GitHub Classroom
Auto-grading

In C programming language

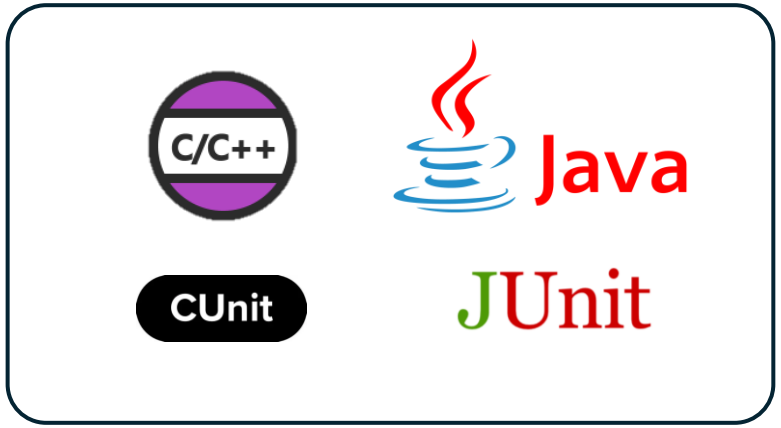
09-2025

Nhóm thực hiện
- Thân Hoàng Lộc
- Trần Quang Bình
- Trần Văn Huy
- Trần Anh Tuấn

Mục tiêu

Kiểm tra nội dung và Chấm điểm tự động
bài làm của sinh viên
đã nộp trên phần mềm GitHub Classroom

Nhóm 1



Nhóm 2

Nhóm 3

Nội dung Nhóm 1 (C/C++)

1. Giới thiệu
2. MinGW, CUnit
3. VS Code Configure C/C++ Compiler – tasks.json
4. VS Code Configure C/C++ Debugger – launch.json
5. Application & Unit testing
6. Makefile & Build command & Test runner
7. Auto-grading run command
8. Auto-grading Reporter

Giới thiệu GitHub Classroom Auto-grading

Tính năng Auto-grading, để kiểm tra và chấm điểm tự động

nội dung bài làm của sinh viên đã nộp trên GitHub Classroom

Giảng viên

- Cấu hình chế độ kiểm tra:
  ngay lập tức hay định kỳ

- Xem báo cáo tổng quát:
  tình hình nộp bài và
  tình trạng bài nộp ✓ Passed hay ✗ Failed

Sinh viên

- Xem và biết kết quả:
  bài đã nộp ✓ Passed hay ✗ Failed

Giới thiệu GitHub Classroom Auto-grading

Có 3 nhóm phương pháp Auto-grading

run command tests

CUnit

MinGW w64

JUnit

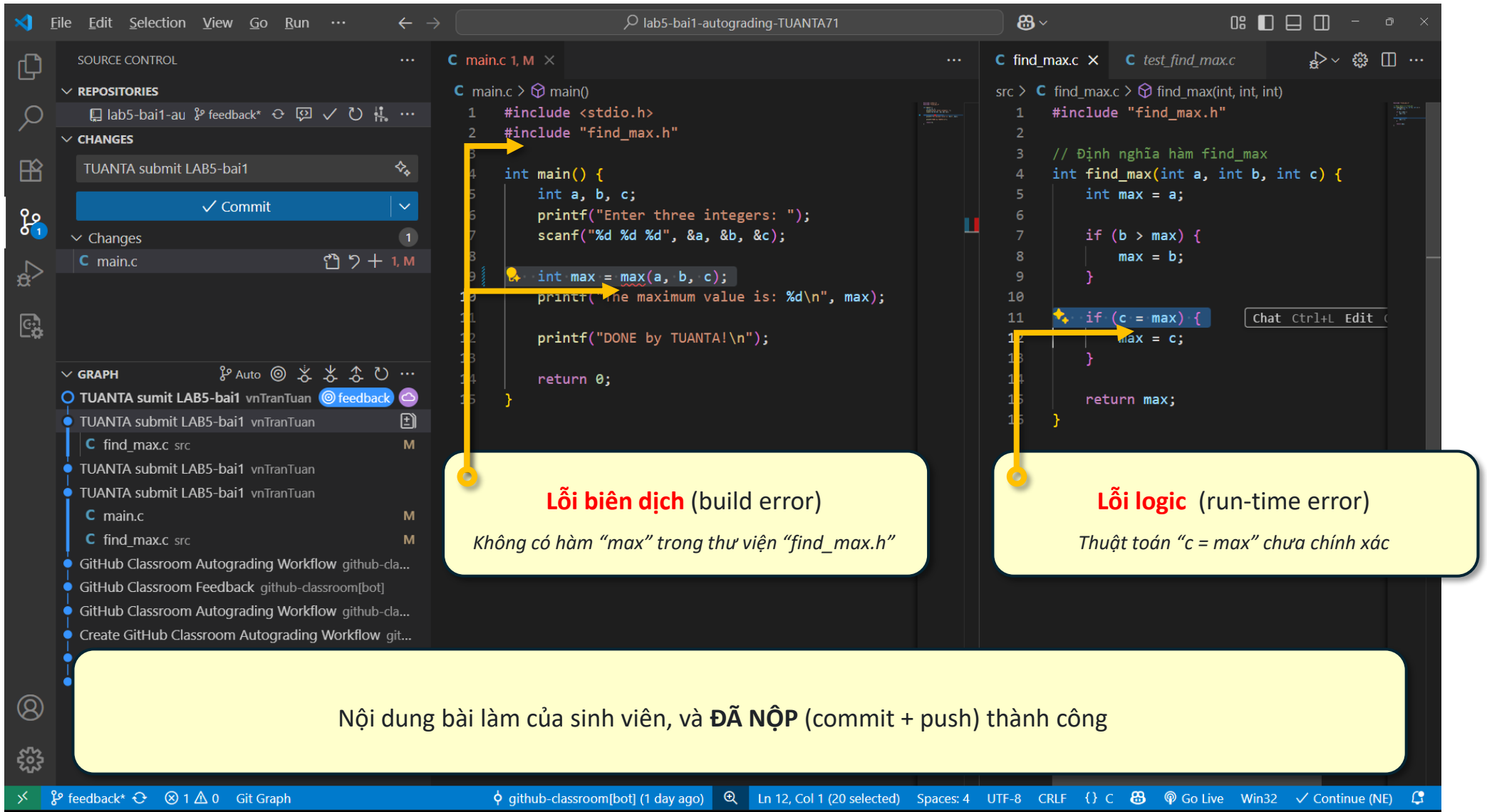
python tests

pytest

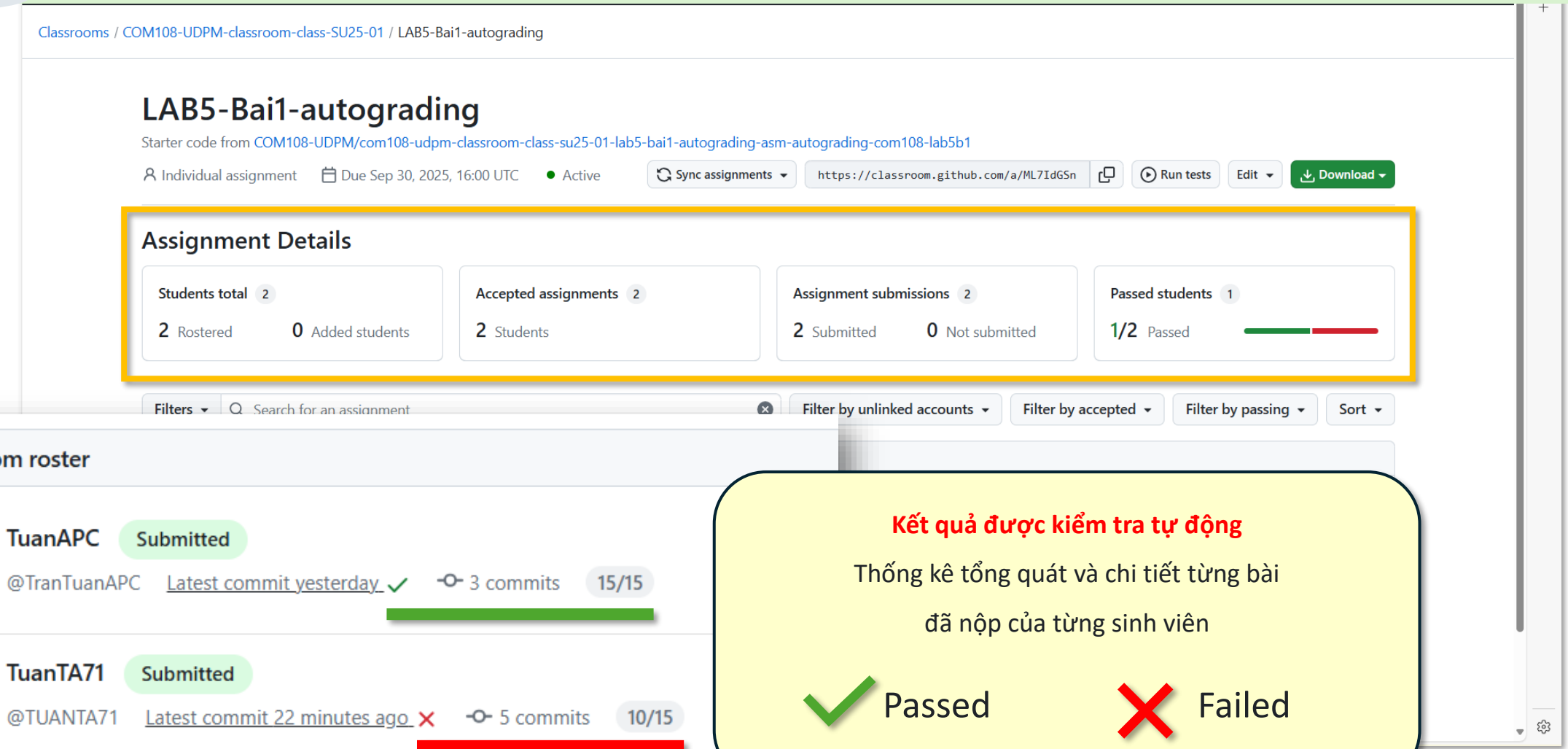
Input/Output tests

BASH & Shell Scripts

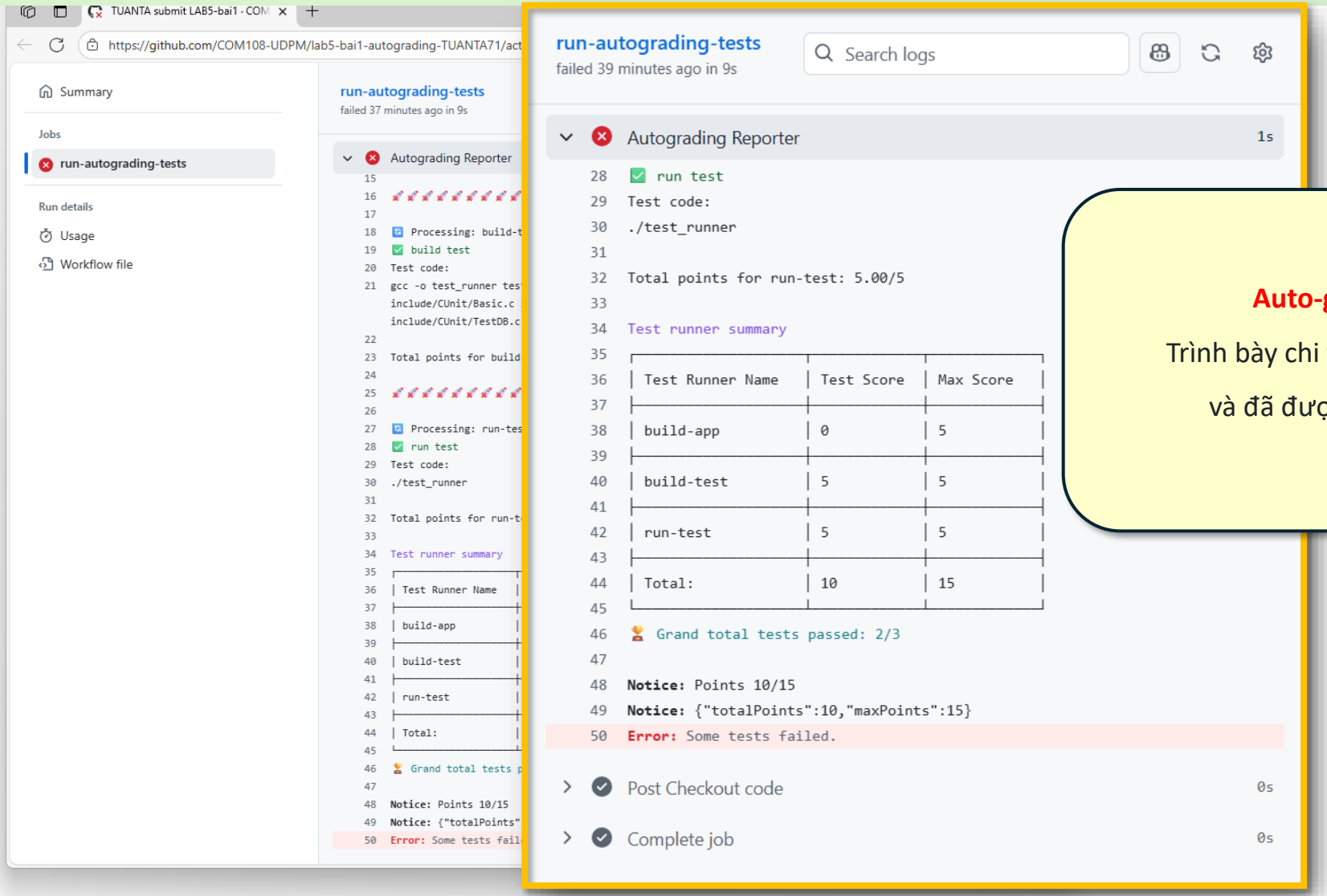
Giới thiệu GitHub Classroom Auto-grading



