EECE 2140: Computing Fundamentals for Engineers

Spring 2026

Lab Assignment #01 Set up GitHub repo and VSCode/Makefile build environment on Linux

Topic: WSL2 + VS Code + GitHub + Overleaf

Student Name	<Wa Kenneth Fan>
NUID	<002221067>
Email	<fan.wa@northeastern.edu>
Instructor	Dr. Fatema Nafa
Due Date	<Jan 11, 2026>

Submission Checklist (must match syllabus requirements)

- Submit **one PDF** generated from this Overleaf report.
- Include **clear screenshots** for each task (text must be readable).
- Include your **GitHub profile link** and **Overleaf project link** in the report.
- File name: `Assignment01_LastName.pdf`

How to complete this assignment

1. Complete each task below in order.
2. For each task, include the required evidence (screenshots, links, and short notes).
3. When you paste a link, ensure it opens correctly (no private/local-only links unless instructed).

Section A: WSL2 Installation Verification

Task 1.

Verify that WSL2 is installed and Ubuntu runs

Checklist Item

Goal: Confirm that WSL is installed, WSL2 is enabled, and your Linux distro launches.
What to do (Windows PowerShell):

- Open **PowerShell** and run:

Listing 1: WSL checks (PowerShell)

```
wsl --status
wsl -l -v
```

Required evidence:

- Screenshot showing output of `wsl -status`
- Screenshot showing output of `wsl -l -v` (Version should be 2)
- Screenshot showing Ubuntu terminal open (your username visible)

Notes (1–3 sentences):
I am using macOS so I had no issue installing WSL2 since I don't need to. For this section I followed the guide on Canvas for macOS systems and attached screenshots of compiler installations, and evidence of a C++ program running in my terminal.

Task 2.

Install build tools inside Ubuntu and verify versions

Checklist Item

Goal: Install the compiler and tools we will use all semester.
What to do (Ubuntu terminal): Run the commands below.

Listing 2: Install tools (Ubuntu)

```
sudo apt update
sudo apt install -y build-essential cmake git
g++ --version
cmake --version
git --version
```

Required evidence:

- Screenshot showing successful install command(s)
- Screenshot showing `g++`, `cmake`, and `git` version outputs

Notes (1–3 sentences):
I attached screenshots of version outputs for g++, cmake, and git

Section B: Visual Studio Code + WSL Integration

Task 3.

Install VS Code and connect it to WSL

Checklist Item

Goal: Use VS Code as your editor while compiling in WSL.
What to do:

- Install VS Code on Windows.
- In VS Code, install the **WSL extension** (Microsoft).
- From Ubuntu terminal, open a folder in VS Code using `code .`

Listing 3: Open VS Code from Ubuntu (WSL)

```
cd ~
mkdir -p eece2140
cd eece2140
code .
```

Required evidence:

- Screenshot of VS Code showing bottom-left text `WSL: Ubuntu` (or similar)
- Screenshot of VS Code integrated terminal showing a Linux prompt (e.g., `username@...: /eece2140$`)

Notes (1–3 sentences):
I have VS Code installed and opened my folder in VS Code using the macOS terminal. In my screenshots I showed the commands I used to open the folder in VS Code and the end result.

Task 4.

Create and run a C++ "Hello World" from VS Code in WSL

Checklist Item

Goal: Confirm you can compile and run a C++ program using Linux tools.
What to do:

- Create `main.cpp` in the `eece2140` folder.
- Compile and run using the terminal commands below.

Listing 4: Compile and run (WSL terminal inside VS Code)

```
g++ -std=c++17 -Wall -Wextra main.cpp -o main
./main
```

Required evidence:

- Screenshot of `main.cpp` open in VS Code
- Screenshot of terminal output showing successful run (your program output)

Paste your C++ code here:

Listing 5: main.cpp

```
#include <iostream>

int main() {
    std::cout << "Hello from EECE 2140 on WSL2!\n";
    return 0;
}
```

Notes (1–3 sentences):
I created the main.cpp file, then compiled and executed it inside of VS Code. The screenshot shows both the file opened in VS Code and the output in the VS Code terminal.

