Operating Systems – COC 3071L

SE 5th A – Fall 2025

Objective

The purpose of this assignment is to:

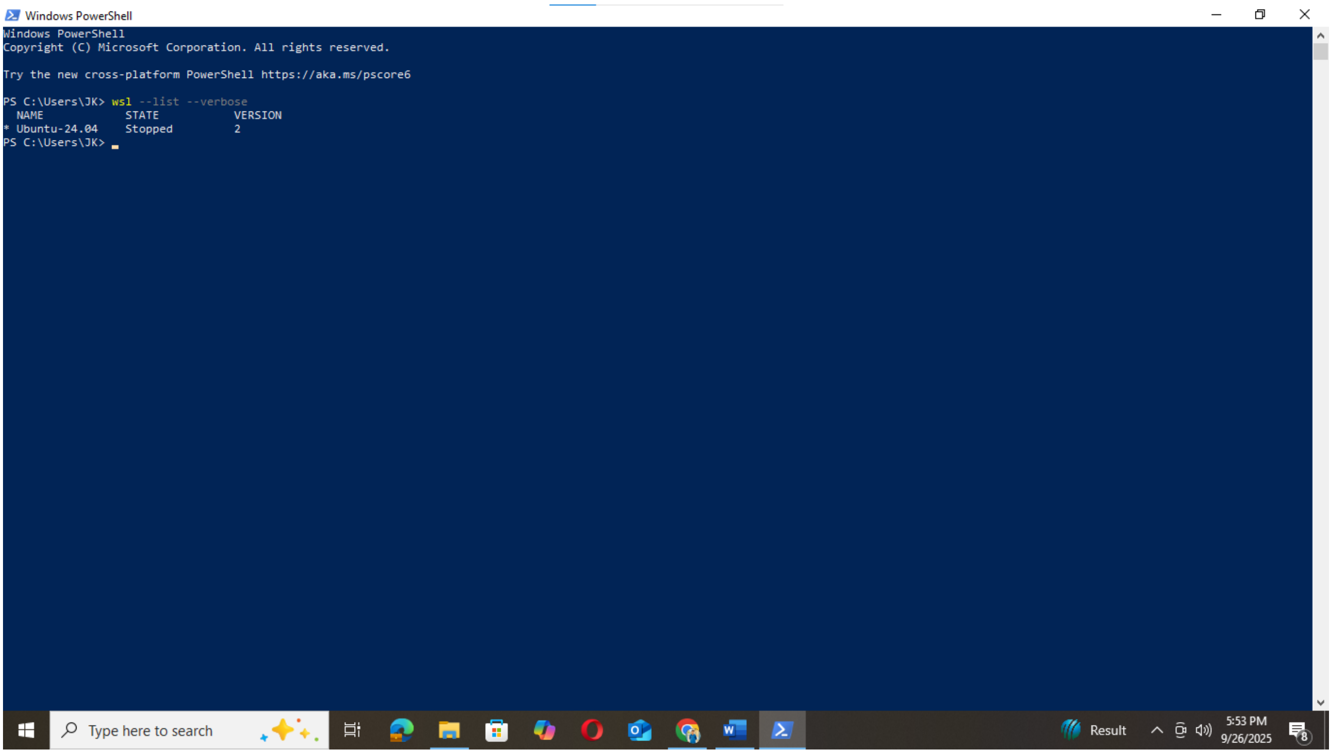
1. Configure **Ubuntu** inside **WSL2 (Windows Subsystem for Linux v2)**.
2. Install and configure **Git** in Ubuntu.
3. Generate and set up **SSH keys** to connect with GitHub.
4. Install the **C development environment** in Ubuntu.
5. Write a **Hello World** program in `C`.

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.

   ```
   Windows PowerShell
   Copyright (C) Microsoft Corporation. All rights reserved.

   Try the new cross-platform PowerShell https://aka.ms/pscore6

   PS C:\Users\JK> wsl --list --verbose
     NAME            STATE           VERSION
   * Ubuntu-24.04    Stopped         2
   PS C:\Users\JK>
   ```