Giới thiệu GitHub Classroom Auto-grading



Giới thiệu GitHub Classroom Auto-grading

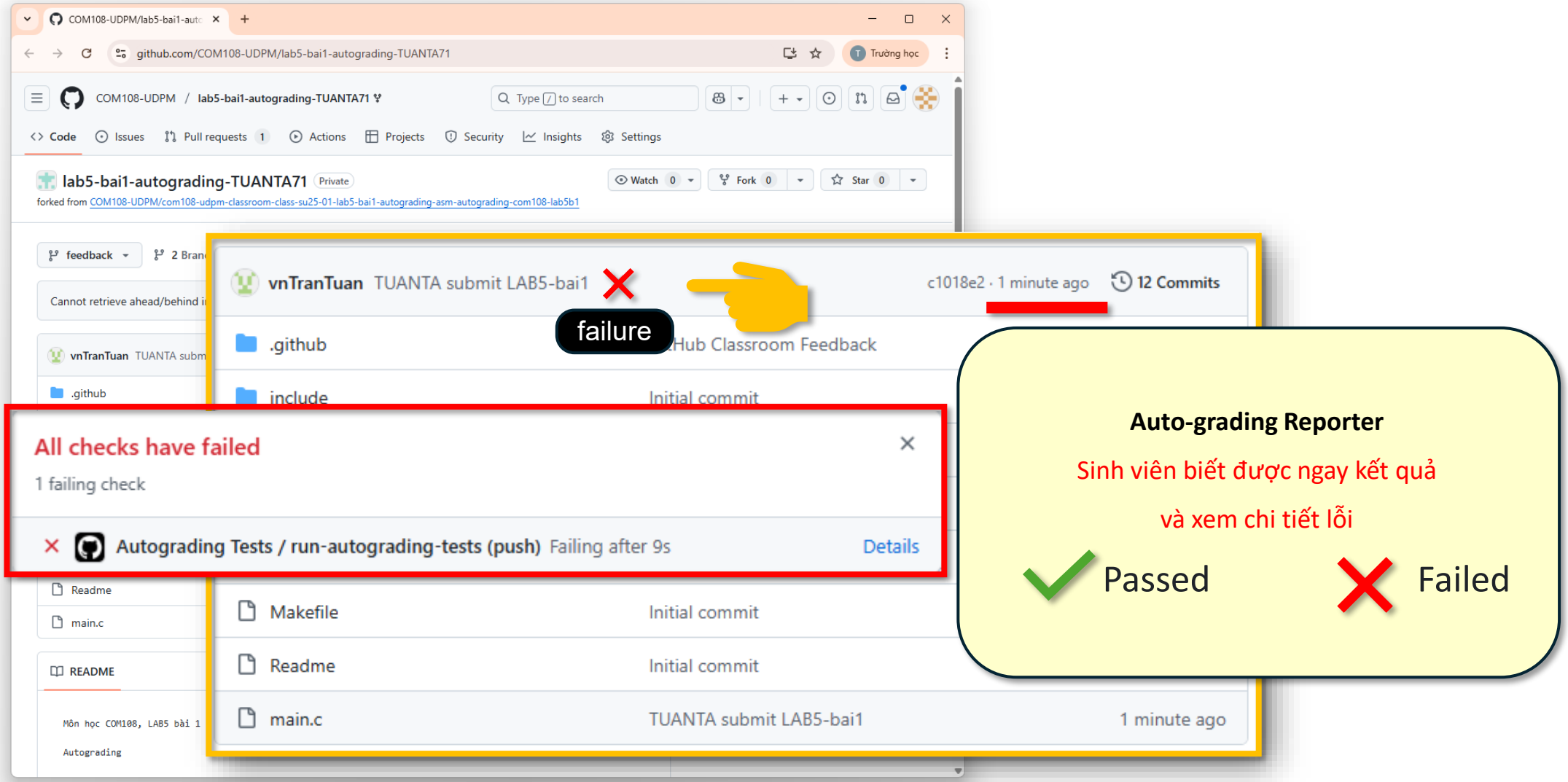


Auto-grading Reporter

Trình bày chi tiết các nội dung đã test

và đã được thực hiện **tự động**

Giới thiệu GitHub Classroom Auto-grading



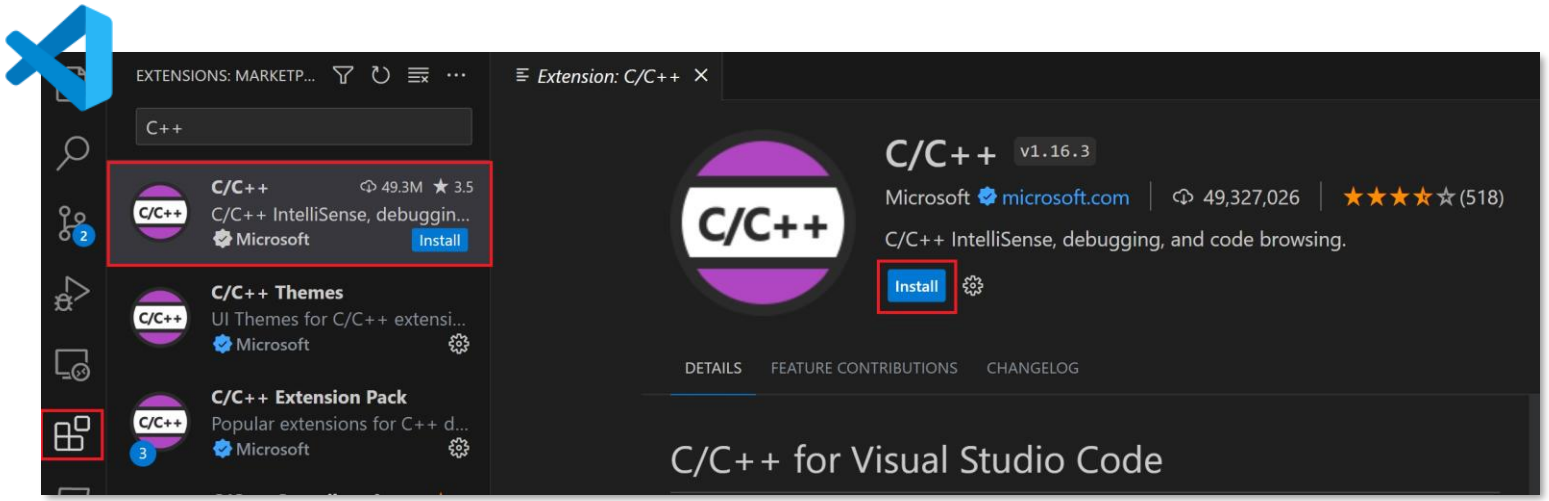
MinGW: *MinGW-w64 (64bit, 32bit), MinGW (32bit)*

Tham khảo cài đặt MinGW-w64, tại https://code.visualstudio.com/docs/cpp/config-mingw#_prerequisites

MinGW-w64
64bit, 32bit

→ **Compiler** **gcc / g++**
Debugger **gdb**

MinGW-w64
<installed folder>
gcc.exe
g++.exe
gdb.exe



CUnit A unit testing framework for C

Tham khảo tại https://cunit.sourceforge.net

Dùng thư viện CUnit để thực hiện 2 việc sau

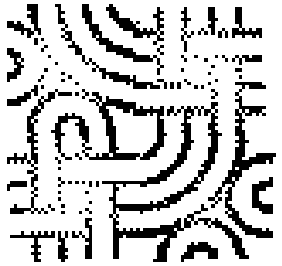
1

Tạo các test case
cho chương trình
(lab / assignment)

2

Tạo các auto-grading
run command
mỗi khi repository commit

Các thư viện khác



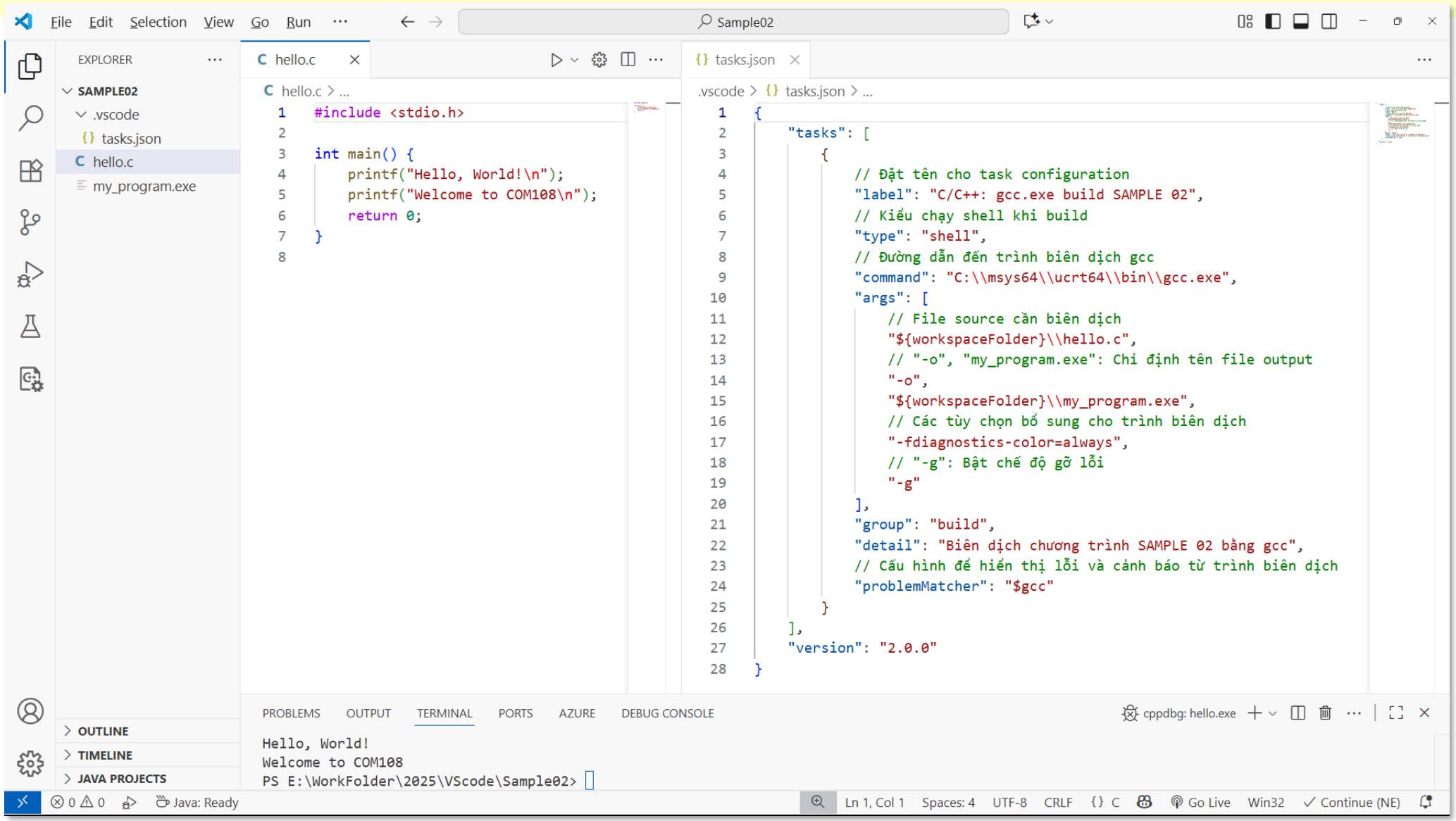
CppUnit Testing Library

https://cppunit.sourceforge.net/doc/1.8.0/index.html

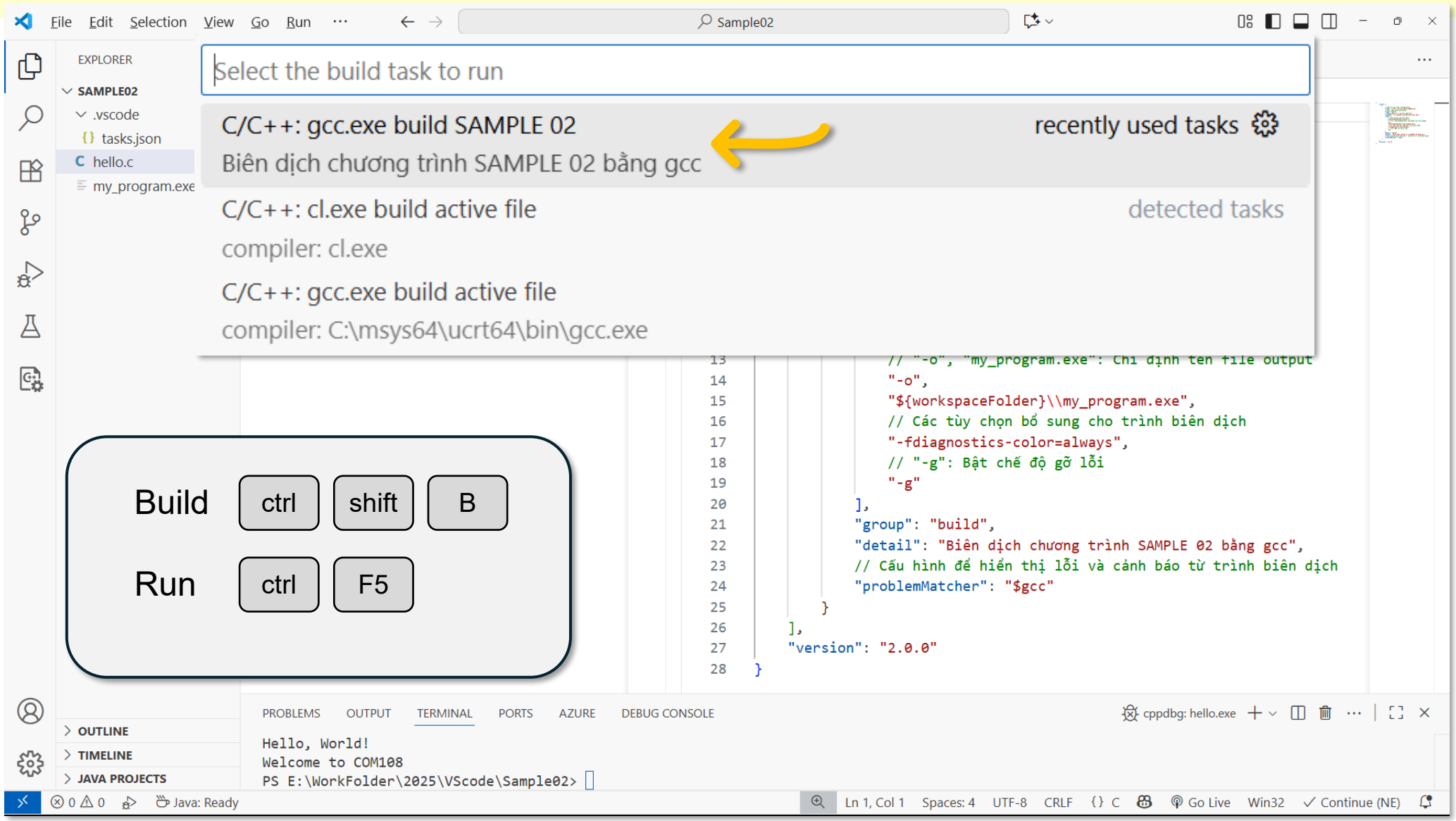
C# Dev Kit

https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csdevkit

VS Code – tasks.json (build configuration)



VS Code – tasks.json (build configuration)



VS Code – tasks.json (build configuration)

.vscode > {} tasks.json > ...

```jsonc
{
    "tasks": [
        {
            // Đặt tên cho task configuration
            "label": "C/C++: gcc.exe build SAMPLE 02",
            // Kiểu chạy shell khi build
            "type": "shell",
            // Đường dẫn đến trình biên dịch gcc
            "command": "C:\\msys64\\ucrt64\\bin\\gcc.exe",
            "args": [
                // File source cần biên dịch
                "${workspaceFolder}\\hello.c",
                // "-o", "my_program.exe": Chỉ định tên file c
                "-o",
                "${workspaceFolder}\\my_program.exe",
                // Các tùy chọn bổ sung cho trình biên dịch
                "-fdiagnostics-color=always",
                // "-g": Bật chế độ gỡ lỗi
                "-g"
            ],
            "group": "build",
            "detail": "Biên dịch chương trình SAMPLE 02 bằng gcc",
            // Cấu hình để hiển thị lỗi và cảnh báo từ trình biên dịch
            "problemMatcher": "$gcc"
        }
    ],
    "version": "2.0.0"
}
```