Section C: GitHub Account and First Repository

Task 5.

Create a GitHub account and set up your profile

Checklist Item

Goal: Create a GitHub account and confirm your profile is accessible.
What to do:

- Create (or confirm) your GitHub account.
- Add a profile picture and your name.

Required evidence:

- Screenshot of your GitHub profile page
- Paste your GitHub profile link below

GitHub Profile Link:
https://github.com/Wallfou

Task 6.

Create a repository and push your Hello World project

Checklist Item

Goal: Create a repo and push code from WSL using git.
What to do (WSL terminal): Use the command sequence below (edit placeholders).

Listing 6: Initialize git repo and push (edit placeholders)

```
cd ~/eece2140
git init
git add main.cpp
git commit -m "Add Hello World"

# Replace with your repo URL from GitHub (HTTPS or SSH)
git branch -M main
git remote add origin <YOUR_REPO_URL_HERE>
git push -u origin main
```

Required evidence:

- Screenshot of terminal showing successful `git push`
- Screenshot of your GitHub repo showing `main.cpp` uploaded

GitHub Repository Link:
https://github.com/Wallfou/EECE2140.git
Notes (1–3 sentences):
I created a repo and commited the main.cpp file. The screenshots include a picture of my profile, successful commit, and successful push in terminal.

Section D: Overleaf Setup and Report Submission

Task 7.

Create an Overleaf account and compile this report

Checklist Item

Goal: Confirm you can use Overleaf and compile a PDF.
What to do:

- Create (or confirm) an Overleaf account.
- Upload this `.tex` file to a new Overleaf project.
- Click **Recompile** and download the PDF.

Required evidence:

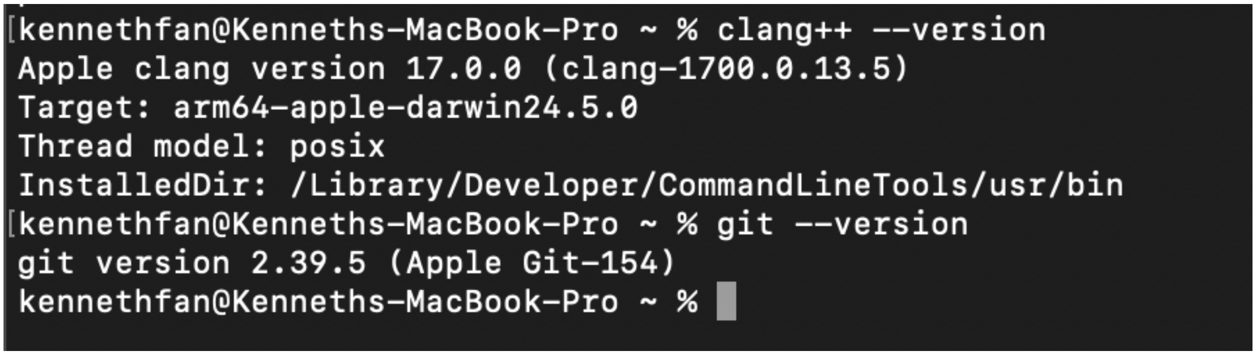
- Screenshot of your Overleaf project (left file tree visible)
- Screenshot of successfully compiled PDF preview

Overleaf Project Link (if shareable):
https://www.overleaf.com/read/sgvfvjnnfjrt#5eb58a
Notes (1–3 sentences):
I added a link for viewing this Overleaf project, and included screenshot of the project tree and pdf preview.

Appendix: Screenshots

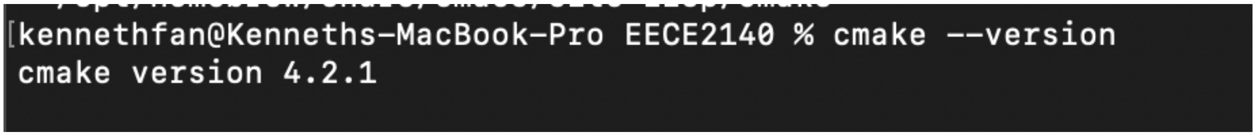
Insert your screenshots here. Ensure they are readable and clearly labeled.