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

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

Part B: Git & GitHub SSH Setup

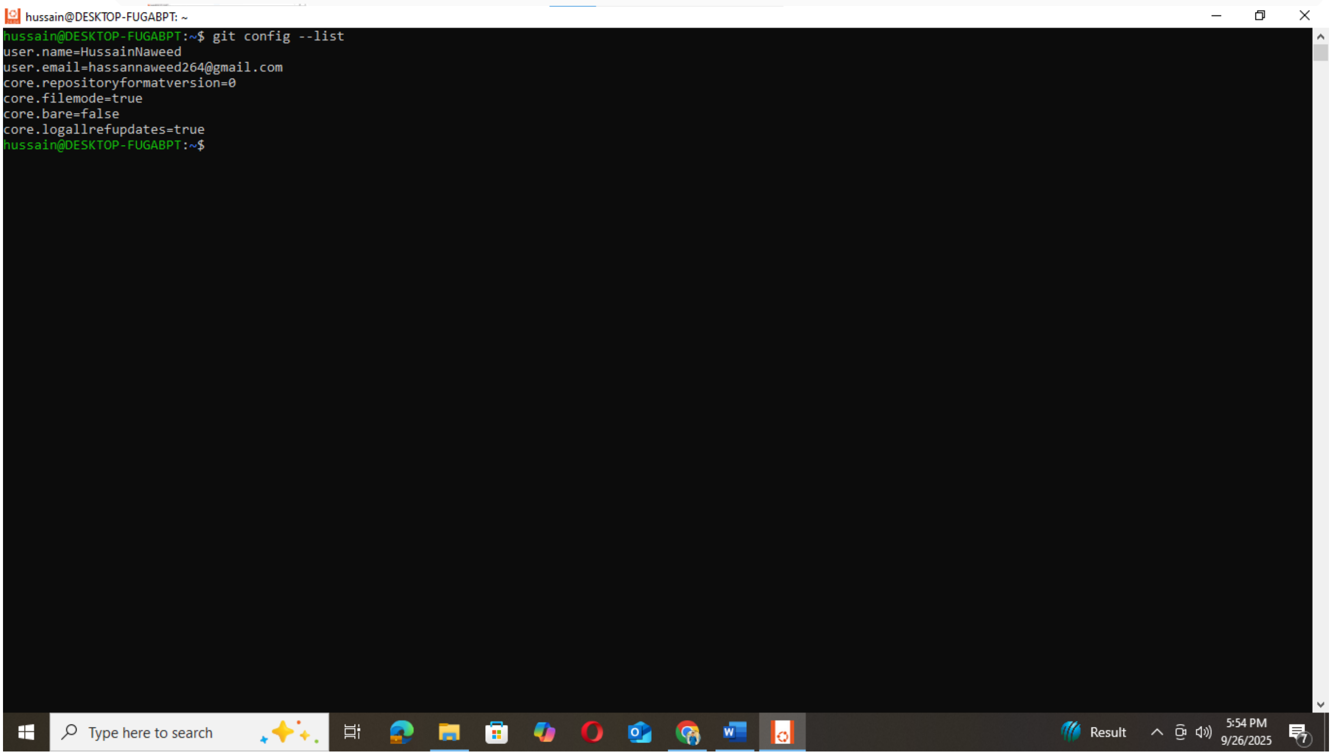
1. **Configure Git**

   - 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
     ```



hussain@DESKTOP-FUGABPT: ~

```
hussain@DESKTOP-FUGABPT:~$ git config --list
user.name=HussainNaweed
user.email=hassannaweed264@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
hussain@DESKTOP-FUGABPT:~$
```

Submit a screenshot.