Đây là "run command", nhóm build

Tóm tắt

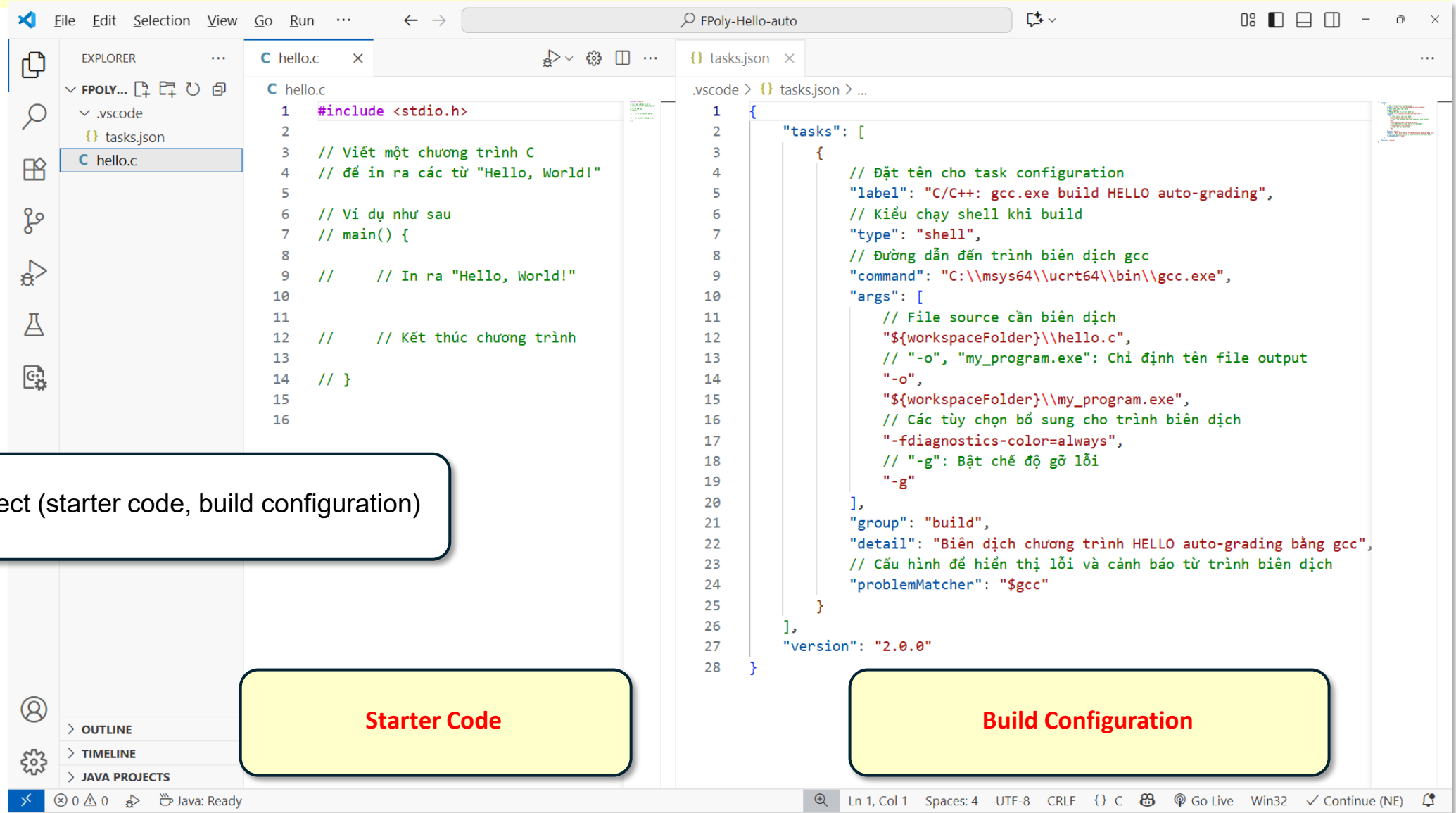
VS Code C programming → GitHub Classroom

- ✓ 1. Cài đặt VS Code
- ✓ 2. Cài đặt MinGW
- ✓ 3. Build configuration – tasks.json
- ✓ 4. Simple build run command

- - Create classroom (Tạo lớp)
- - Create assignment, starter code
- - Add auto-grading

VS Code – Hello-auto-grading

Bước 1 : Tạo Project (starter code, build configuration)



