

# CS6.201 Introduction to Software Systems

## Lab Activity 1

January 8, 2025

### Introduction to Git and GitHub

---

## Instructions

- Ensure that Git is installed on your system.
  - Create a GitHub account if you do not already have one.
  - Submission for this activity must be done on Moodle.
- 

## 1 Configure Git

- 1.1 Configure the global Git username.
- 1.2 Configure the global Git email.
- 1.3 Verify the Git configuration.

## 2 Distance Finder

Write a C program to calculate the distance between two points in a 2D plane.

- 2.1 Work within a new directory named `dist`.
- 2.2 Define a struct `Point` with two members, `x` and `y`, of type `int`. Save the definition in a file named `point.h`.
- 2.3 Define a function `distance` that takes two `Point` arguments and returns the Euclidean distance between them. Save the implementation in a file named `distance.c`.
- 2.4 Complete the subtasks described below. Save the output of each subtask in file named '`<subtask>.txt`' (e.g., '`subtask2.1.1.txt`').

## 2.1 Checkpoint 1: Save Your Progress

- 2.1.1 Initialize a Git repository.
- 2.1.2 Add the files to the staging area. (No output required)
- 2.1.3 Check the repository status.
- 2.1.4 Commit the changes.
- 2.1.5 Verify that the commit was successful.

## 2.2 Experimentation: Calculate Manhattan Distance

Switch to finding the Manhattan distance instead of the Euclidean distance.

- 2.2.1 Create a new branch named `manhattan`. (No output required)
- 2.2.2 Switch to the `manhattan` branch.
- 2.2.3 Verify the branch creation and switch.
- 2.2.4 Modify the `distance` function to calculate the Manhattan distance. (No output required)
- 2.2.5 Compare the changes.
- 2.2.6 Add the modified files to the staging area. (No output required)
- 2.2.7 Check the repository status.
- 2.2.8 Commit the changes.

## 2.3 Revert Changes: Start Fresh

Revert to the previous commit and discard the Manhattan distance implementation.

- 2.3.1 Reset the repository to the previous commit. Experiment with or without the `--hard` option.
- 2.3.2 Verify the repository status and history.

# Hit Submit

Zip the `dist` directory, ensuring it adheres to the provided file structure. Submit the zip file on Moodle.

```
dist
|-- distance.c
|-- point.h
|-- subtask2.1.1.txt
|-- subtask2.1.3.txt
|-- subtask2.1.4.txt
|-- subtask2.1.5.txt
|-- subtask2.2.2.txt
|-- subtask2.2.3.txt
|-- subtask2.2.5.txt
|-- subtask2.2.7.txt
|-- subtask2.2.8.txt
|-- subtask2.3.1.txt
|-- subtask2.3.2.txt
1 directory, 13 files
```

---

## 3 KYC: Know Your Companions

Interact and collaborate with your peers using GitHub.

### 3.1 Duplicate the Repository

Duplicate the provided repository into your GitHub account.

### 3.2 Introduce Yourself

Clone your repository locally and create a new file named `<rollno>.txt`. Fill in the following details.

*Name:*

*Branch:*

*Place of residence:*

*Hobbies:*

*Reason for joining IIIT:*

### 3.3 Push Changes

Commit and push the updated repository to your GitHub account.

### 3.4 Create a Pull Request

Submit a pull request to the original repository.

### 3.5 Sync Your Repository

Synchronize your repository with the original one to view introductions from other students.