
**National Textile University**

**Department of Computer Science**

| | |
|---|---|
| **Name:** | **Ayeza Naseem** |
| **Class:** | **SE-A (5th)** |
| **Roll no.** | **23-NTU-CS-1143** |
| **Subject:** | **Operating System** |
| **Lab report:** | **Hometask 1** |
| **Date:** | **25-09-2025** |

# Operating Systems – COC 3071L

## SE 5th A – Fall 2025

## Objective

The purpose of this assignment is to:

1. Configure **Ubuntu** inside **WSL2 (Windows Subsystem for Linux v2)**.
2. Install and configure **Git** in Ubuntu.
3. Generate and set up **SSH keys** to connect with GitHub.
4. Install the **C development environment** in Ubuntu.
5. Write a **Hello World** program in `C`.

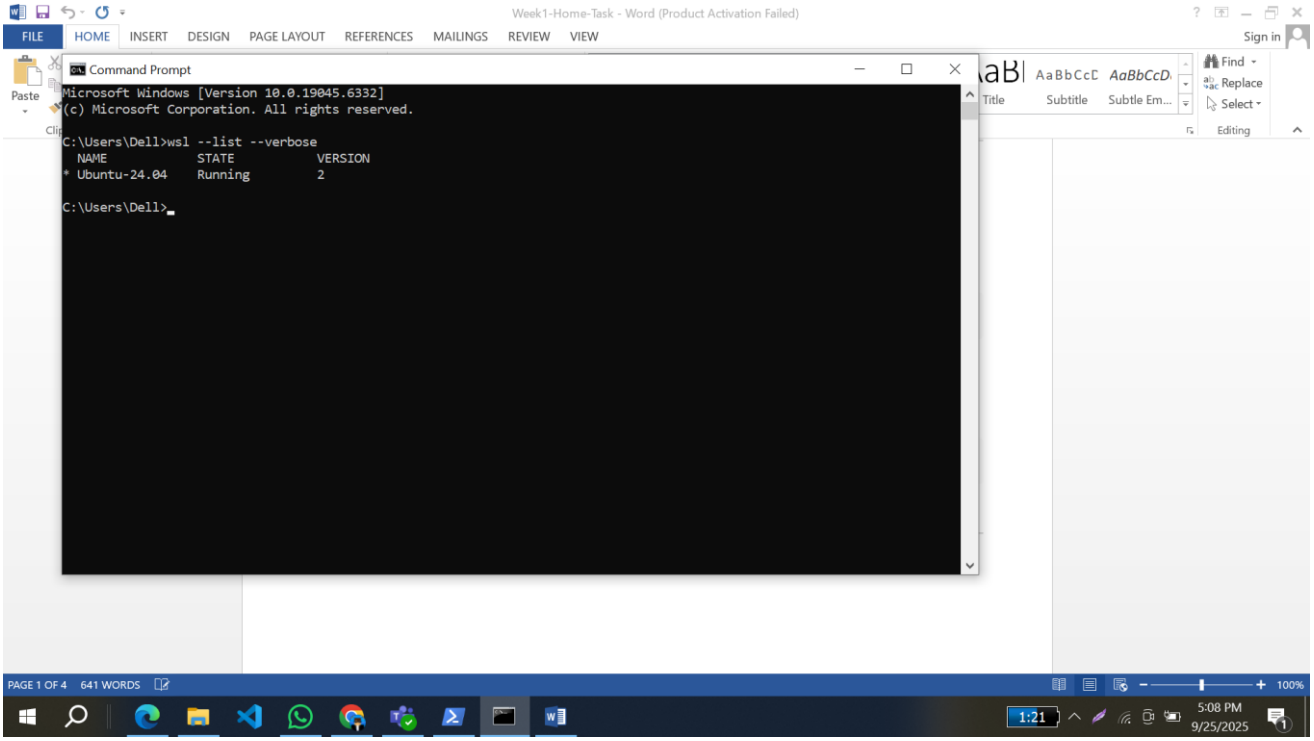
---

# Part A: WSL2 & Ubuntu Setup

1. **Verify WSL2 and Ubuntu installation**
   - Verify installation by running the following command in powershell:

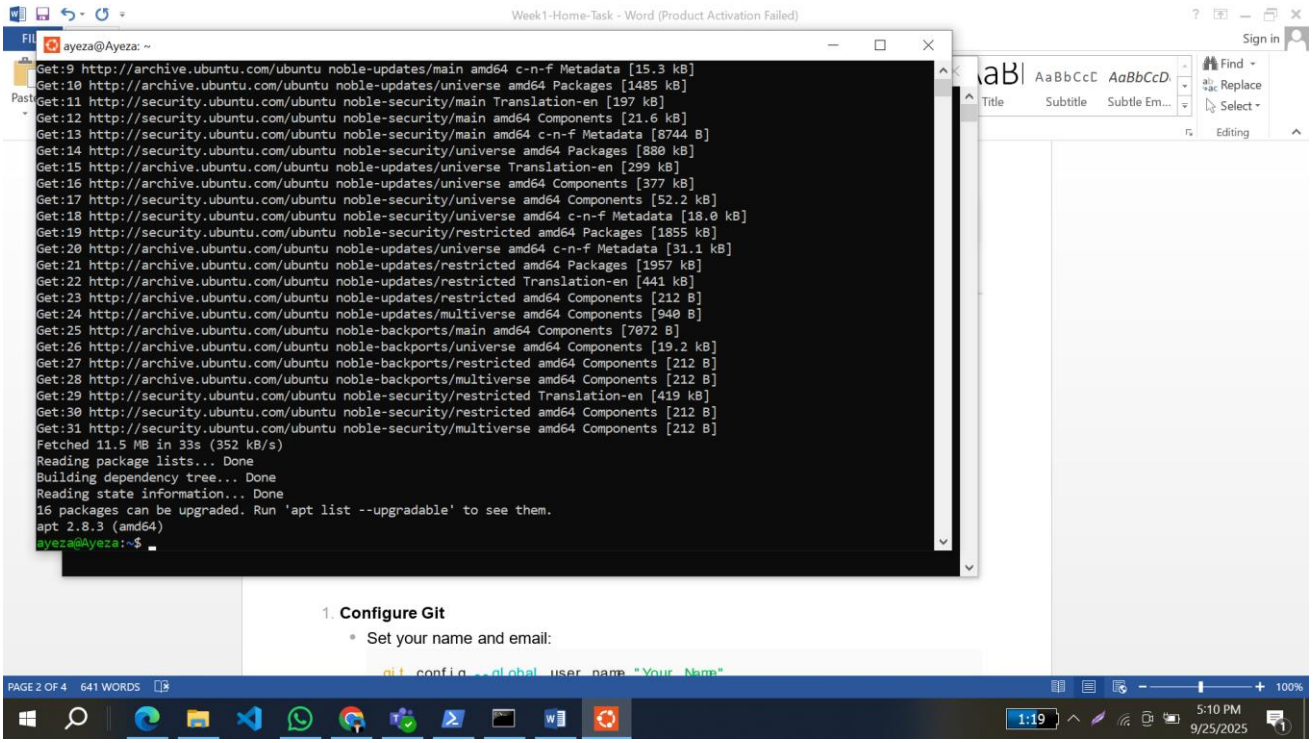
     ```
     wsl --list --verbose
     ```

     Submit a screenshot showing Ubuntu installed and running on WSL2.




2. **Update Ubuntu environment**
   - Run the following command in Ubuntu:

     ```
     sudo apt update && sudo apt upgrade -y
     ```




---

# Part B: Git & GitHub SSH Setup

Submit a screenshot.

