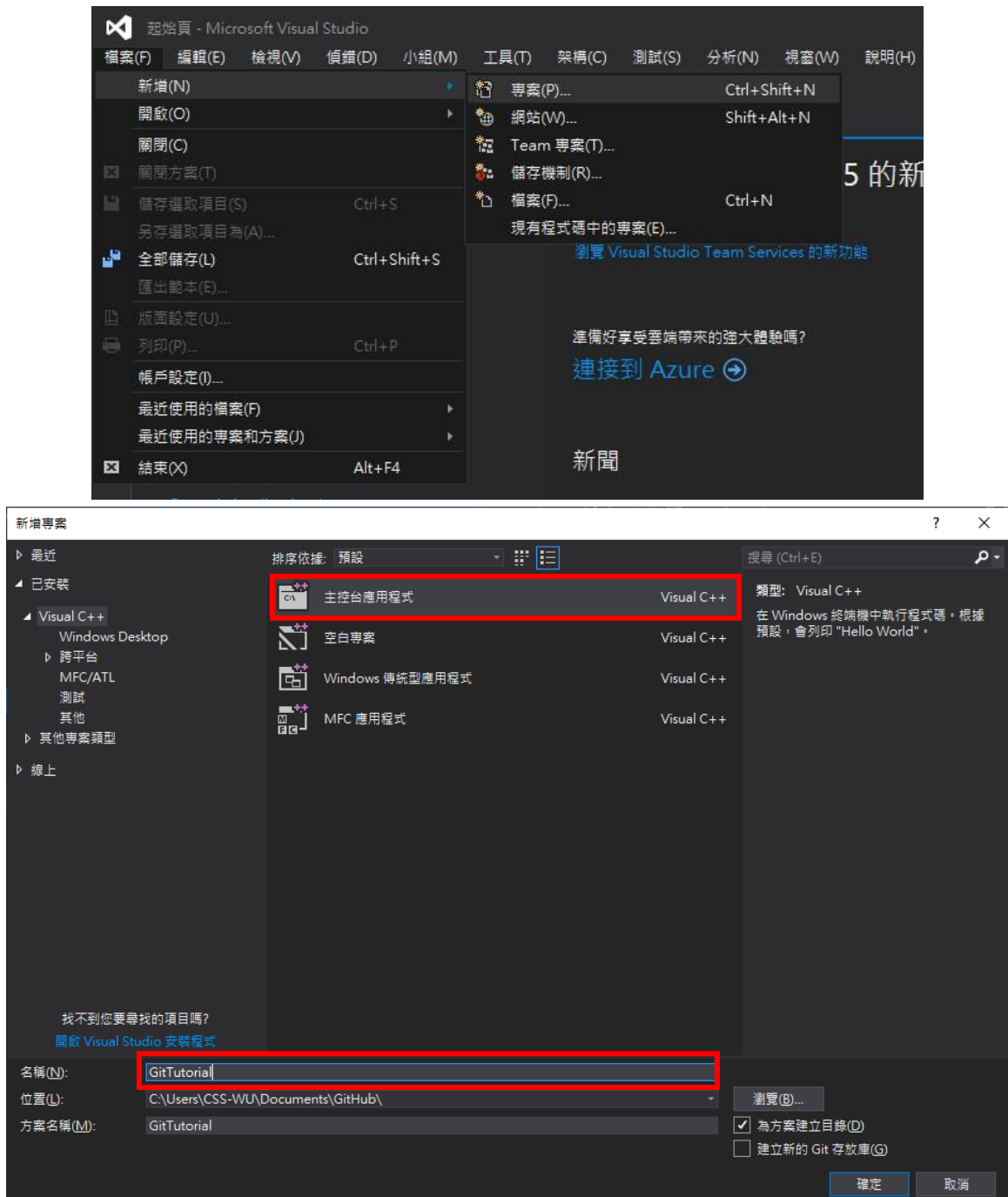


練習 5: Commit/Push 程式到伺服器

Commit 為提交已 stage 的檔案至 Git，Commit 僅提交到本機，Push 以後才會上傳到伺服器，Commit 建議每做一點修改就 Commit 一次，一口氣 Commit 大量檔案可能發生未預期的問題；本練習的目的是由某乙先寫一段程式，並將程式 Commit/Push 到 git 上，使得以後某甲可以從 git 上取得這段程式並且做修改。