A1: (Mac) Checking clang and git installation



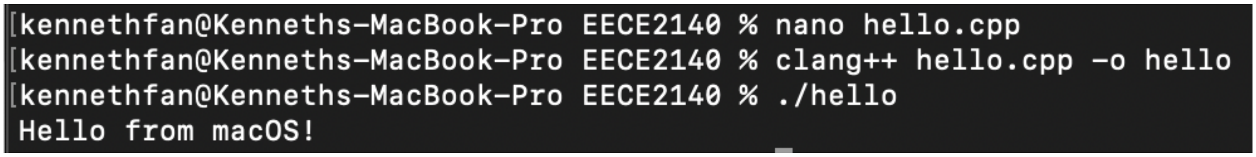
```
[kennethfan@Kenneths-MacBook-Pro ~ % clang++ --version
 Apple clang version 17.0.0 (clang-1700.0.13.5)
 Target: arm64-apple-darwin24.5.0
 Thread model: posix
 InstalledDir: /Library/Developer/CommandLineTools/usr/bin
[kennethfan@Kenneths-MacBook-Pro ~ % git --version
 git version 2.39.5 (Apple Git-154)
 kennethfan@Kenneths-MacBook-Pro ~ %
```

A2: (Mac) Checking CMake installation



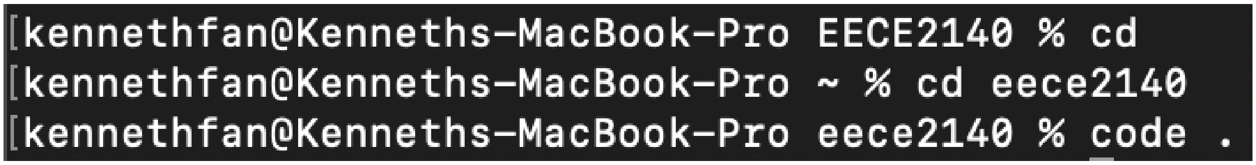
```
[kennethfan@Kenneths-MacBook-Pro EECE2140 % cmake --version
 cmake version 4.2.1
```