2. **Generate SSH Keys**
   - Run:

     ```
     ssh-keygen -t ed25519
     ```

   - Copy the public key:

     ```
     cat ~/.ssh/id_ed25519.pub
     ```

   - Add this key to your GitHub account under **Settings → SSH and GPG keys**.
3. **Test Connection**

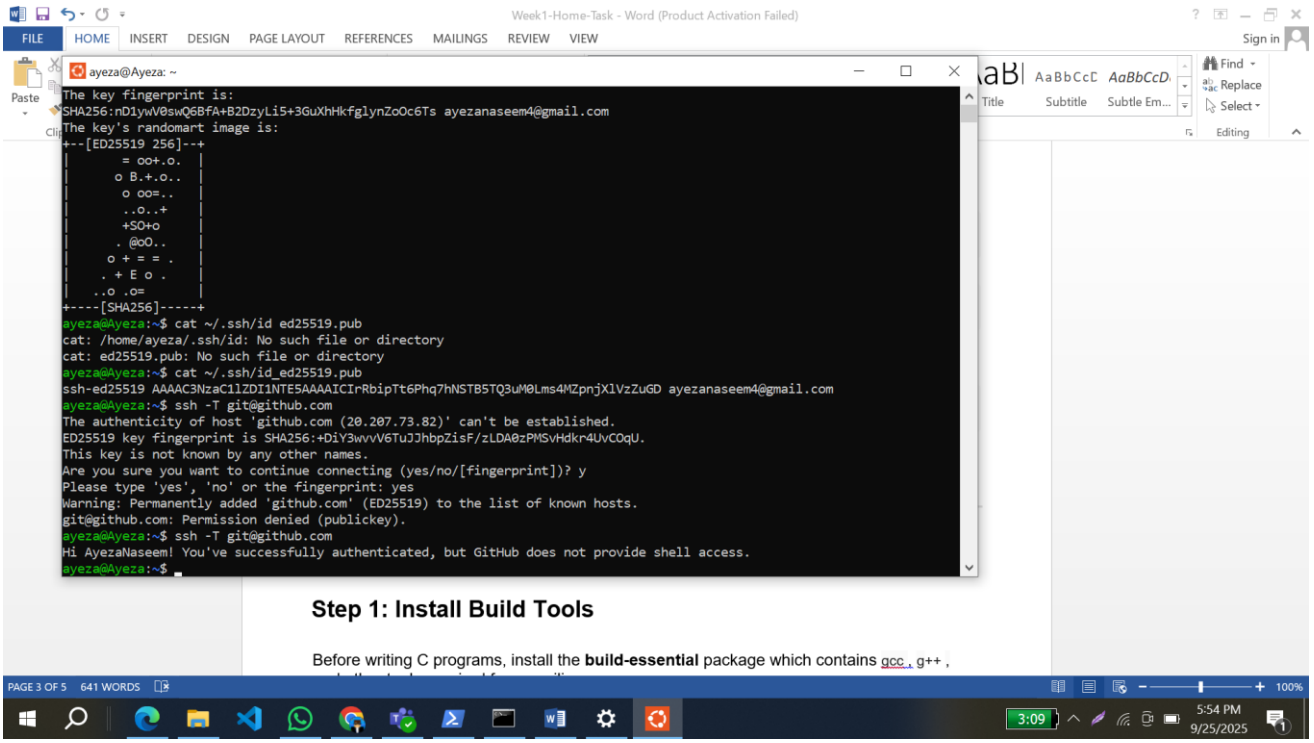
   ```
   ssh -T git@github.com
   ```

   - Submit a screenshot showing successful authentication.
1. **Configure Git**
   - Set your name and email:

     ```
     git config --global user.name "Your Name"
     git config --global user.email "your@email.com"
     ```

   - Show your config:

     ```
     git config --list
     ```




# Part C: C Programming Environment & Practice

## Step 1: Install Build Tools

Before writing C programs, install the **build-essential** package which contains `gcc` , `g++` , and other tools required for compiling.