1. 在練習 3 Clone 的目錄下，新增一個 Visual C++ 主控台應用程式



2. 在 GitTutorial.cpp 貼上下方程式碼並編譯執行：

```
#include <string>
#include <iostream>
#include <cstdlib>
#include <ctime>

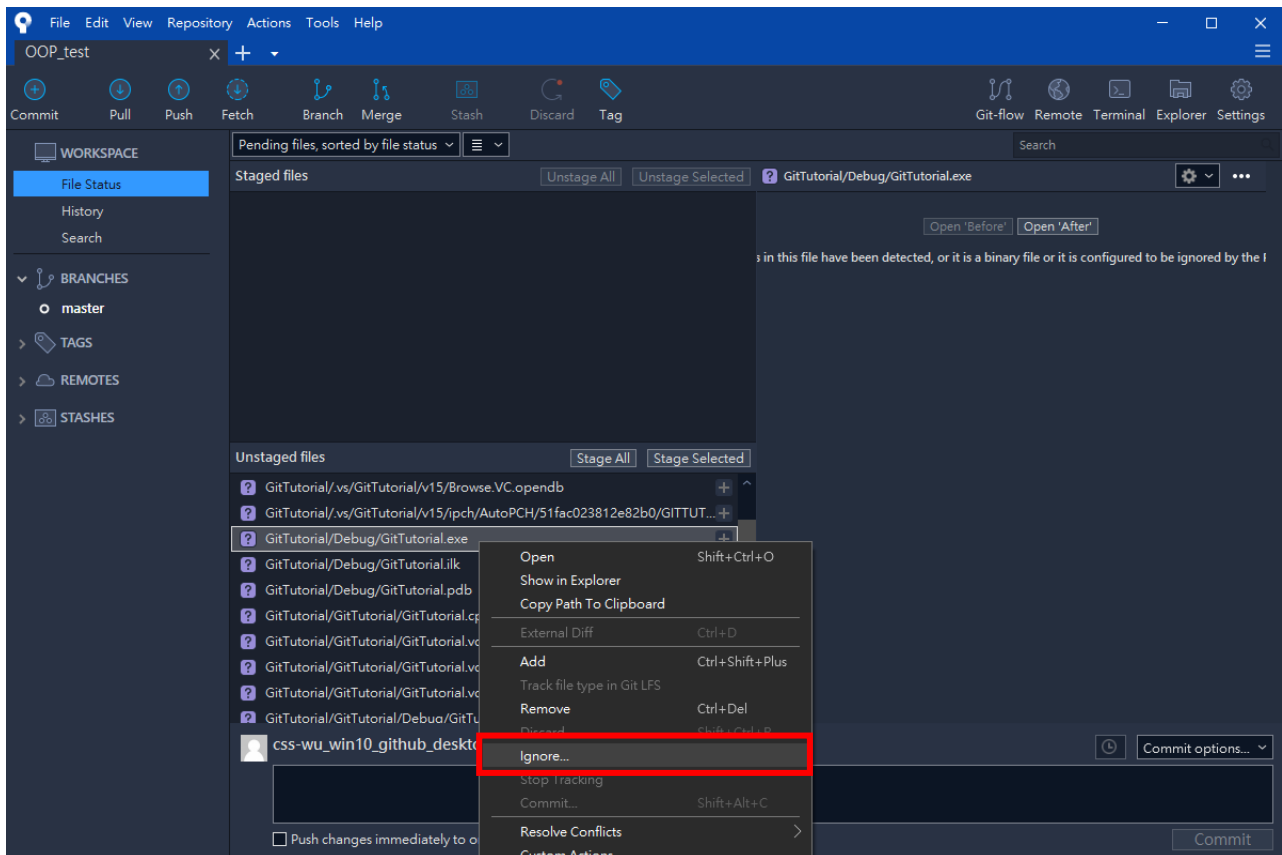
bool Guess(int number) {
    return false;
}

int main()
{
}
```