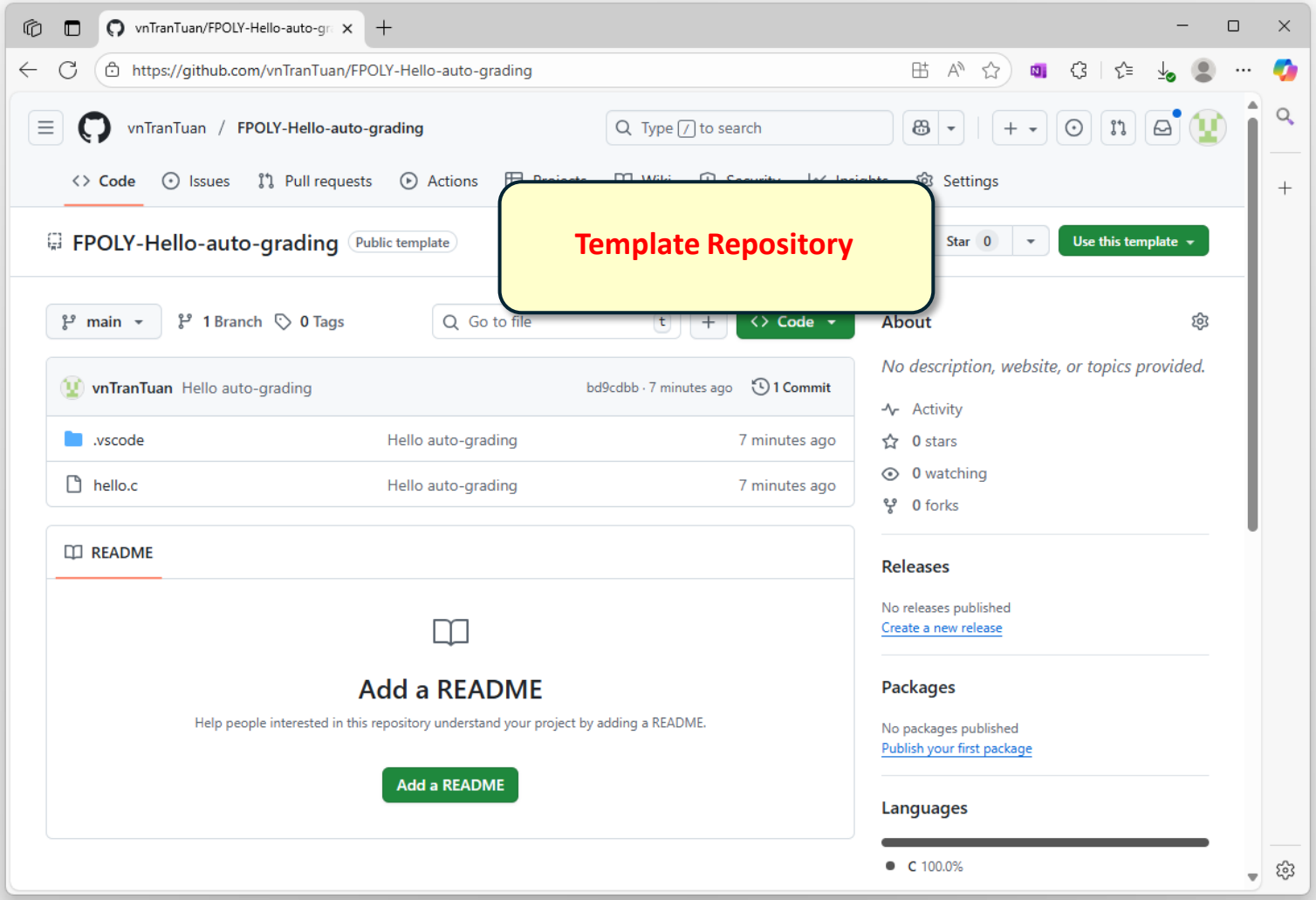
GitHub – Hello-auto-grading repository

Bước 2 – Template repository

a. Tạo repository

b. Commit & Push Project

c. Bật Template Repository

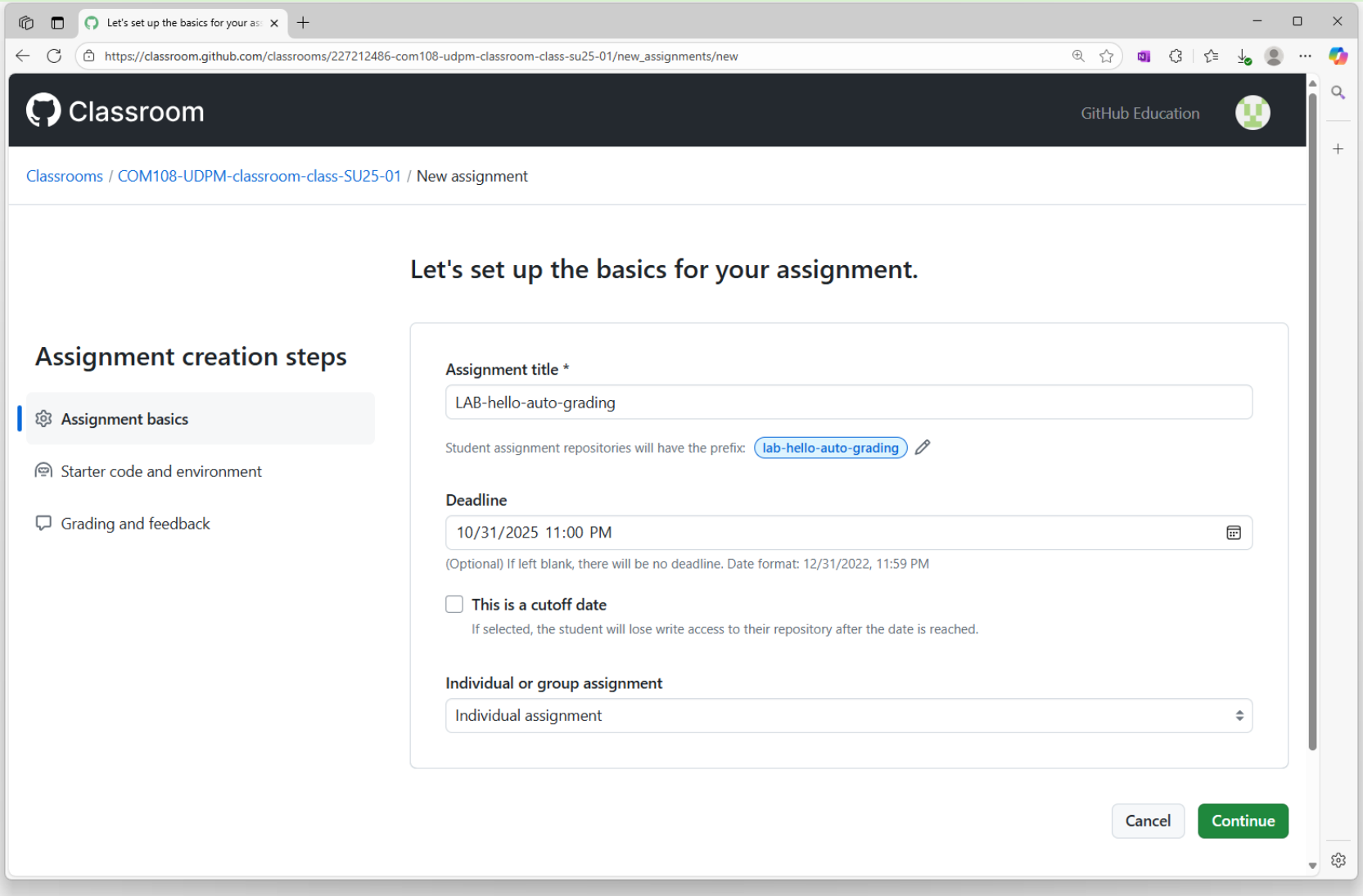


Tạo Assignment Auto-grading

Bước 3 – New assignment

a. New assignment

b. Assignment basics

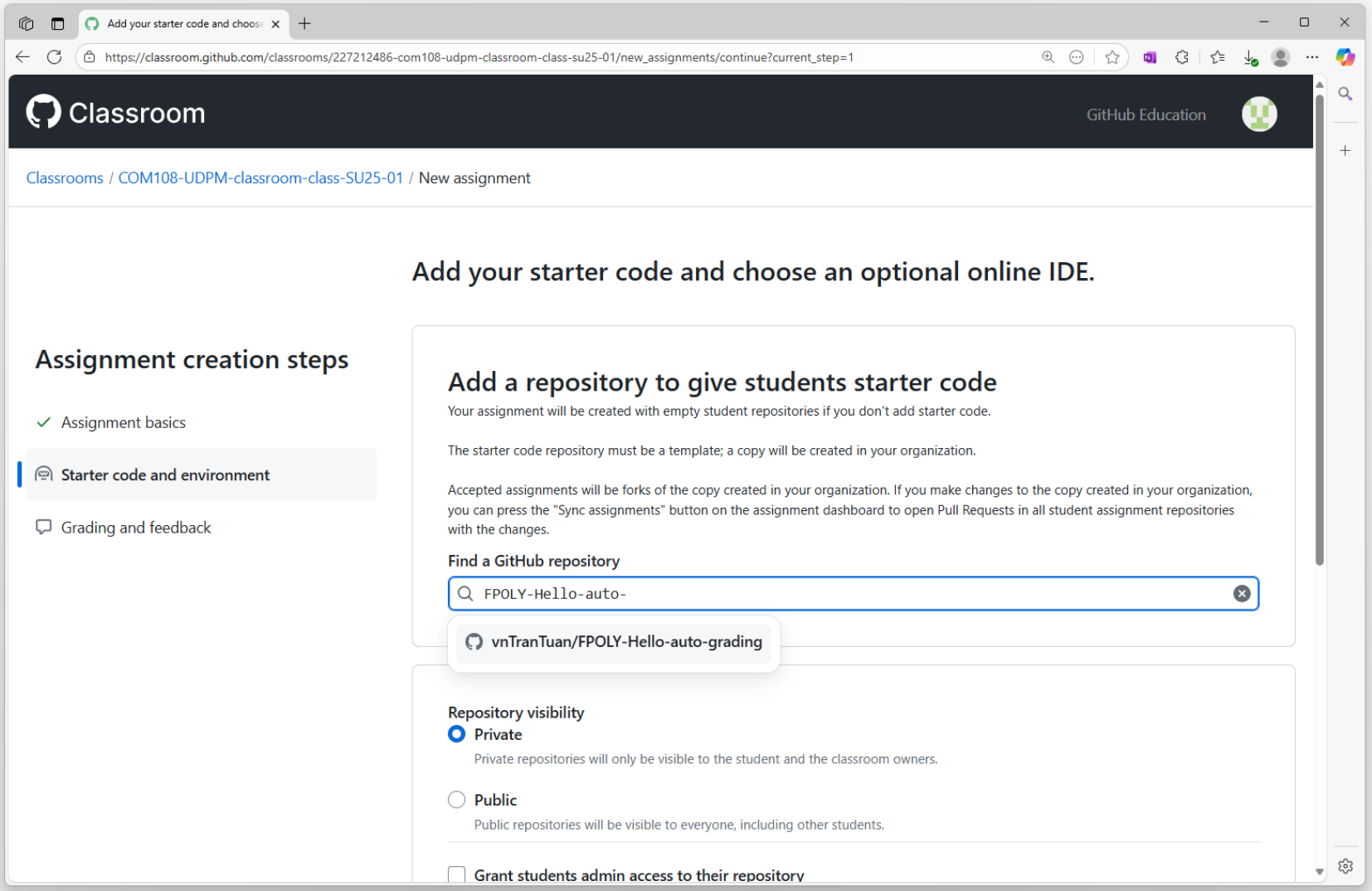


Tạo Assignment Auto-grading

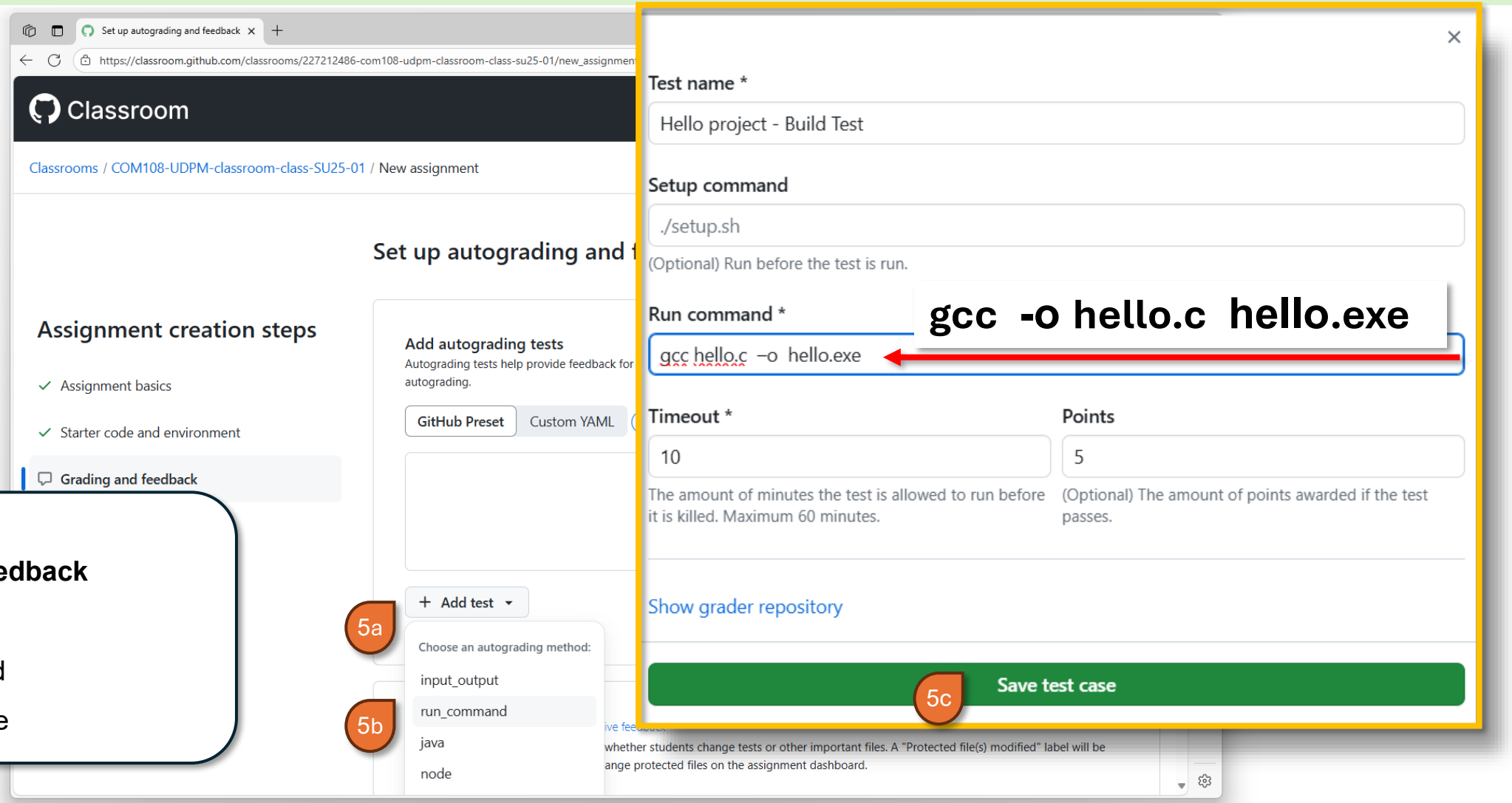
Bước 4 –

Starter code and environment

Select GitHub repository



Tạo Assignment Auto-grading



Bước 5 –

Grading and feedback

a. Add test

b. run_command

c. Save test case