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

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

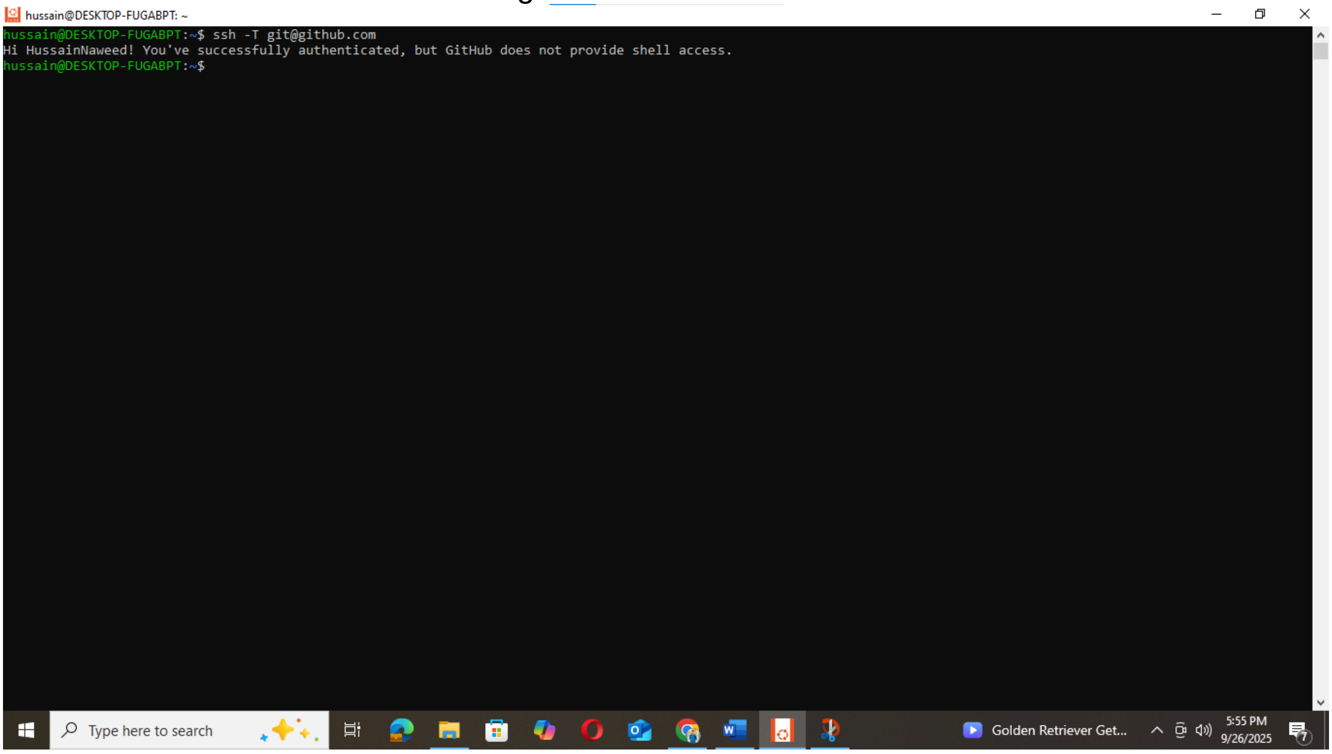
   - Copy the public key:

     ```
     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.



```
hussain@DESKTOP-FUGABPT:~$ ssh -T git@github.com
Hi HussainNaweed! You've successfully authenticated, but GitHub does not provide shell access.
hussain@DESKTOP-FUGABPT:~$
```

Part C: C Programming Environment & Practice

Step 1: Install Build Tools

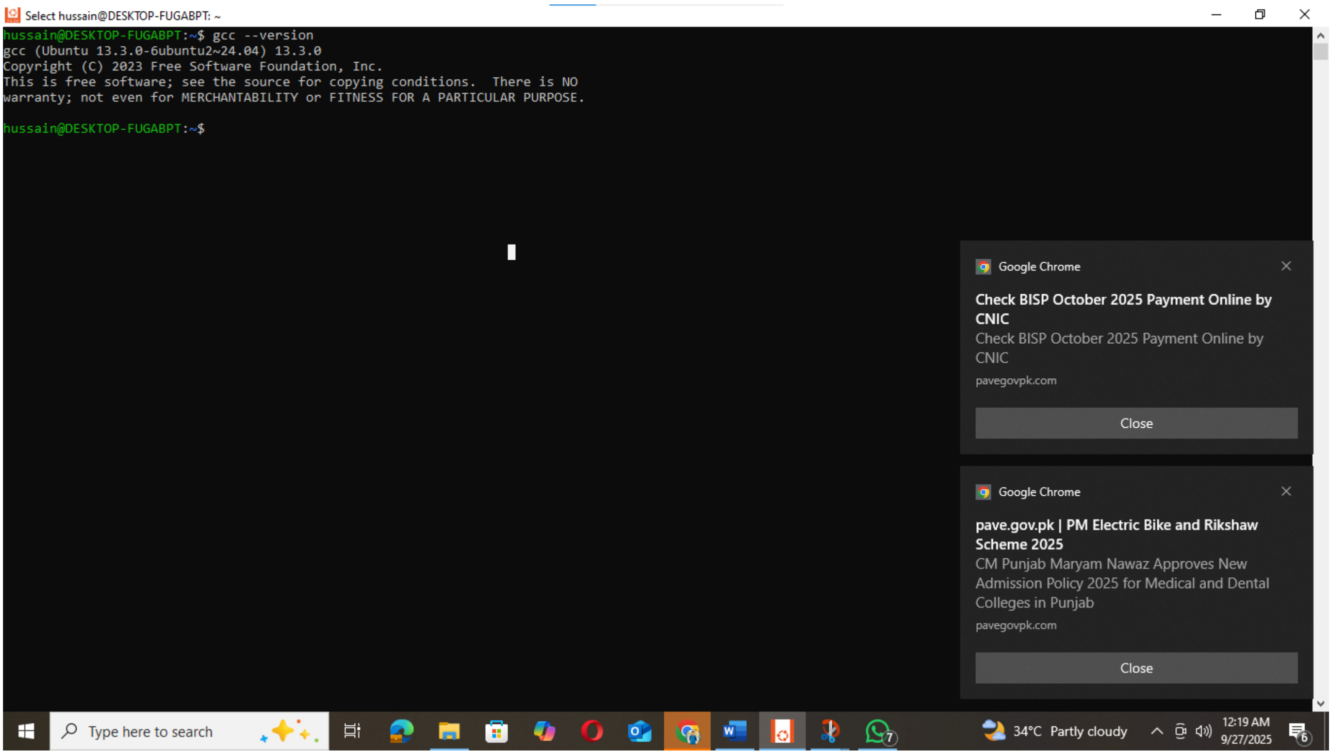
Before writing C programs, install the **build-essential** package which contains `gcc`, `g++`, and other tools required for compiling.