A3: (Mac) Running C++ program



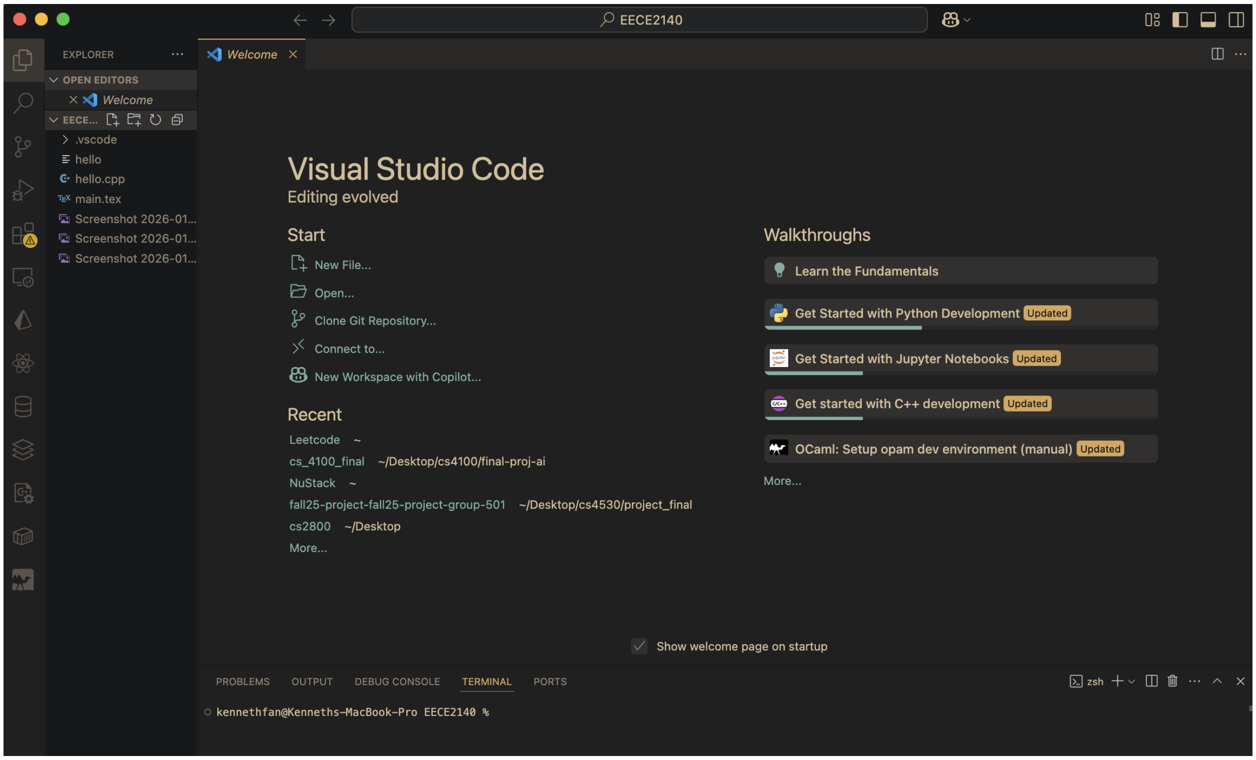
```
[kennethfan@Kenneths-MacBook-Pro EECE2140 % nano hello.cpp
[kennethfan@Kenneths-MacBook-Pro EECE2140 % clang++ hello.cpp -o hello
[kennethfan@Kenneths-MacBook-Pro EECE2140 % ./hello
