CMPS 312 Mobile Application Development

LAB 1: Setting Up the Android Development Environment and Version Control

---

Objective

1. Learn the basics of Git and GitHub for version control.
2. Set up your Flutter development environment and familiarize yourself with it.
3. Create, run, and debug simple Flutter applications.

By the end of this lab, you will have a foundational understanding of the tools developers use to create Flutter apps. You'll also know how to develop and debug a simple Flutter application and utilize Git and GitHub to manage and track changes in your code.

Overview:

This lab is divided into two parts that provide essential skills for Flutter app development and effective version control using GitHub:

- **Part A: Introduction to GitHub :** Get started with GitHub, a platform for collaborative software development. You will create a GitHub repository to manage all your lab work, including in-lab assessments and exam submissions.

- **Part B: Android Development Setup :** Flutter Development Setup: Set up your Flutter development environment by installing the necessary SDKs and tools. You will also develop and test a basic Flutter application.

Feel free to ask for assistance during the lab. Our goal is to help you grasp GitHub and establish an efficient Flutter development environment for successful app creation.

PART A – GitHub

This section will guide you through essential GitHub operations, including handling merge conflicts, ensuring your ability to manage code collaboratively and effectively.

1. Install GitHub Desktop:
    a. Download and install the GitHub Desktop app from https://desktop.github.com.

2. Create a GitHub Account:
    a. If you don't have an account, sign up at GitHub https://github.com. Your GitHub username must follow the format firstname-QuUsername (e.g., ali-am1206290 or hana-hm334490).
3. Set Up a Repository:
    a. Create a new repository on GitHub for your lab work.
    b. Clone the repository to your local machine using GitHub Desktop.
4. Initial Commit:
    a. Create a new file in your repository and make your first commit.
    b. Push the changes to your GitHub repository.
5. Branching and Merging:
    a. Create a new branch from the master repository.
    b. Make changes to files in the branch.
    c. Attempt to merge the branch back into the master, resolving any conflicts.
6. Pull Requests:
    a. Open a pull request to merge your branch into the master repository.
    b. Document the steps you took to resolve any merge conflicts.

PART B – Flutter Development Setup

1. Install Flutter SDK:

Download and install the Flutter SDK from the official **Flutter website**, then follow the steps outlined below to complete the setup.

- **Windows:** Download the Flutter SDK from the Flutter website by following this step by step instructions
- **macOS:** Download the Flutter SDK from the Flutter website by following this step by step instructions

2. Creating and Running Project:

- Create a New Flutter Project:
- Open your preferred IDE and create a new Flutter project.
- Name the application "Hello World" and set the package name as `com.cmps312.helloworld`.
- Ensure Dart is selected as the programming language.
- Finish the setup process.

2.1 Running project on Android:

- **Using a Physical Device:**
  - Connect your device via USB and ensure USB debugging is enabled in the Developer Options.
  - Run the Flutter app by clicking the "Run" button in your IDE.
- **Using an Emulator:**
  - Open the AVD Manager in your IDE to set up an Android Emulator.
  - Run your Flutter app on the emulator.

2.2 Running project on iOS (macOS only):

- **Using a Physical Device:**
  - Connect your iOS device via USB.
  - Ensure you have set up the necessary provisioning profiles and certificates in Xcode.
  - Run the Flutter app by clicking the "Run" button in your IDE.
- **Using the iOS Simulator:**
  - Open the iOS Simulator through Xcode (or directly from your IDE if it supports it).
  - Run your Flutter app on the simulator.