Run:

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

Verify installation by checking the version of `gcc`:

```
gcc --version
```



```
hussain@DESKTOP-FUGABPT:~$ gcc --version
gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

hussain@DESKTOP-FUGABPT:~$
```

Google Chrome

Check BISP October 2025 Payment Online by CNIC

Check BISP October 2025 Payment Online by CNIC

pavegovpk.com

Close

Google Chrome

pave.gov.pk | PM Electric Bike and Rikshaw Scheme 2025

CM Punjab Maryam Nawaz Approves New Admission Policy 2025 for Medical and Dental Colleges in Punjab

pavegovpk.com

Close

- Submit a screenshot of successful installation and version output.

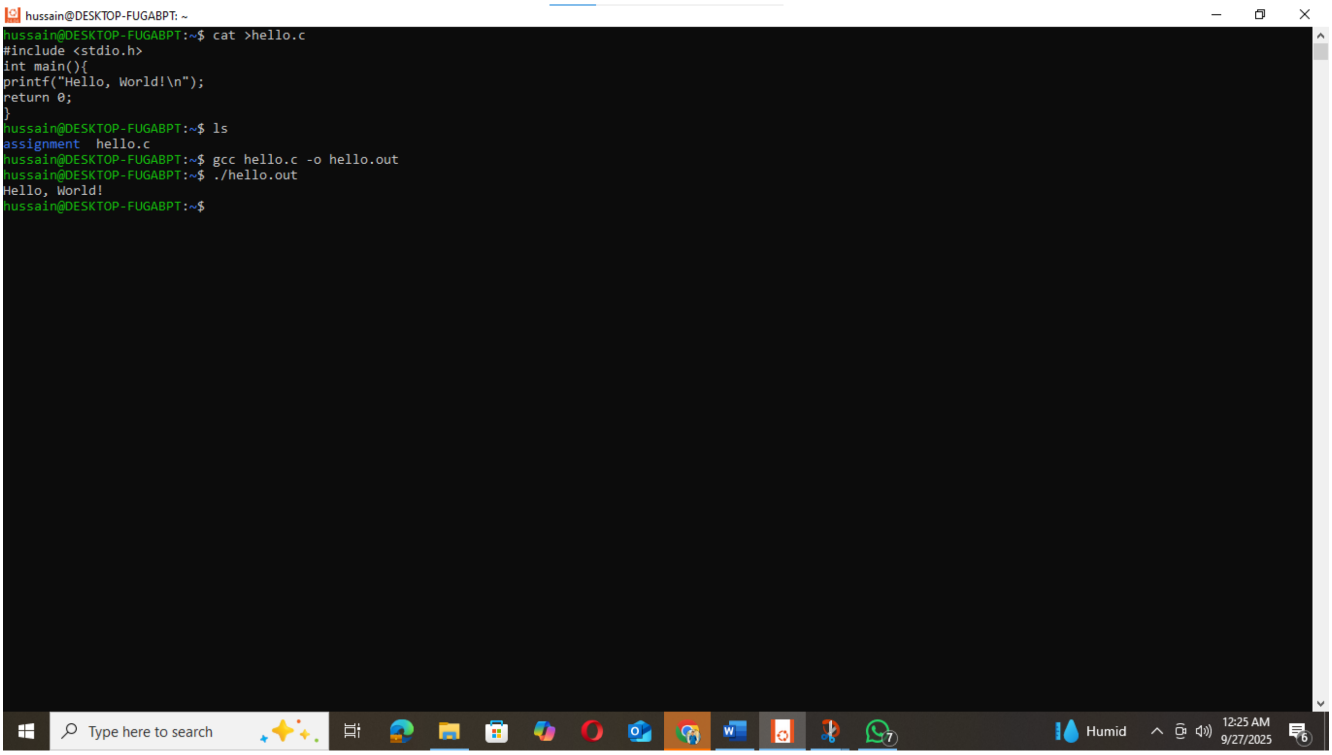
Step 2: How to run a C Program

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.