 Hello from macOS!
```

B1: (Mac) Commands used to open VS Code from terminal



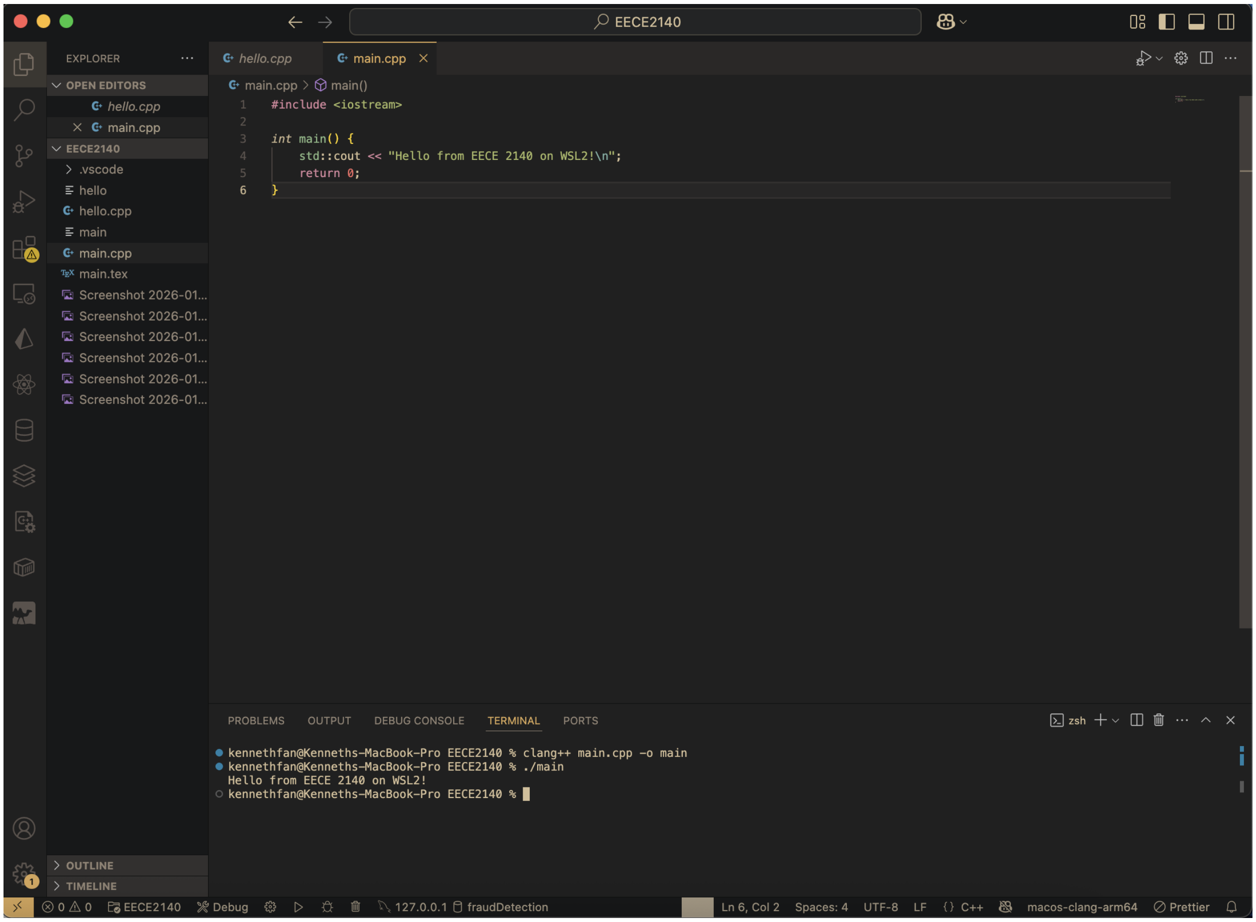
```
[kennethfan@Kenneths-MacBook-Pro EECE2140 % cd
[kennethfan@Kenneths-MacBook-Pro ~ % cd eece2140
[kennethfan@Kenneths-MacBook-Pro eece2140 % code .
```