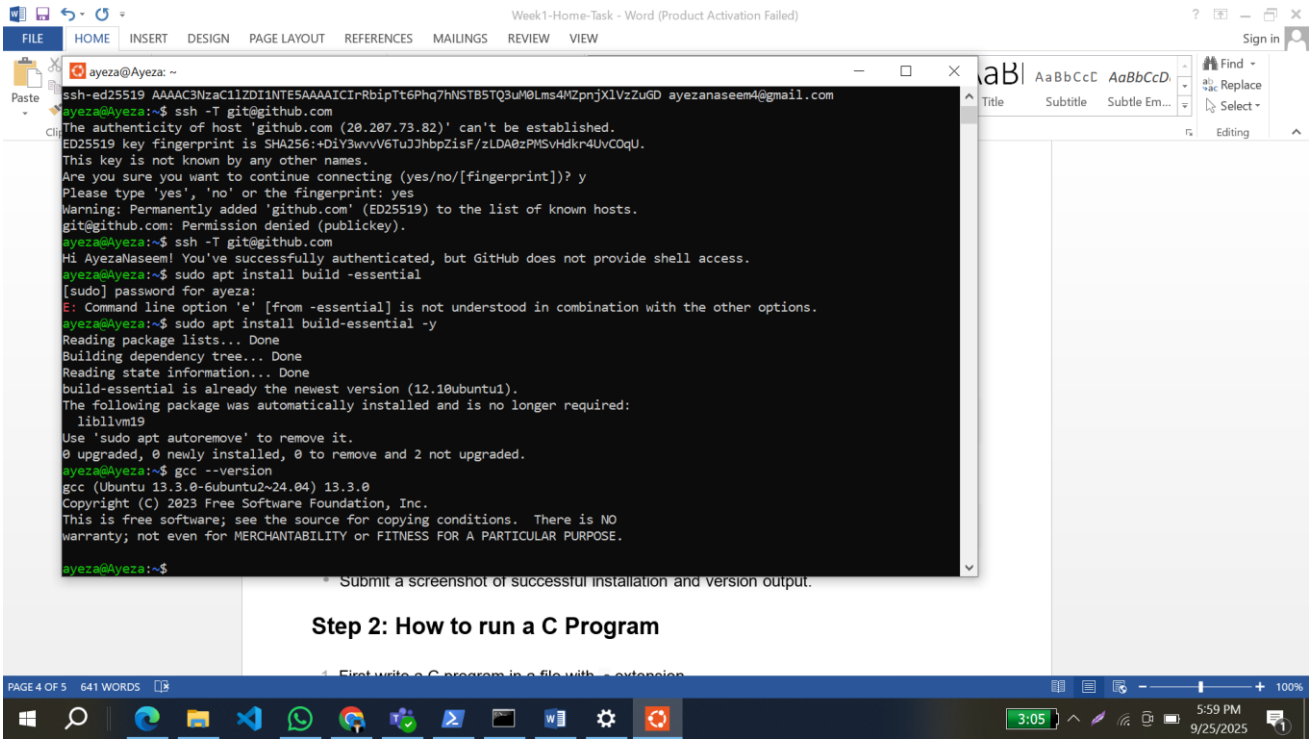
Run:

```
sudo apt install build-essential
```

Verify installation by checking the version of `gcc` :

```
gcc --version
```

- Submit a screenshot of successful installation and version output.




# Step 2: How to run a C Program

1. First write a C program in a file with `.c` extension.
2. Compile the file using `gcc filename.c -o filename.out`
3. Execute it using `./filename.out`

### Breakdown

- `gcc`
  - This is the GNU Compiler Collection command.
  - It compiles C (and other languages like C++) programs into machine code that can be executed by the computer.
- `filename.c`
  - This is the source code file you wrote in C.
  - Example: hello.c contains your C program.
- `-o filename.out`
  - The option -o means “output.”
  - By default, gcc creates an executable file named a.out if you don’t specify anything.
  - With -o, you can choose the name of the output executable.

In this case, the compiled file will be named filename.out.

## Step 3: Write a C Program

Write a simple C program of your choice. It can be a **Hello World** program or any other.