

CMPUT 301 - Lab 8: Test-Driven Development (TDD) with Continuous

1 Open Android Studio & Clone the Repo

- Go to **File** → **New** → **Project from Version Control**
- Paste this URL: <https://github.com/Durlov1603025/Lab08.git>
- Click Clone
- Wait until Gradle sync finishes.


2 Connect Android Studio to Your GitHub Account (if not already)

- Go to **File** → **Settings** → **Version Control** → **GitHub** → **Add Account**
- Log in (browser authorization is easiest)
- Back in Android Studio, you should now see your account connected.

3 Share Project to Your Own GitHub Repository

- In Android Studio, go to **VCS** → **Share Project on GitHub** (or **Git** → **GitHub** → **Share Project on GitHub** if VCS isn't visible)
- Choose a repo name
- Add a short description, (e.g. CMPUT 301 Lab 8 TDD Starter)
- Click **Share** or **Share Anyway**
- Android Studio will automatically push your local code to your new GitHub repository.

4 Verify Continuous Integration (CI) Setup

- Make a small change locally (e.g., add a comment)
- **Commit** → **Push**
- Go to your **GitHub repo** → **Actions** tab
 - You'll see your workflow running automatically (yellow = running)
 -  If all tests pass, you'll see a green checkmark
- Congratulations! You've verified your CI pipeline.

5 Write a Failing Test for hasCity() (Start of TDD)

- Navigate to: `app/src/test/java/com/example/Lab08/CustomListTest.java`
This is where you'll write your first failing test (TDD). This file contains tests for the `CustomList` class.

- Modify or add new test methods to test the **hasCity()** feature

```
package com.example.lab08;

import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CustomListTest {
    @Test
    public void testHasCity() {
        CustomList list = new CustomList();
        City calgary = new City("Calgary", "AB");
        list.addCity(calgary);

        // This will fail initially because hasCity() doesn't
        exist
        assertTrue(list.hasCity(calgary));
    }
}
```

- It's normal to see an error..... **hasCity()** doesn't exist yet.

6 Run Tests (RED Phase)



- Go to **Project View** → Right-click on the **test folder** → Run '**Tests in Lab08.app**'
- You'll see **red X results**..... perfect! The test failed.
- **Commit & Push** this failing test to GitHub
- Go to your repo → **Actions tab** → CI workflow should show **X** (failing build).

7 Implement the hasCity() Function (GREEN Phase)

- Navigate to **CustomList.java**:
app/src/main/java/com/example/Lab08/CustomList.java
- Add this method inside the CustomList class:

```
public boolean hasCity(City city) {
    return cities.contains(city);
}
```

8 Run Tests Again (Should Pass)

- Run tests again (**Project view** → **right-click** → **Run Tests**)
- This time they should pass (**green** )
- Commit & Push again
- Go to **GitHub** → **Actions tab** → CI will run automatically again and show  (passing build).