B2: (Mac) Result of opening VS Code from terminal



```
kennethfan@Kenneths-MacBook-Pro EECE2140 %
```

B3: (Mac) C++ file execution inside VS Code (Showing both main.cpp opened and output of the file in terminal)

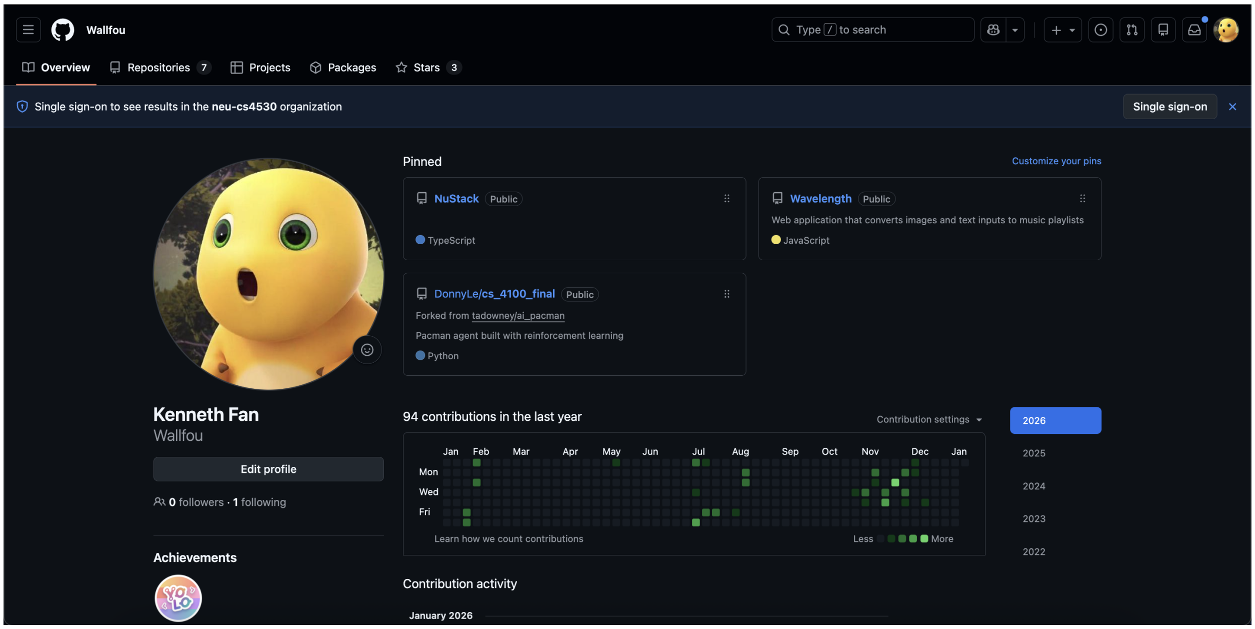


```cpp
#include <iostream>

int main() {
    std::cout << "Hello from EECE 2140 on WSL2!\n";
    return 0;
}
```

```
● kennethfan@Kenneths-MacBook-Pro EECE2140 % clang++ main.cpp -o main
● kennethfan@Kenneths-MacBook-Pro EECE2140 % ./main
  Hello from EECE 2140 on WSL2!
○ kennethfan@Kenneths-MacBook-Pro EECE2140 % 
```