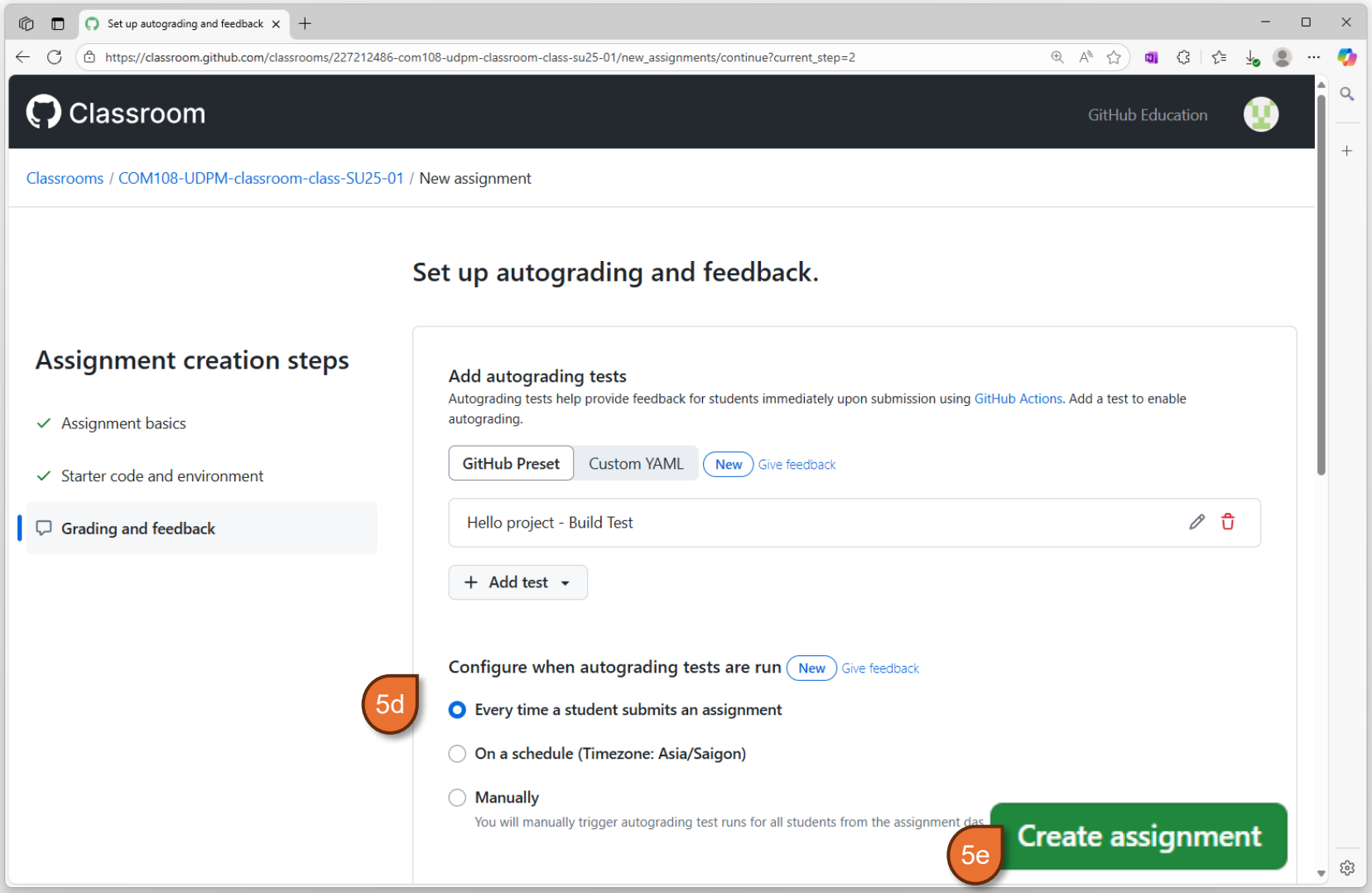
Tạo Assignment Auto-grading

Bước 5 –

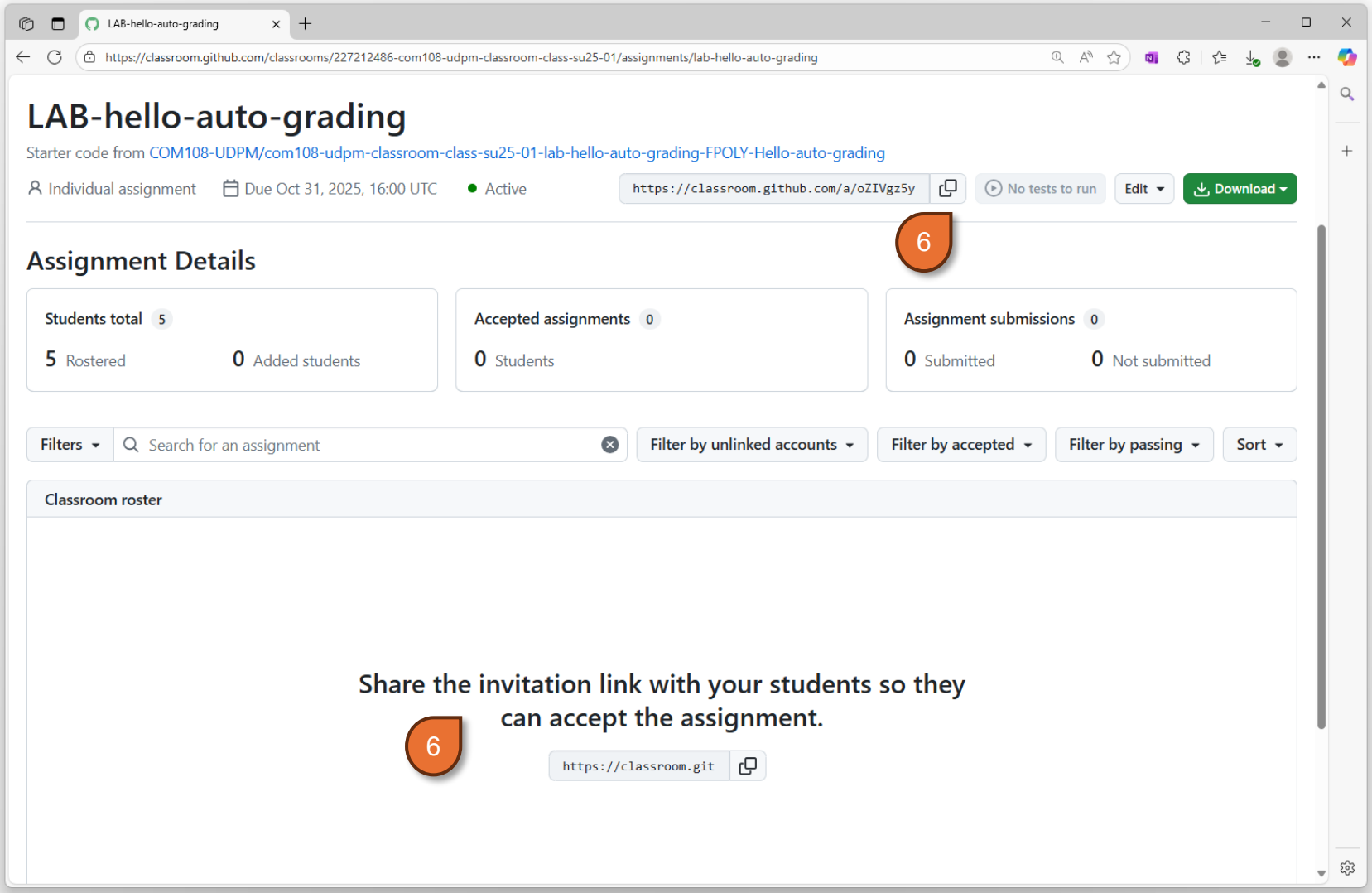
Grading and feedback

d. Configuration auto grading

e. Create assignment



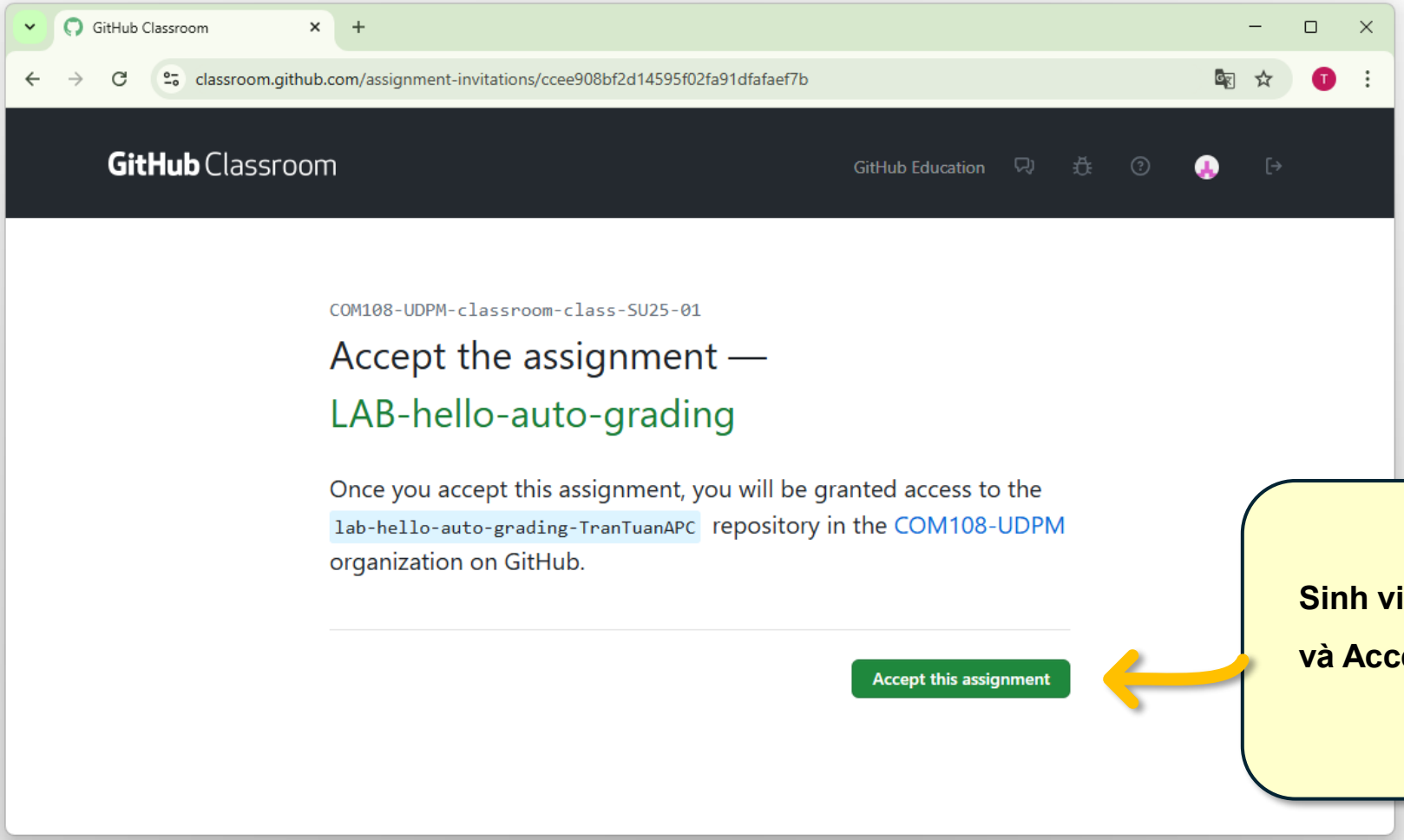
Share Assignment link



Bước 6 –

Share assignment link

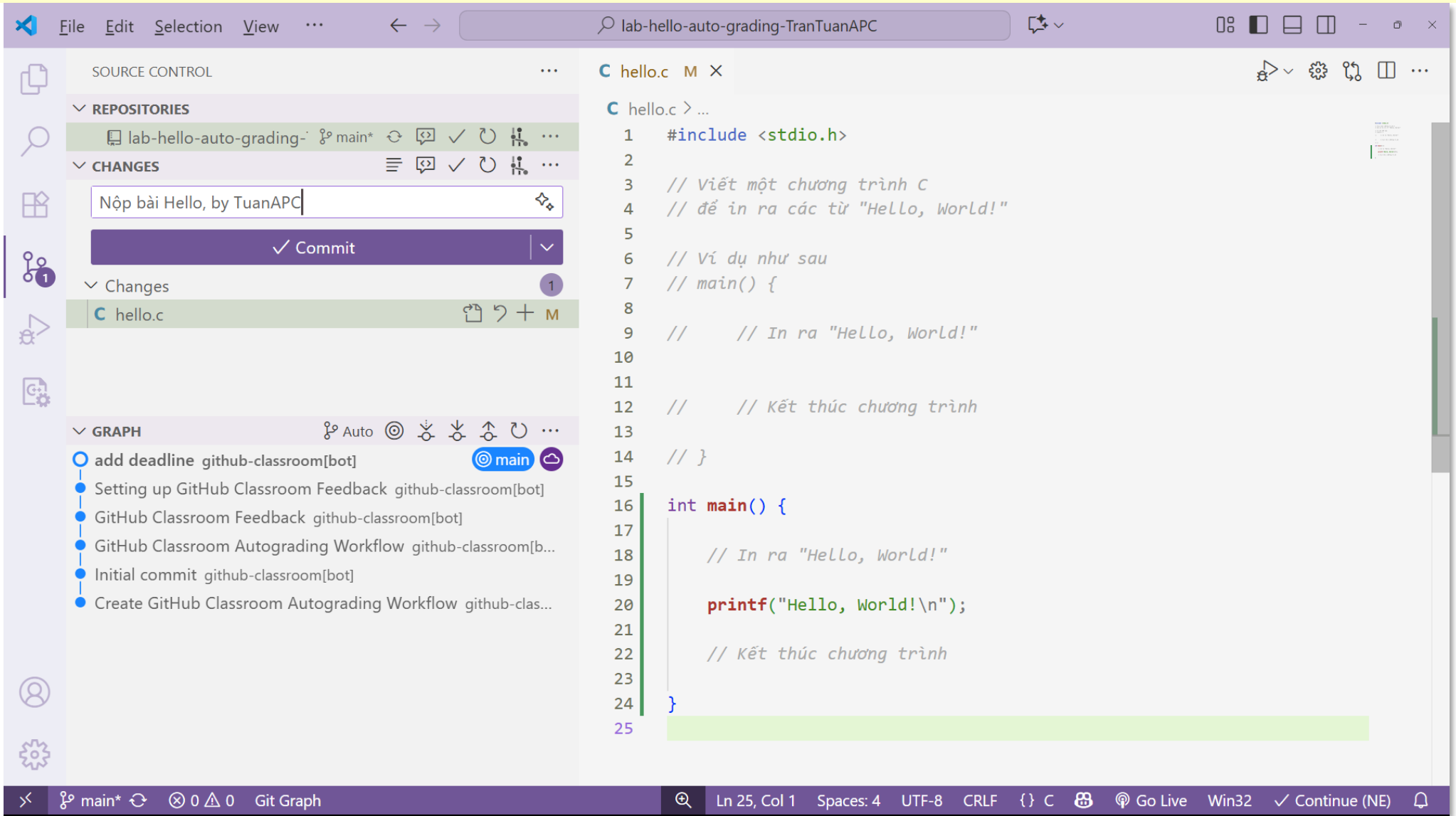
Accept Assignment link



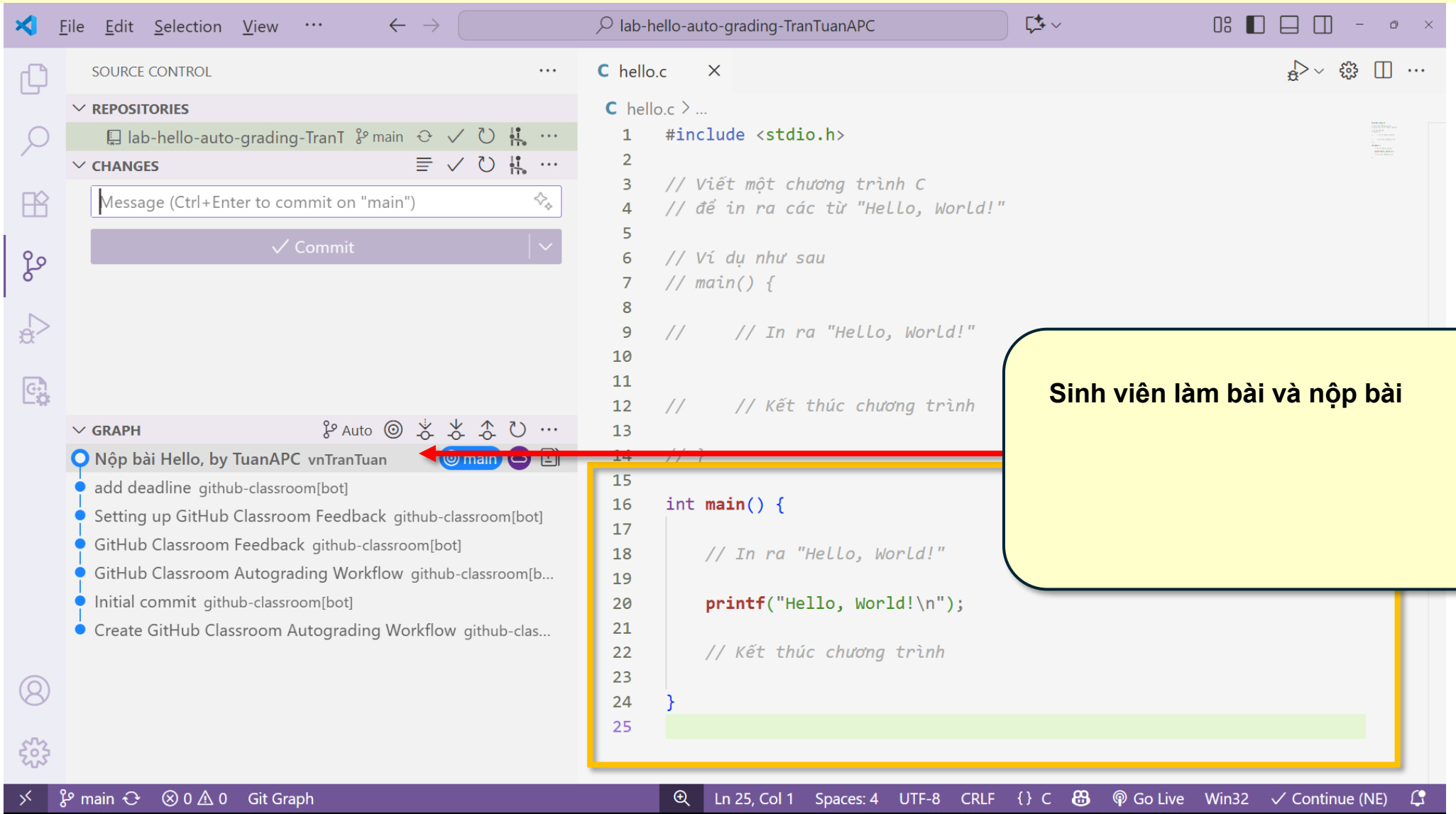
Sinh viên nhận link

và Accept assignment

VS Code – Clone assignment & Do homework



VS Code – Submit assignment



Assignment auto-grading reporter

Giảng viên nhận thông tin

Trạng thái và kết quả (điểm)