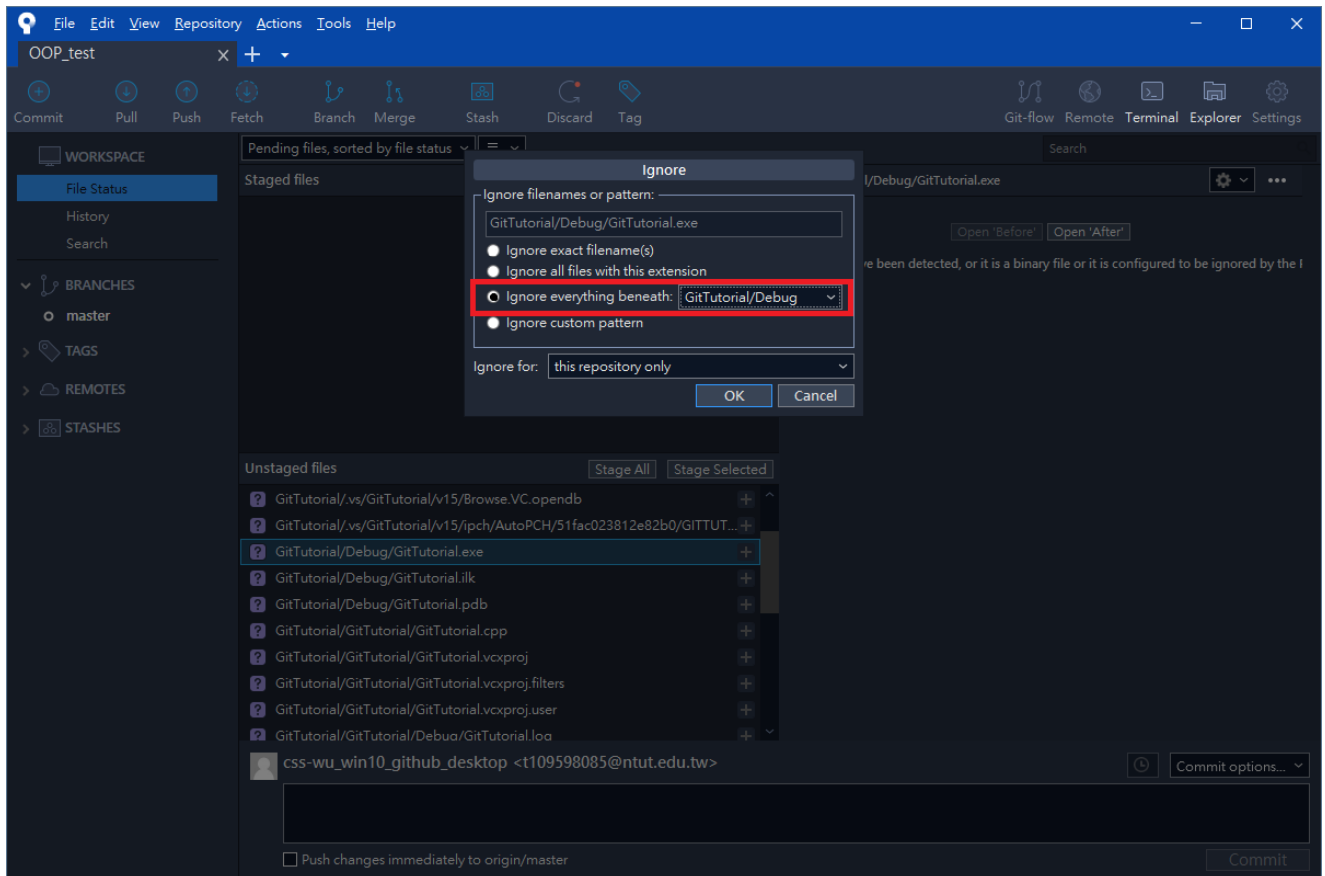
3. Gitignore :

(Gitignore 可以設定檔案或路徑使其不再被 Git 追蹤，詳細可以使用<https://www.gitignore.io/>，輸入所使用的IDE 及語言，會自動產生 ignore 檔，且在註解有說明該檔案或目錄用途為何)

