Developers (or any version that supports Java EE). If not, download it from the Eclipse official website.
* **Apache Tomcat 9.0**: Ensure Tomcat 9.0 is installed on your system. If not, download it from the [Apache Tomcat website](https://tomcat.apache.org/download-90.cgi).

**1. Open Eclipse and Configure the Server**

1. **Open Eclipse**: Start your Eclipse IDE.
2. **Open the Servers View**:
   * Go to Window > Show View > Other....
   * In the "Show View" dialog, type Servers and select it, then click Open.
3. **Add a New Server**:
   * In the "Servers" view, right-click and select New > Server.
   * In the "New Server" wizard, expand the "Apache" node, select Tomcat v9.0 Server, and click Next.
4. **Configure the Tomcat Server**:
   * **Server Name**: You can leave the default name or provide a custom name.
   * **Tomcat Installation Directory**: Click Browse... and navigate to the directory where you installed Tomcat (e.g., C:\Program Files\Apache Software Foundation\Tomcat 9.0).
   * **JRE Selection**: Ensure that the correct JRE is selected. You can use the default JRE or select another installed JDK.
5. **Add Projects to the Server**:
   * In the next window, you'll have the option to add existing projects to the server. If you have a project you want to run on Tomcat, select it; otherwise, you can add projects later.
   * Click Finish.

**2. Test the Server Configuration**

1. **Start the Server**:
   * In the "Servers" view, right-click on the newly added Tomcat server and select Start.
   * Eclipse will start the Tomcat server. You should see output in the "Console" view indicating that Tomcat is running.
2. **Access Tomcat**:
   * Open a web browser and go to http://localhost:8080. You should see the Tomcat default homepage, confirming that the server is running correctly.

**Conclusion:**

* Students will learn to use Github successfully to push their work into remote repositories, Work with other users codes using Fork and may be contribute as well
* Install Eclipse IDE and Tomcat Server
* Configure Tomcat into Eclipse IDE

**Note: Upload your work in the tesselator through the screens shots saved in document with explanation**