của các bài đã nộp

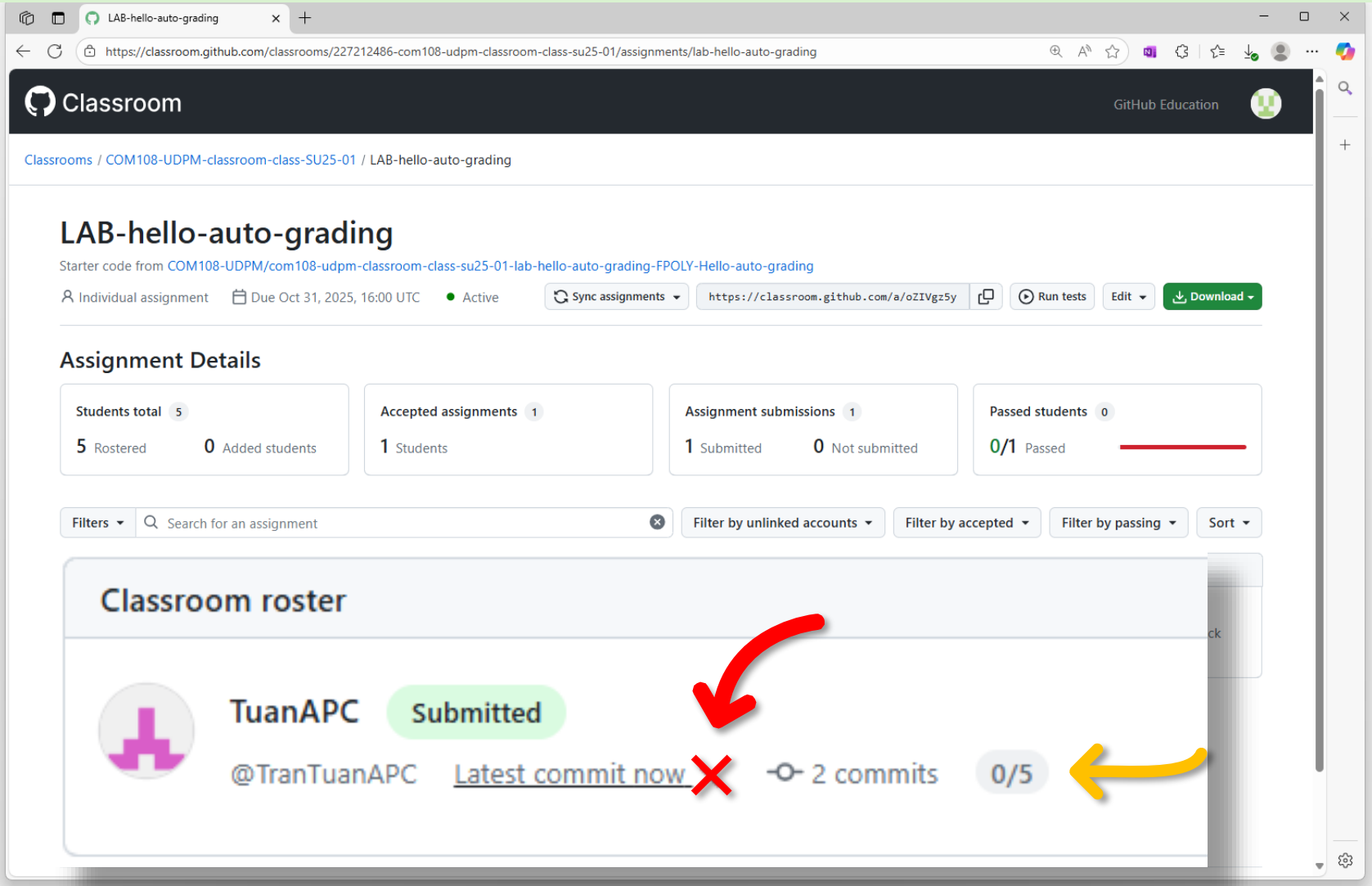
Ví dụ

Bài đã nộp

Submitted

Trạng thái

❌ Failed



Assignment auto-grading reporter

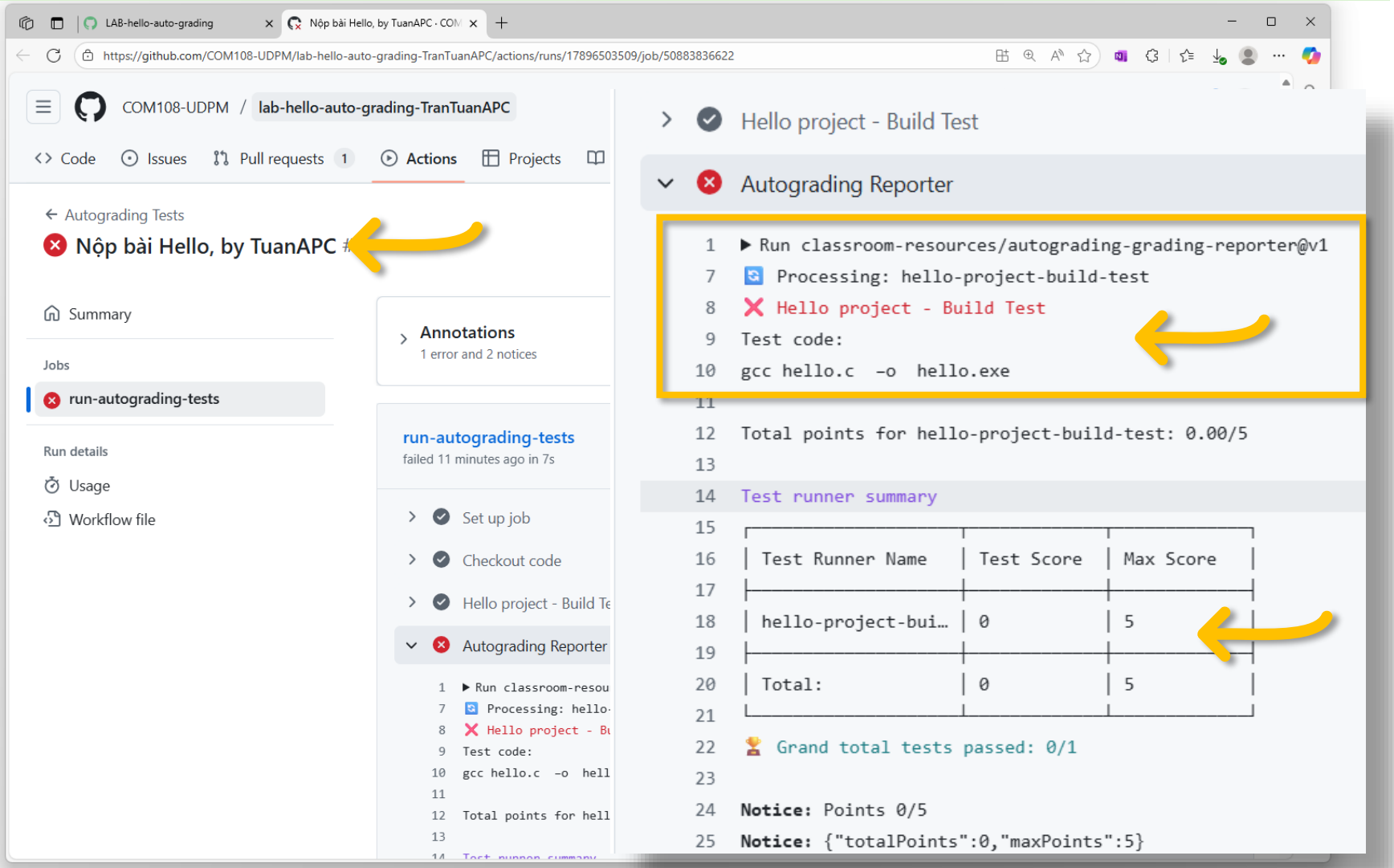
Giảng viên nhận thông tin

Chi tiết kết quả auto-grading

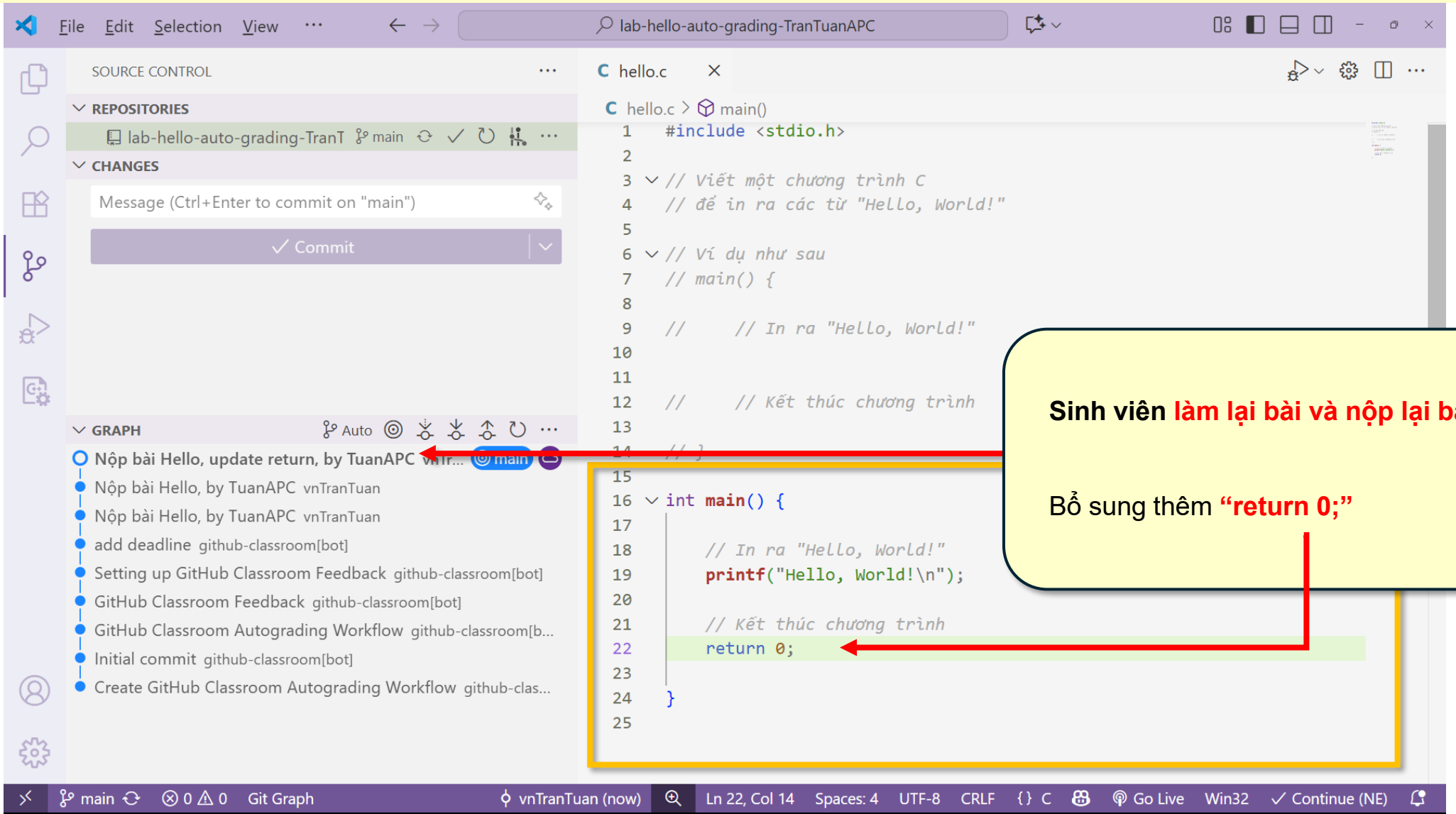
❌ Failed

Do bài làm chưa đúng các ý sau

1. Hàm main() có kiểu trả về là int
2. Thiếu lệnh “return <int>;”



VS Code – Submit assignment



Sinh viên làm lại bài và nộp lại bài

Bổ sung thêm **"return 0;"**

Assignment auto-grading reporter

Giảng viên nhận thông tin

ĐÃ CẬP NHẬT

Trạng thái và kết quả (điểm)

của các bài đã nộp

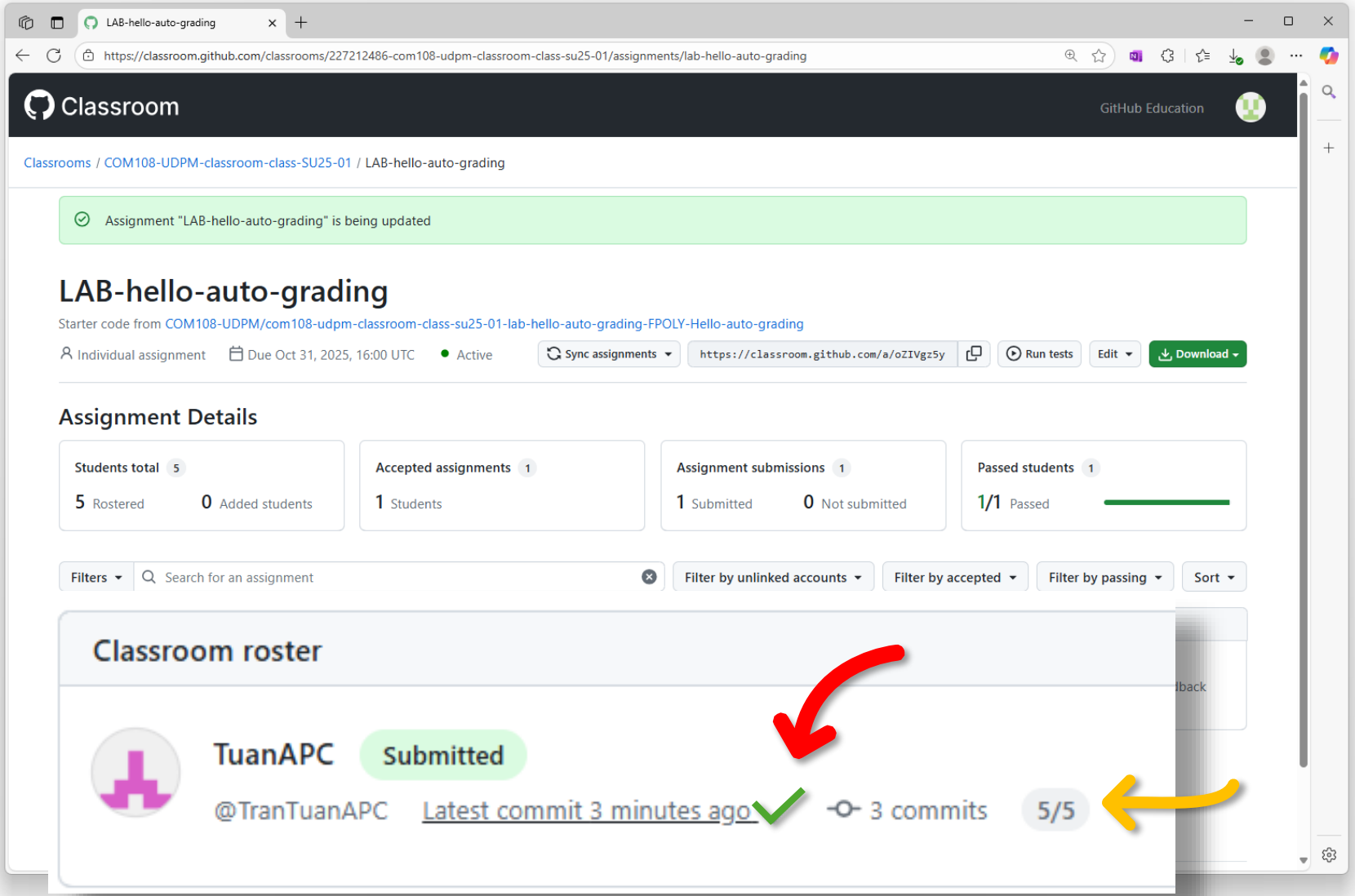
Ví dụ

Bài đã nộp

Submitted

Trạng thái

✓ Passed



Assignment auto-grading reporter

Summary

Jobs

run-autograding-tests

Run details

Usage

Workflow file

Giảng viên nhận thông tin

KẾT QUẢ CHI TIẾT

Add test run_command

run-autograding-tests

succeeded 3 minutes ago in 7s

Search logs

Hello project - Build Test 1s

```
1  ▼ Run classroom-resources/autograding-command-grader@v1
2    with:
3      test-name: Hello project - Build Test
4      command: gcc hello.c -o hello.exe
5      timeout: 10
6      max-score: 5
```

Autograding Reporter 1s

```
1  ▶ Run classroom-resources/autograding-grading-reporter@v1
7  🔄 Processing: hello-project-build-test
8  ✅ Hello project - Build Test
9  Test code:
10 gcc hello.c -o hello.exe
11
12 Total points for hello-project-build-test: 5.00/5
13
14 Test runner summary
15 ┌──────────────────┬────────────┬─────────────┐
16 │ Test Runner Name │ Test Score │ Max Score   │
17 ├──────────────────┼────────────┼─────────────┤
18 │ hello-project-bui… │ 5          │ 5           │
19 ├──────────────────┼────────────┼─────────────┤
20 │ Total:           │ 5          │ 5           │
21 └──────────────────┴────────────┴─────────────┘
22 🎉 Grand total tests passed: 1/1
23
24 Notice: Points 5/5
25 Notice: {"totalPoints":5,"maxPoints":5}
```

Tóm tắt

VS Code C programming

- ✓ 1. Cài đặt VS Code
- ✓ 2. Cài đặt MinGW
- ✓ 3. Build configuration – tasks.json
- ✓ 4. Simple build run command

GitHub Classroom

- Create classroom (Tạo lớp)
- ✓ Create assignment, starter code
- ✓ Add auto-grading

Student work

- ✓ Accept assignment
- ✓ Do homework
  - Clone Starter Code
  - Do homework
  - Commit & Publish

GitHub Classroom

- ✓ Run auto-grading
- ✓ Auto-grading reporter

- ✓ Submit assignment

Tóm tắt

GitHub Classroom

- Create classroom (Tạo lớp)
- ✓ Create assignment, starter code
- ✓ Add auto-grading

✓ ĐÃ THỰC HIỆN

Với trường hợp đơn giản, là

✓ 1. Chương trình đơn giản một source file

✓ 2. Lỗi đơn giản, lỗi biên dịch (build error)