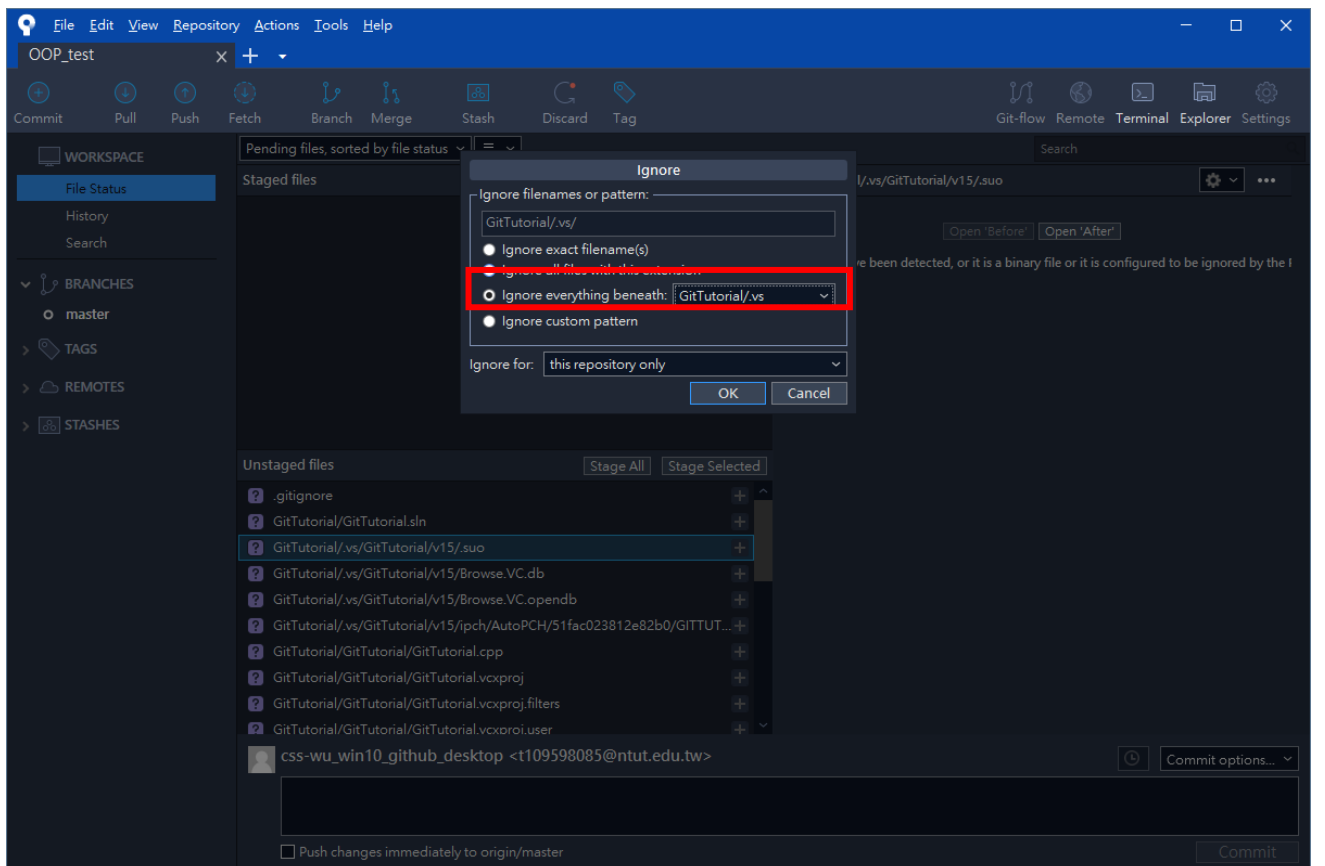
3-1. 在 unstaged file 對一個在GitTutorial/Debug路徑的檔案右鍵，選擇 ignore (有編譯執行過才會有Debug資料夾的檔案出現)



