

**PROG6001**

**Managing Software Development**

**Tutorial Module 2**

**Distributed Version Management**

Contents

[Distributed Version Management 3](#_Toc98094685)

[Objectives 3](#_Toc98094686)

[Overview 3](#_Toc98094687)

[Review 3](#_Toc98094688)

[A simple project example 4](#_Toc98094689)

[GitHub in an IDE environment 4](#_Toc98094690)

# Distributed Version Management

## Objectives

On completion of this tutorial you should be able to:

1. Use GitHub to fork, modify files, and create pull requests – Testing, Testing, Testing
2. Recognise and use git commands

## Overview

Distributed version management is managing system component versions across the Internet. In this tutorial, we will look at GitHub, a popular version management system used in public domain systems and in private development projects. Next to GitLab and Atlassian’s Bitbucket, it is one of the most widely used products. Your GitHub account credentials will allow you to logon to GitLab as well.

You are required to create a GitHub account (in case you haven’t already during one of your programming units). It is advised to use a personal email address to be able to access it long after losing access to your SCU email.

***Note:*** It is important to be aware that the work you are doing in GitHub is public, you should avoid disclosing any personal data, your student ID or any other private details.

## Review

To start we will review the version distributed version management terminology you have seen in module content. GitHub uses slightly different terminology but it should be easy to follow. The following table is a summary.

|  |  |
| --- | --- |
| **Study guide/Textbook** | **GitHub** |
| Repository | Repository  Usually called a “Repo” |
| Mainline | Main – also consider the main branch |
| Branch | Branch – in GitHub a “fork” operation creates a branch |
| Merge | Merge – this is more general, e.g. a fork can be merged with the in or any other branch |
|  | Commit – save changes to your branch. Separate commits form a codeline |
| Push | Push – copy repository from user’s machine to the GitHub repository |
| Pull | Pull – take another branch and merge with your branch. Any branch may pull any branch.  Pull request – a developer requests the main owner to merge with the main (mainline) |

At this stage, if you have not yet read the module 2 content or textbook you should do it to help with the remainder of this tutorial.

## A simple project example

We will now create a simple project in one account and go through the pull request and merge process. This is explained in the GitHub “Hello World” tutorial. Note that there are other tutorials on the GitHub website that walk you through other more complicated processes. Make sure you read and understand the documentation in this tutorial.

 Activity 2-1

1. Sign up to GitHub and do the GitHub “Hello world” tutorial:  
   <https://guides.github.com/activities/hello-world/>
2. Logon to GitHub and find the following public repository: *SCUteach/PROG6001-Simple*.  
   Are you able to upload/overwrite anything in the SCUteach repository? (discuss)  
   Are you able to change the text file directly in that repository? (discuss – what is advised and automatically put into place by GitHub when you try?)  
   Fork the *SCUteach/PROG6001*-Simple repository – what happens when you do (discuss)  
   What would happen if you *clone* the repository (eg. download a ZIP file of it?) Is it the same as creating a fork? (discuss)  
   Create a pull request (push operation) of your whole repository back into the main repository. Check your list of pull requests to see how GitHub records them (note that your request might not be pulled).  
   Create another pull request for the modified text file only to see the difference.  
   *(pay attention to how the assignment task is phrased to ensure you request the correct pull request – also test how to delete or recall a pull request)*
3. Go to the following GitHub project:

<https://github.com/Scanate/EthList>

This is a crowdsourced document on the Ethereum blockchain system. It is using GitHub to allow the public to maintain an HTML document.

* Click on the “Contributors” link in the Insights area to see how people have been involved in pull requests contributing to the content
* If you are lucky, you may see an active pull request. Have a look at it. The owner of the main branch must decide if it should be merged.

 Activity 2-2

1. Familiarise yourself with the commands for git branching:  
   <https://learngitbranching.js.org/>

## GitHub in an IDE environment

Most IDEs will provide the developer with a direct way of communicating with GitHub to enable direct access via the IDE and the possibility for creating repositories, creating pull requests and more. There are many resources available on the Internet on how this works and how to set it up in your IDE of choice for your programming language of choice. Please research outside of this tutorial if you would like to use GitHub via the IDE you are using. Always be aware of the security settings of your own repositories (public/private).