C1: GitHub profile



Single sign-on to see results in the **neu-cs4530** organization

Pinned

NuStack Public
TypeScript

Wavelength Public
Web application that converts images and text inputs to music playlists
JavaScript

DonnyLe/cs_4100_final Public
Forked from tadowney/ai_pacman
Pacman agent built with reinforcement learning
Python

Kenneth Fan
Wallfou

0 followers · 1 following

Achievements

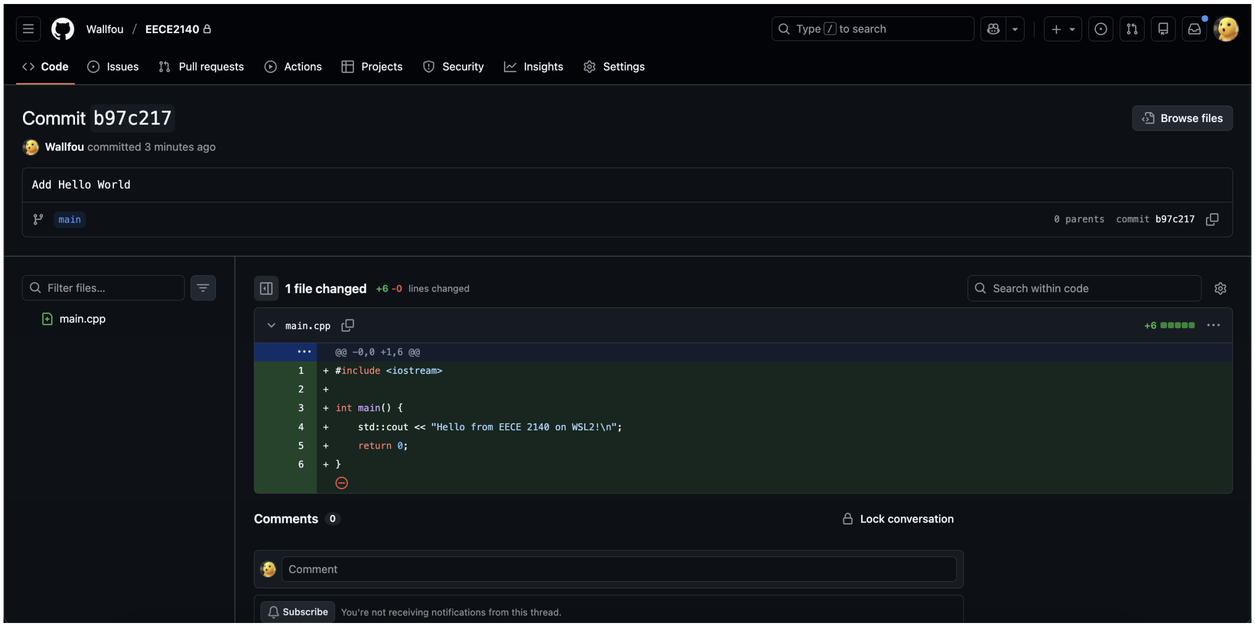
94 contributions in the last year

Learn how we count contributions

Contribution activity

January 2026

C2: Screenshot of successful commit in Github repo



Commit b97c217

Wallfou committed 3 minutes ago

```
Add Hello World
```

main

0 parents commit b97c217

main.cpp

1 file changed +6 -0 lines changed

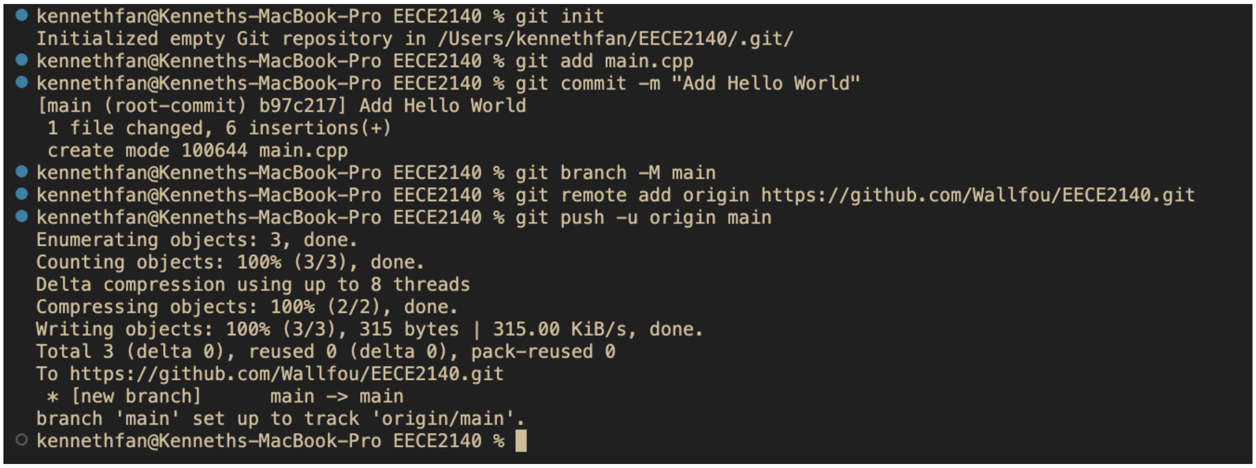
main.cpp +6

```
@@ -0,0 +1,6 @@
+ #include <iostream>
+
+ int main() {
+     std::cout << "Hello from EECE 2140 on WSL2!\n";
+     return 0;
+ }
```