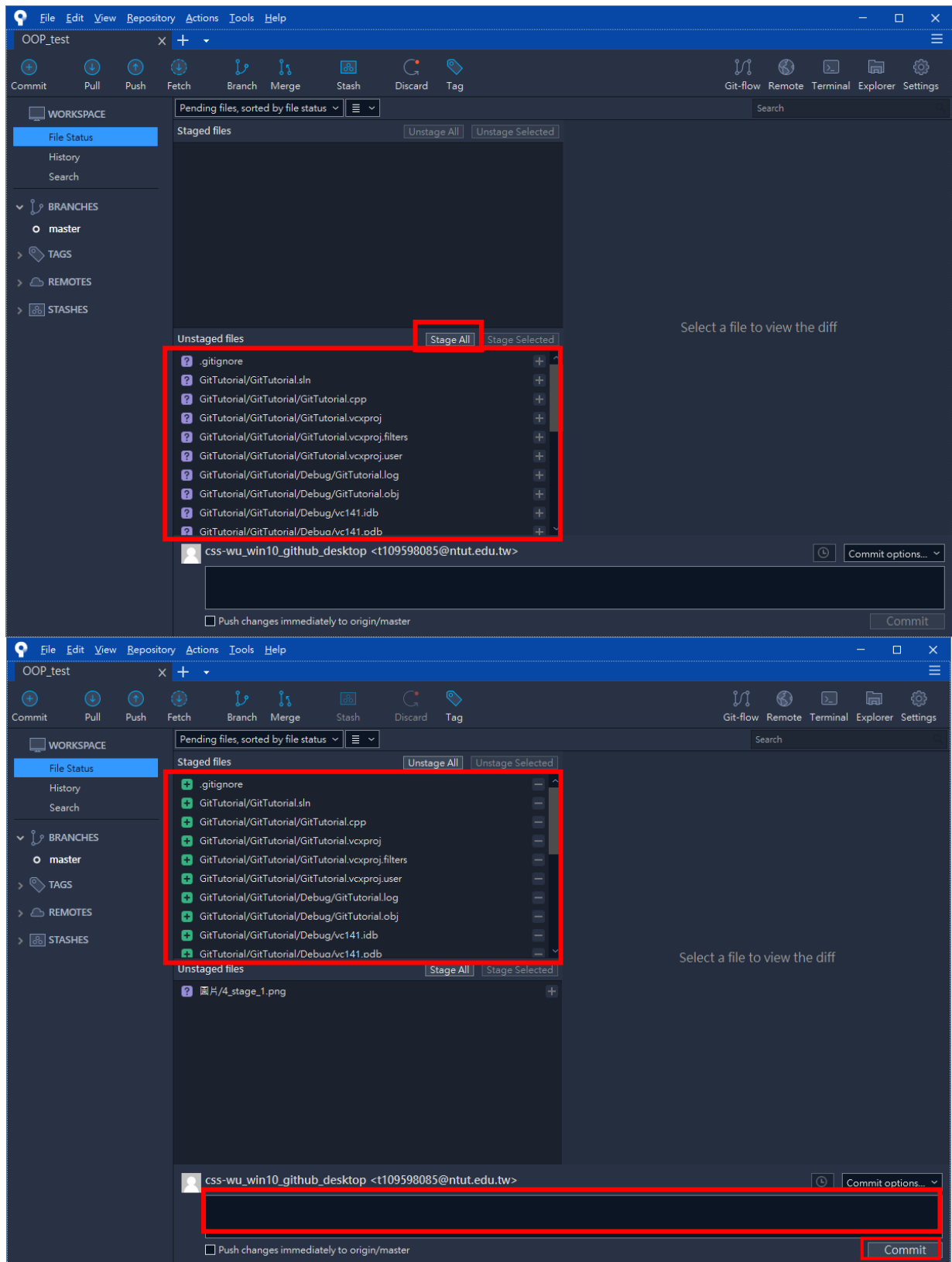
3-2. ignore 所有Debug 目錄下的所有檔案



3-3. 用一樣的方法忽略GitTutorial/.vs/

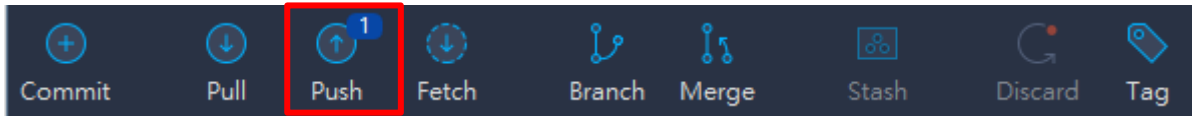


4. 按下Stage All或是打勾你要的檔案，接著填上你的 commit log 並按下 commit，commit log 請填寫有意義的 Message

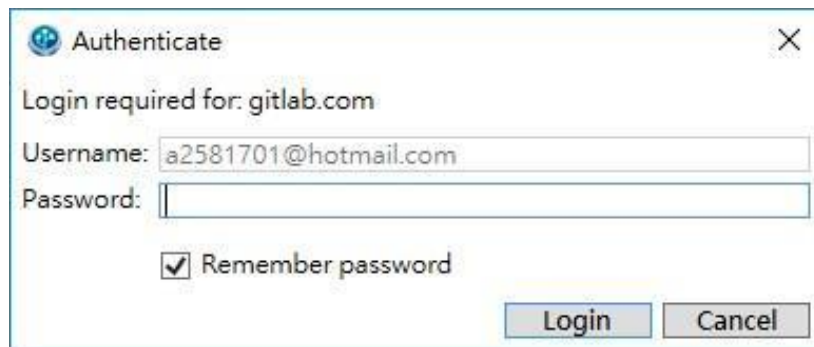


(填寫清楚的 log，可以讓你快速知道每個 commit 的版本差異)