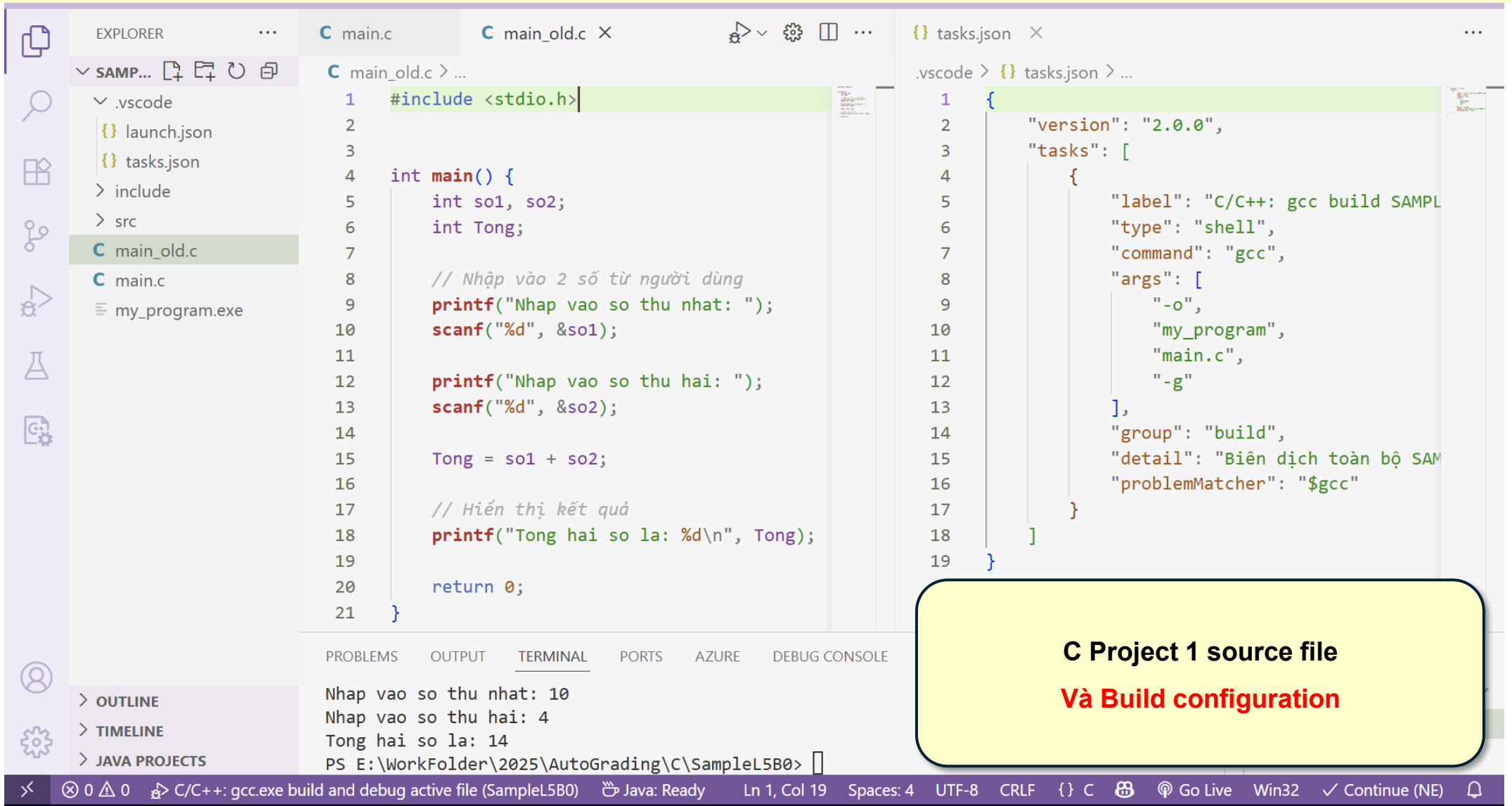
C & CUnit project

BỔ SUNG

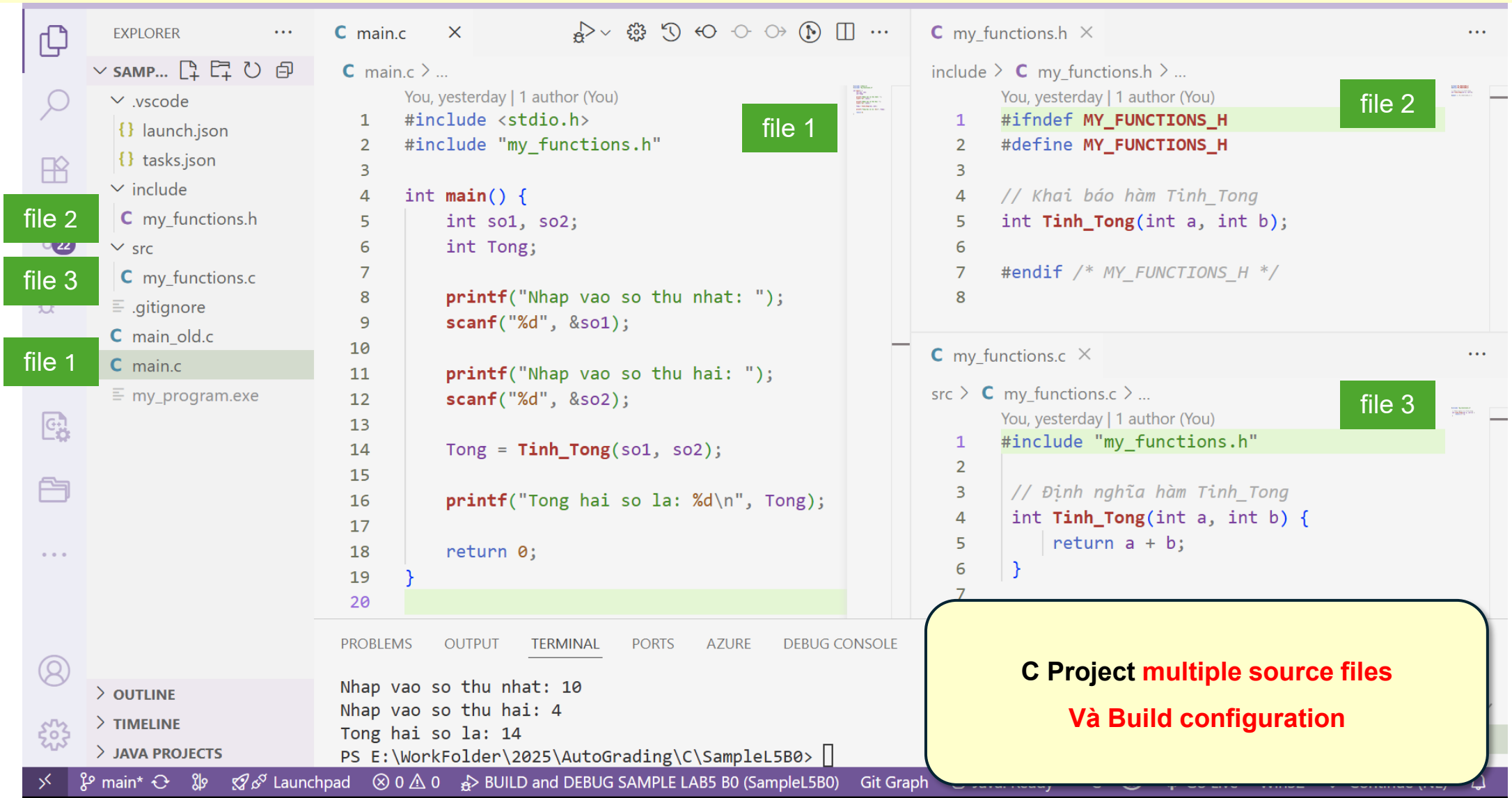
Với trường hợp phức tạp hơn, là

1. Chương trình nhiều source files
2. Tạo Test case để kiểm tra, sử dụng thư viện CUnit

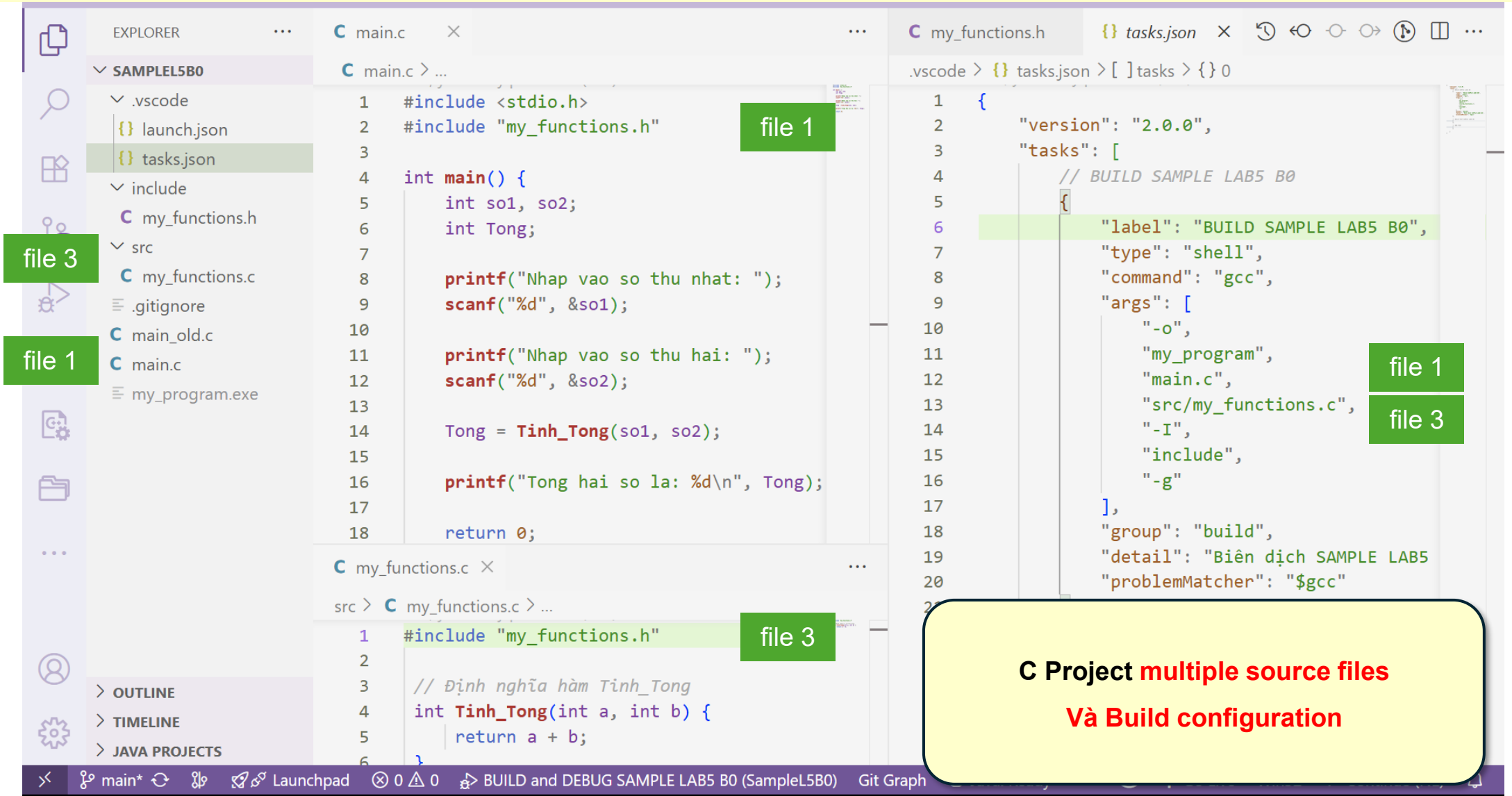
VS Code – C Project multiple source files



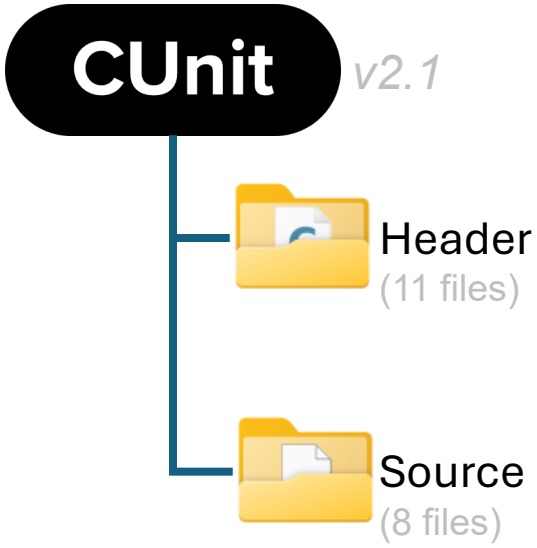
VS Code – C Project multiple source files

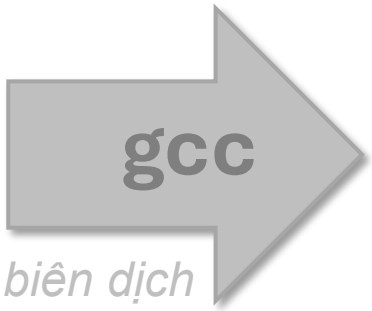


VS Code – C Project multiple source files

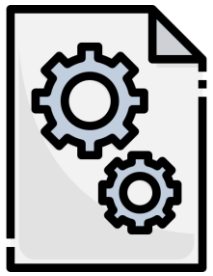


GCC – build Cunit (local)





Static library



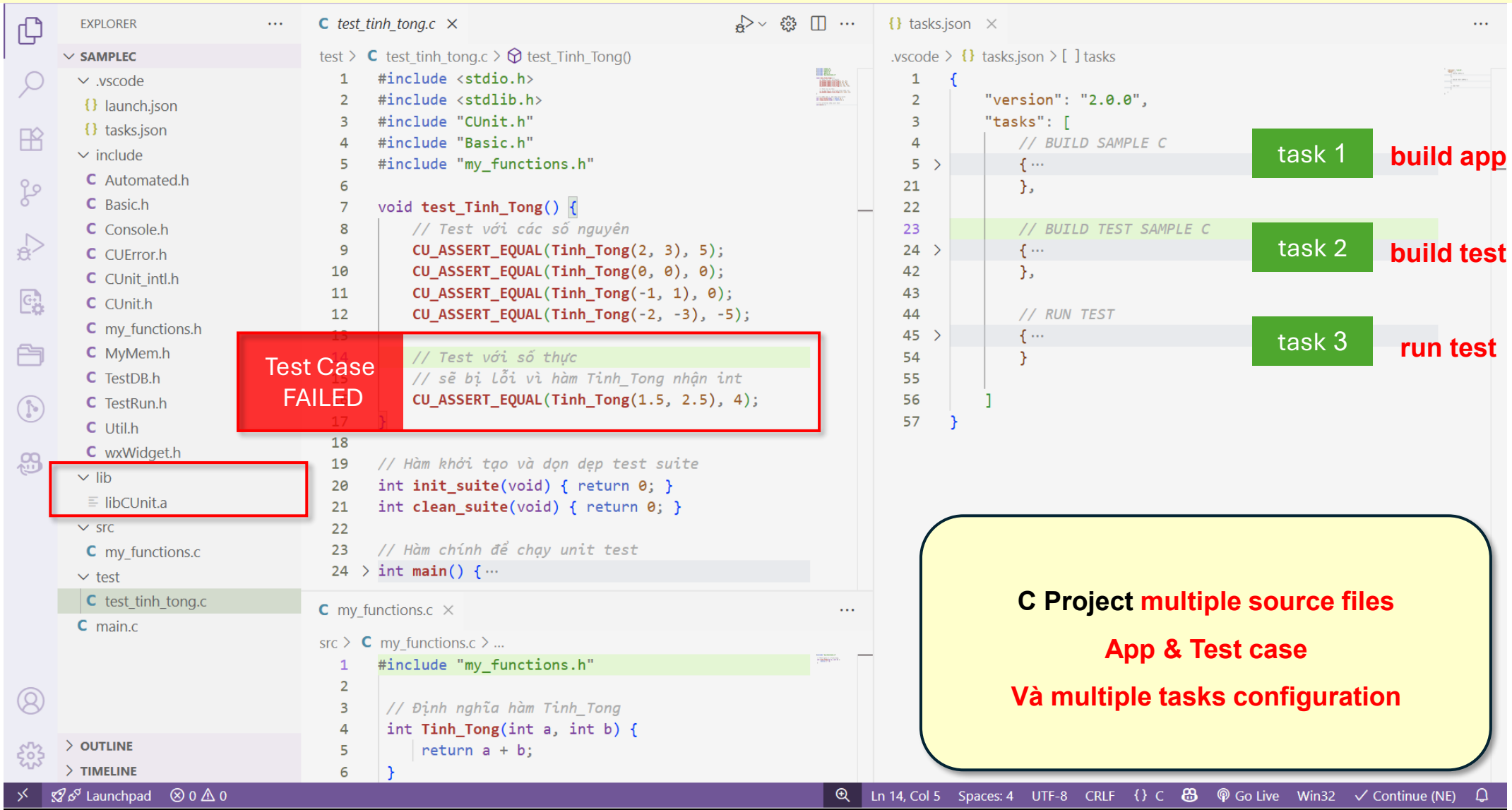
`libCUnit.a`
file

Sử dụng thư viện

`libCUnit.a`

để viết test case

VS Code – C Project & CUnit test case



VS Code – C Project & CUnit test case

task 1 **build app**

```
// BUILD SAMPLE C
{
    "label": "BUILD SAMPLE C",
    "type": "shell",
    "command": "gcc",
    "args": [
        "-o",
        "my_program",
        "main.c",
        "src/my_functions.c",
        "-I",
        "include",
        "-g"
    ],
    "group": "build",
    "detail": "Biên dịch SAMPLE C",
    "problemMatcher": "$gcc"
},
```

task 2 **build test**

```
// BUILD TEST SAMPLE C
{
    "label": "BUILD TEST SAMPLE C",
    "type": "shell",
    "command": "gcc",
    "args": [
        "-o",
        "test_runner",
        "test/test_tinh_tong.c",
        "src/my_functions.c",
        // link CUnit library, tại thư mục lib
        "-L", "lib", "-lCUnit",
        "-I",
        "include",
        "-g"
    ],
    "group": "build",
    "detail": "Biên dịch TEST SAMPLE C",
    "problemMatcher": "$gcc"
},
```

task 3 **run test**

```
// RUN TEST
{
    "label": "run_test",
    "type": "shell",
    "command": "./test_runner",
    "dependsOn": "BUILD TEST SAMPLE C",
    "group": {
        "kind": "test",
        "isDefault": true
    }
}
```