Comments 0

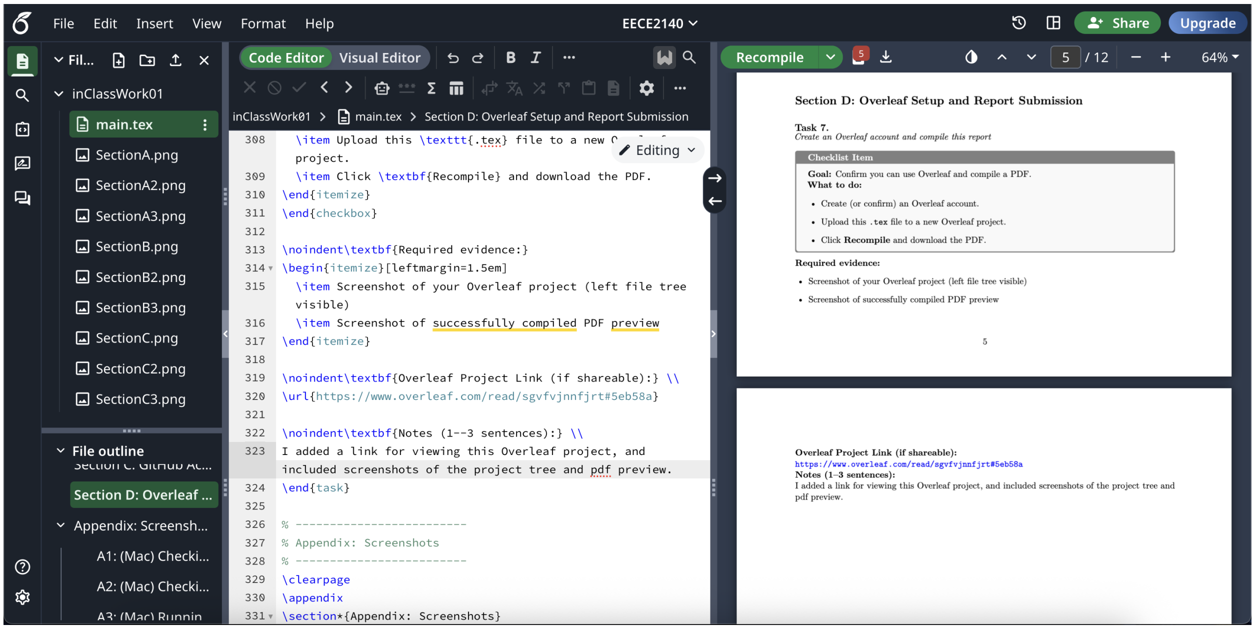
You're not receiving notifications from this thread.

C3: Screenshot of successful push in terminal



```
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git init
  Initialized empty Git repository in /Users/kennethfan/EECE2140/.git/
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git add main.cpp
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git commit -m "Add Hello World"
  [main (root-commit) b97c217] Add Hello World
   1 file changed, 6 insertions(+)
   create mode 100644 main.cpp
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git branch -M main
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git remote add origin https://github.com/Wallfou/EECE2140.git
● kennethfan@Kenneths-MacBook-Pro EECE2140 % git push -u origin main
  Enumerating objects: 3, done.
  Counting objects: 100% (3/3), done.
  Delta compression using up to 8 threads
  Compressing objects: 100% (2/2), done.
  Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
  Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
  To https://github.com/Wallfou/EECE2140.git
   * [new branch]      main -> main
  branch 'main' set up to track 'origin/main'.
○ kennethfan@Kenneths-MacBook-Pro EECE2140 % █
```

D1: Overleaf project tree on the left and compiled PDF preview on the right



```
  \item Upload this \texttt{.tex} file to a new O
  project.
  \item Click \textbf{Recompile} and download the PDF.
\end{itemize}
\end{checkbox}

\noindent\textbf{Required evidence:}
\begin{itemize}[leftmargin=1.5em]
  \item Screenshot of your Overleaf project (left file tree
  visible)
  \item Screenshot of successfully compiled PDF preview
\end{itemize}

\noindent\textbf{Overleaf Project Link (if shareable):} \\
\url{https://www.overleaf.com/read/sgvfvjnnfjrt#5eb58a}

\noindent\textbf{Notes (1--3 sentences):} \\
I added a link for viewing this Overleaf project, and
included screenshots of the project tree and pdf preview.
\end{task}

% ------------------------
% Appendix: Screenshots
% ------------------------
\clearpage
\appendix
\section*{Appendix: Screenshots}
```

Section D: Overleaf Setup and Report Submission

Task 7.
Create an Overleaf account and compile this report

Checklist Item

Goal: Confirm you can use Overleaf and compile a PDF.
What to do:

- Create (or confirm) an Overleaf account.
- Upload this `.tex` file to a new Overleaf project.
- Click **Recompile** and download the PDF.

Required evidence:

- Screenshot of your Overleaf project (left file tree visible)
- Screenshot of successfully compiled PDF preview

5

Overleaf Project Link (if shareable):
https://www.overleaf.com/read/sgvfvjnnfjrt#5eb58a
Notes (1–3 sentences):
I added a link for viewing this Overleaf project, and included screenshots of the project tree and pdf preview.

Reminder: Your screenshots must be clear, readable, and directly support each checklist item.