Git Tutorial for Computer Science Students: Assignment 1

Charlie Rui Chai
rchai@oxy.edu
Occidental College

1 Introduction

Git is a distributed version control system. This tutorial is designed to introduce Computer Science major students to the fundamentals of using Git. We will focusing on the basics of Init-ing/Cloning, Adding, Committing, Pushing, and topic of stashes, branches, merging and dealing with conflicts. By the end of the tutorial, you should have a basic understanding of how to use Git to manage your code.

2 Setting Up Git

Before starting, ensure you have Git installed on your computer. You can verify this by opening a terminal and typing:

```
git --version
```

If Git is not installed, please visit to download and install it.

3 Git Basics

3.1 Init

```
git init
```

This command creates a new Git repository in your project directory.

3.2 Clone

```
git clone https://repository-url.git
```

Replace https://repository-url.git with the actual URL of the repository.

3.3 Add

```
git add filename
```

To track changes to a file or files.

3.4 Commit

```
git commit -m "Commit message"
```

To save your changes, commit them to the repository with a message.

3.5 Push

```
git push origin main
```

To upload your local commits to a remote repository.

4 Using Branches

To manage new features or experiments in your project, you can use branches. This allows you to work on different versions of your project simultaneously without affecting the main codebase.

4.1 Create a New Branch

```
git branch feature-x
```

4.2 Switch to the New Branch

```
git checkout feature-x
```

Alternatively, create and switch to a new branch in one command:

```
git checkout -b feature-x
```

5 Making Changes and Stashing

While working on your project, you might want to save your work without committing it. Git stash is useful in this scenario.

5.1 Make Some Changes

Let's modify 'hello.py':

```
print("Hello, Git from feature-x!")
```

5.2 Stash Your Changes

If you're not ready to commit:

```
git stash
```

5.3 Apply Your Stashed Changes

When ready to continue:

```
git stash pop
```

6 Merging Changes

After completing your work on a branch, you may want to merge those changes back into the main branch.

6.1 Switch Back to the Main Branch

```
git checkout main
```

6.2 Merge the Feature Branch

```
git merge feature-x
```

This command merges the changes from 'feature-x' into the 'main' branch, incorporating your new features or changes into the main project.

7 References

[1] https://www.overleaf.com/learn/latex/Tutorials
[2] https://www.gitkraken.com/learn/git/tutorials