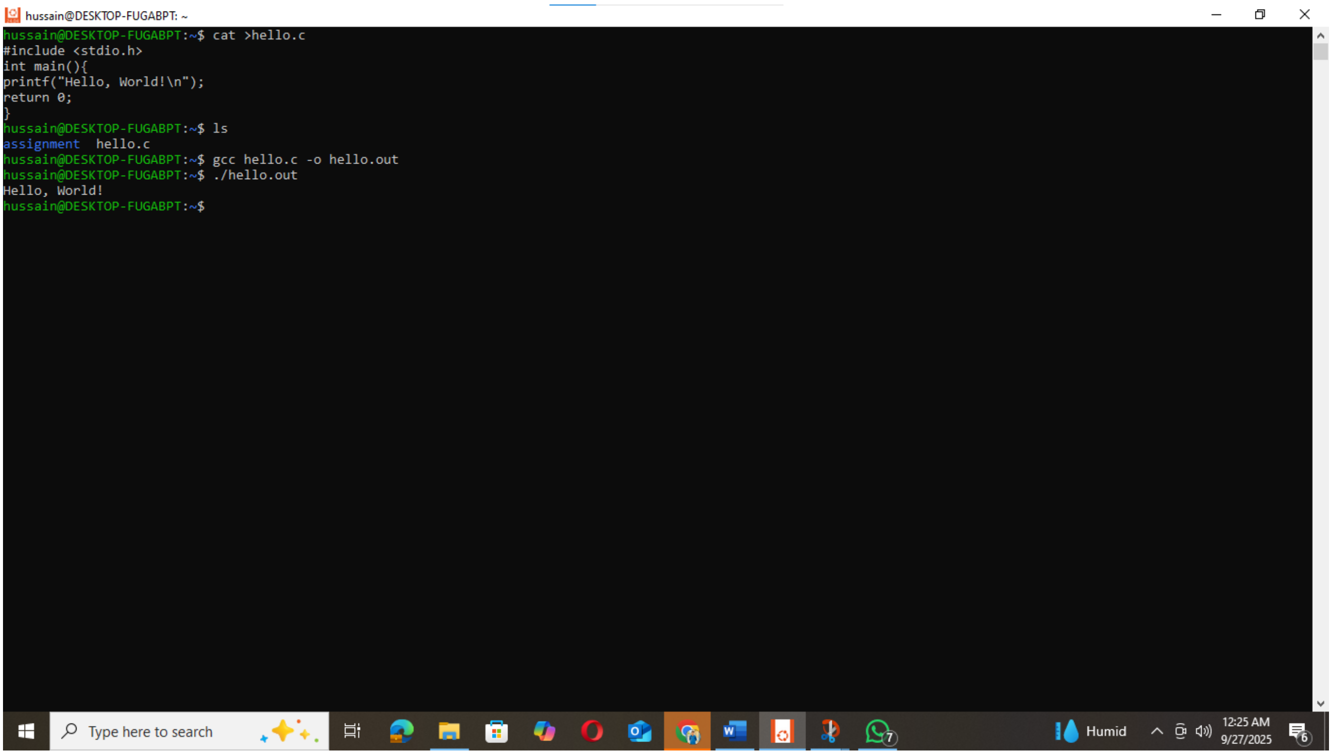


```
hussain@DESKTOP-FUGABPT:~$ cat >hello.c
#include <stdio.h>
int main(){
printf("Hello, World!\n");
return 0;
}
hussain@DESKTOP-FUGABPT:~$ ls
assignment  hello.c
hussain@DESKTOP-FUGABPT:~$ gcc hello.c -o hello.out
hussain@DESKTOP-FUGABPT:~$ ./hello.out
Hello, World!
hussain@DESKTOP-FUGABPT:~$
```

Step 3: Write a C Program

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

> **Submission:** For the program, submit:
>
> - The C source code (`.c` file).
> - Screenshot of execution (`./program`)



```
hussain@DESKTOP-FUGABPT:~$ cat >hello.c
#include <stdio.h>
int main(){
printf("Hello, World!\n");
return 0;
}
hussain@DESKTOP-FUGABPT:~$ ls
assignment  hello.c
hussain@DESKTOP-FUGABPT:~$ gcc hello.c -o hello.out
hussain@DESKTOP-FUGABPT:~$ ./hello.out
Hello, World!
hussain@DESKTOP-FUGABPT:~$
```

Deadline

- Submit before **12:00 AM, 25 September, 2025**.