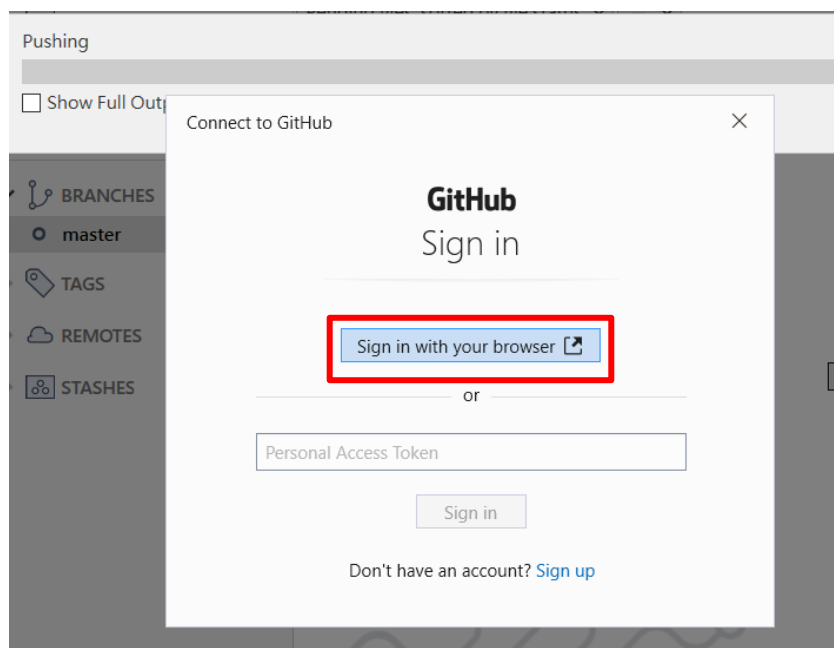
5. Commit 完之後如果有伺服器上有新的工作副本，Push 按鈕會出現一則commit 未 push，按下 push 之後可能需要再次輸入帳號密碼接著就成功的上傳到遠端伺服器上



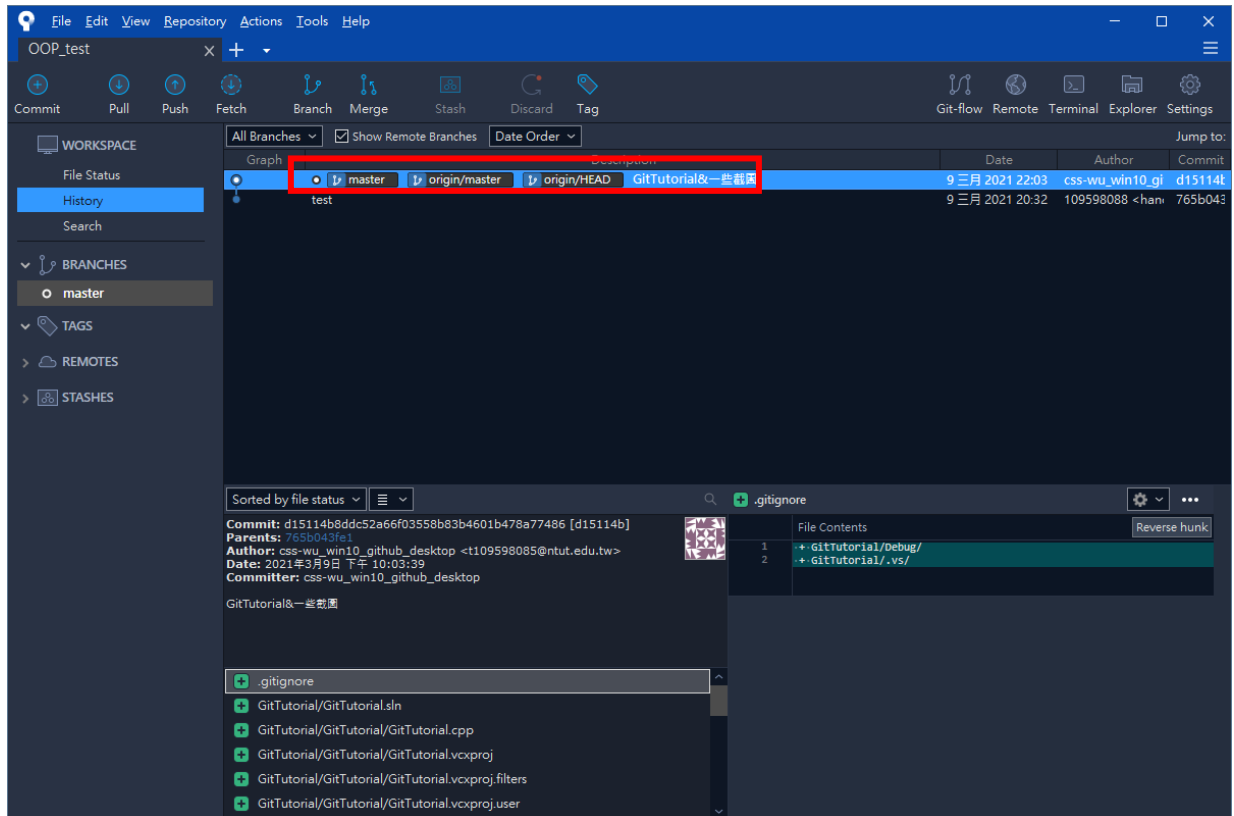
如果有出現類似視窗，請輸入你的 Github 帳號密碼



若僅出現下列視窗，點擊 sign in 以用瀏覽器登入Github 帳號



6. 成功上傳至遠端伺服器

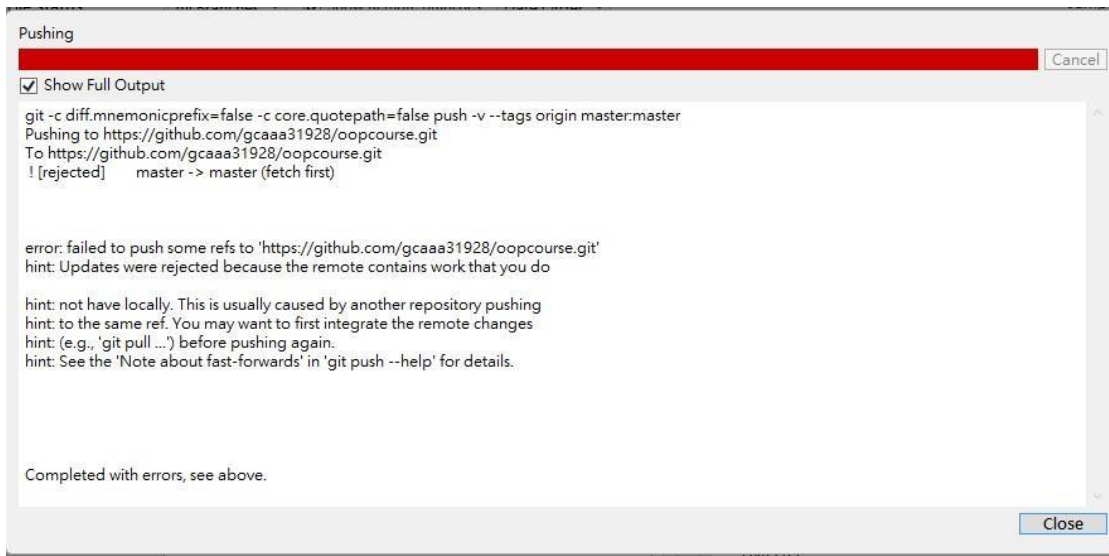


(master(本機檔案)與 origin/master(伺服器檔案)會在同一條 commit 上)

課程上使用 *Git* 時，請將檔案大小控制在 500MB 以內，
否則伺服器容易發生問題，如果圖片過多，可以
透過<https://tinypng.com/>壓縮圖片

Appendix :

如果無法成功 push 至伺服器，且出現以下的 error message



發生原因可能為練習時步驟有誤，可參考練習 8-乙解決