C Project multiple source files

App & Test case

Và multiple tasks configuration

Tạo Auto-grading run_command

task 1 **build app**

```
4   // BUILD SAMPLE C
5   {
6       "label": "BUILD SAMPLE C",
7       "type": "shell",
8       "command": "gcc",
9       "args": [
10          "-o",
11          "my_program",
12          "main.c",
13          "src/my_functions.c",
14          "-I",
15          "include",
16          "-g"
```

gcc -o my_program
main.c
src/my_functions.c
-I include

task 2 **build test**

```
23  // BUILD TEST SAMPLE C
24  {
25      "label": "BUILD TEST SAMPLE C",
26      "type": "shell",
27      "command": "gcc",
28      "args": [
29          "-o",
30          "test_runner",
31          "test/test_tinh_tong.c",
32          "src/my_functions.c",
33          // link CUnit library, tại thư mục lib
34          "-L", "lib", "-lCUnit",
35          "-I",
36
37
38
39
40
41
42
43
```

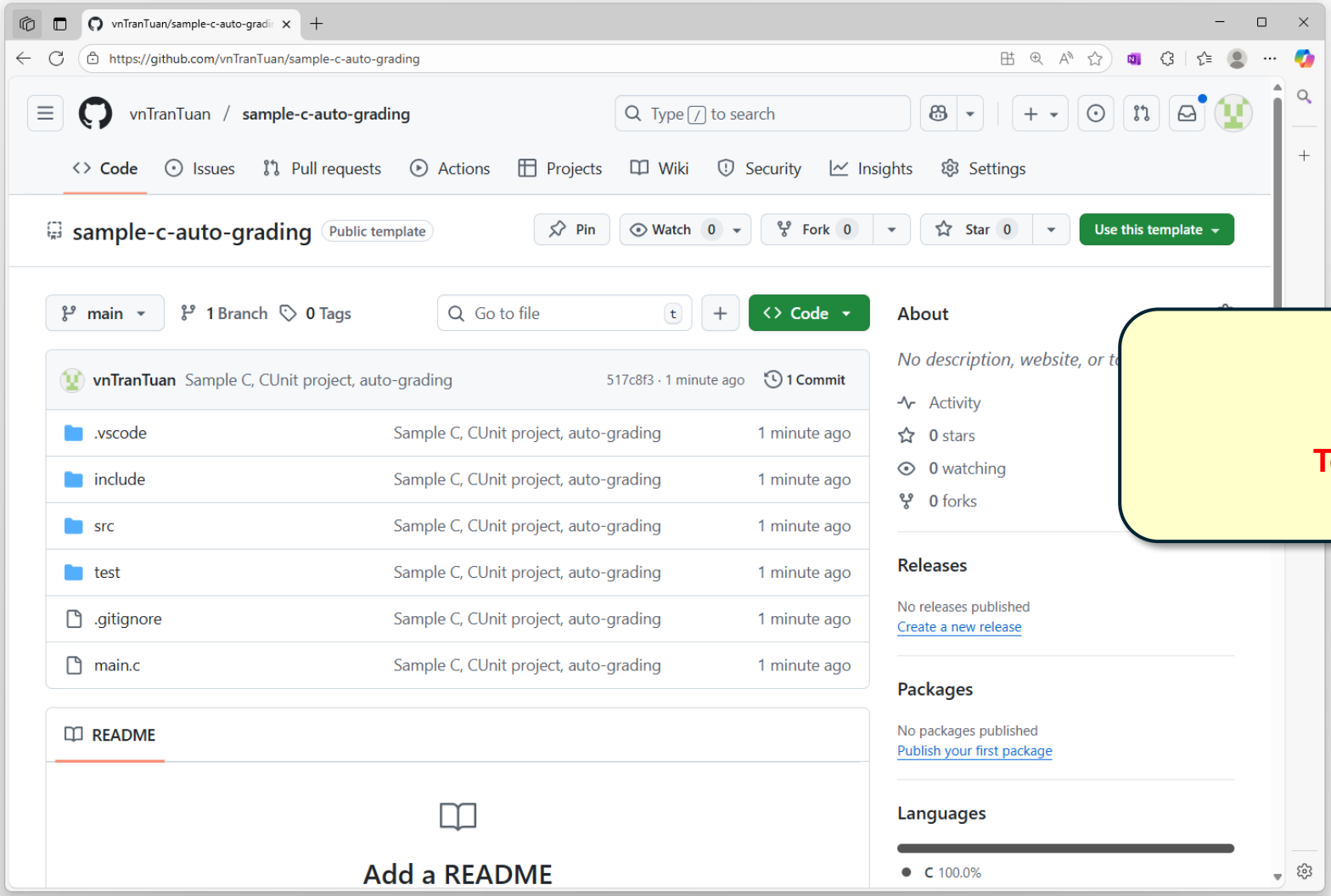
gcc -o test_runner
test/test_tinh_tong.c
src/my_functions.c
-I include
-Llib -lCUnit

task 3 **run test**

```
44  // RUN TEST
45  {
46      "label": "run_test",
47      "type": "shell",
48      "command": "./test_runner",
49      "dependsOn": "BUILD TEST SAMPLE C",
50      "group": {
51          "kind": "test",
52          "isDefault": true
53      }
54  }
55
```

./test_runner

GitHub – C & CUnit Project, repository



C & CUnit Project

Template repository

Tóm tắt

GitHub Classroom

- Create classroom (Tạo lớp)
- ✓ Create assignment, starter code
- ✓ Add auto-grading

C & CUnit project

Auto-grading

✓ ĐÃ THỰC HIỆN

Với trường hợp đơn giản, là

✓ 1. Chương trình đơn giản **một source file**

✓ 2. Lỗi đơn giản, lỗi biên dịch (**build error**)

BỔ SUNG

Với trường hợp phức tạp hơn, là

✓ 1. Chương trình **nhiều source files**

✓ 2. Tạo **Test case** để kiểm tra, sử dụng thư viện **CUnit**

Tạo Assignment Auto-grading

Tiếp tục add auto-grading ở trang sau

Gồm các test như sau

1. **Test build app** (5 điểm):

   để kiểm tra “build error”

2. **Build test** (5 điểm):

   để biên dịch chương trình test

3. **Run test** (5 điểm):

   để chạy chương trình test

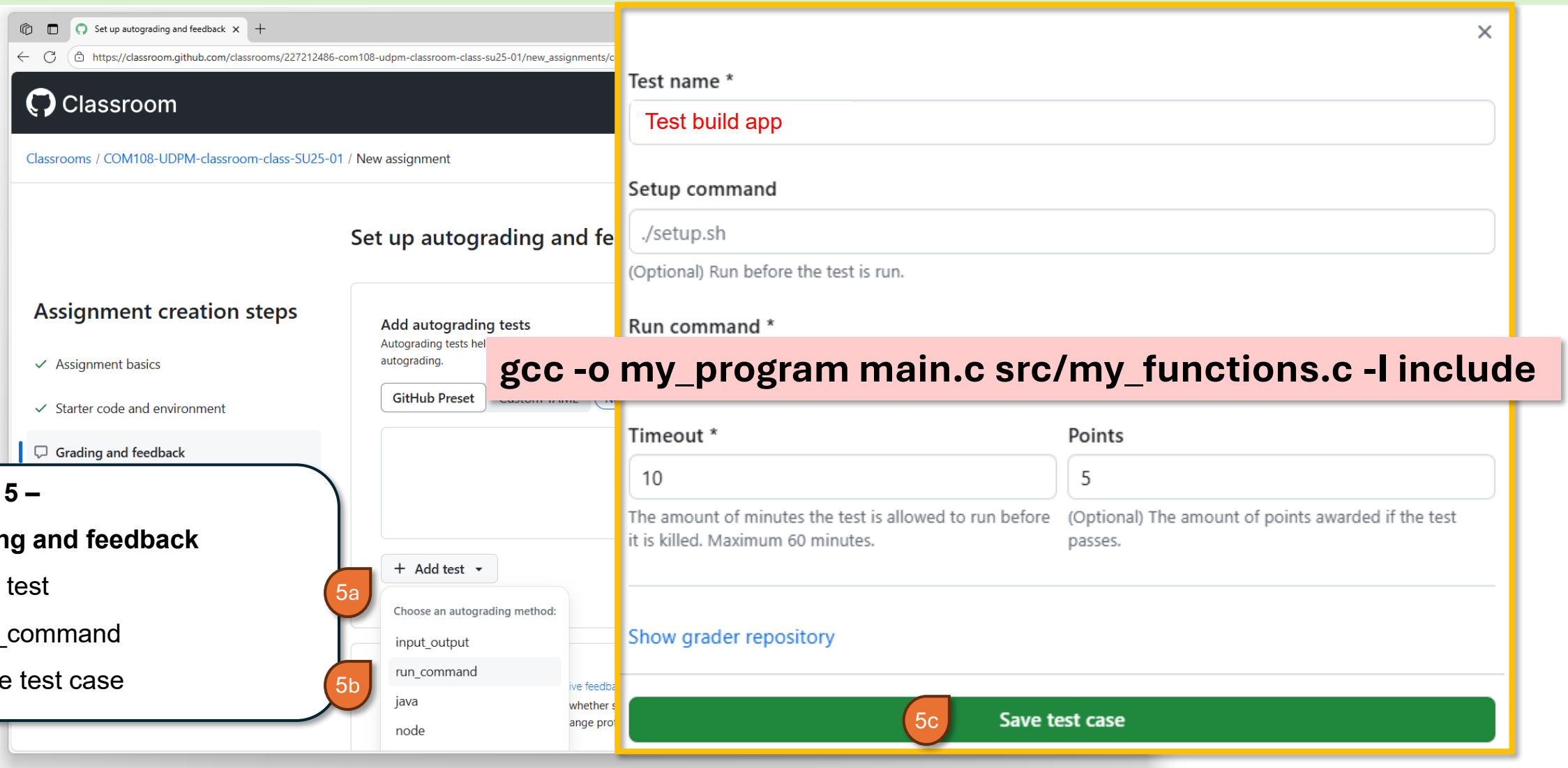
Tham khảo các bước Tạo assignment

(slide phần 1)

a. Assignment basics

b. Starter code and environment

Tạo Assignment Auto-grading, Test build app



Bước 5 –

Grading and feedback

a. Add test

b. run_command

c. Save test case

Tạo Assignment Auto-grading, Build test

Bước 5 –

Grading and feedback

a. Add test

b. run_command

c. Save test case

Tạo Assignment Auto-grading, Build test

Run_command Build test có thể sử dụng 2 cách sau :

1 CUnit đã được biên dịch

- Lệnh ngắn
- CUnit đã được biên dịch tại máy local
- Trình biên dịch ở local và GitHub Server có thể khác nhau, và dẫn đến “error” khi chạy “auto-grading”

2 CUnit SẼ được biên dịch ở GitHub Server

- Lệnh đầy đủ
- CUnit source files cần được upload lên GitHub Server
- CUnit và chương trình cùng được biên dịch ở GitHub Server, và phù hợp để chạy “auto-grading”

1

```
gcc -o test_runner test/test_tinh_tong.c src/my_functions.c -I include -Llib -lCUnit
```

2

```
gcc -c cunit-src/*.c -I include && ar rcs lib/libCUnit.a *.o && gcc -o test_runner test/test_tinh_tong.c src/my_functions.c -I include -Llib -lCUnit
```

Tạo Assignment Auto-grading, Build test

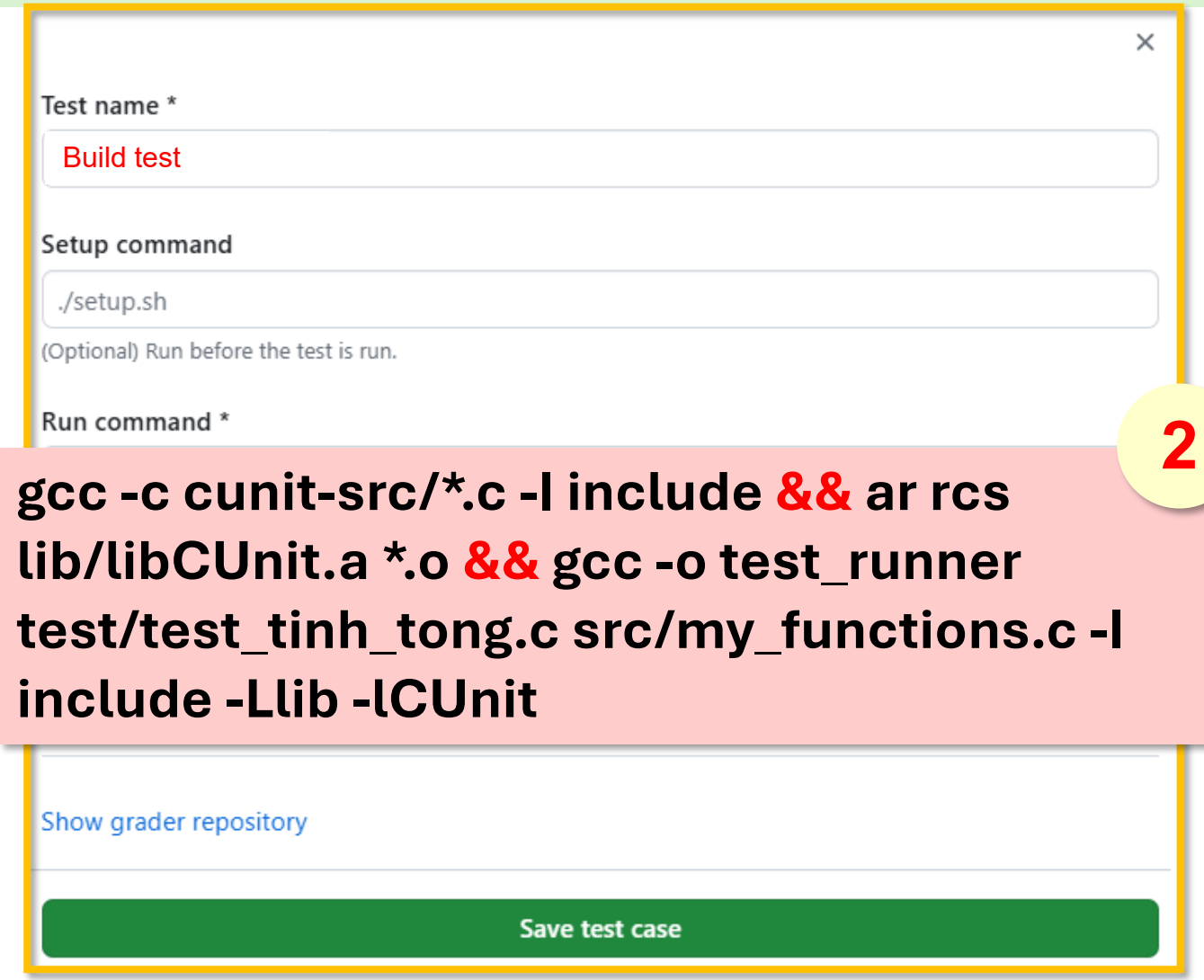
Run_command có trật tự và nội dung như sau :

- "`gcc`" : lệnh biên dịch
- `-o test_runner` : kết quả là file "test_runner.exe"
- `test/test_tinh_tong.c` : source của chương trình kiểm thử
- `src/my_functions.c` : có sử dụng thư viện tự tạo
- `-I include` : thư mục chứa các header
- `-Llib` : chỉ định thư mục "lib" chứa file thư viện CUnit đã được biên dịch
- `-lCUnit` : liên kết thư viện CUnit vào chương trình

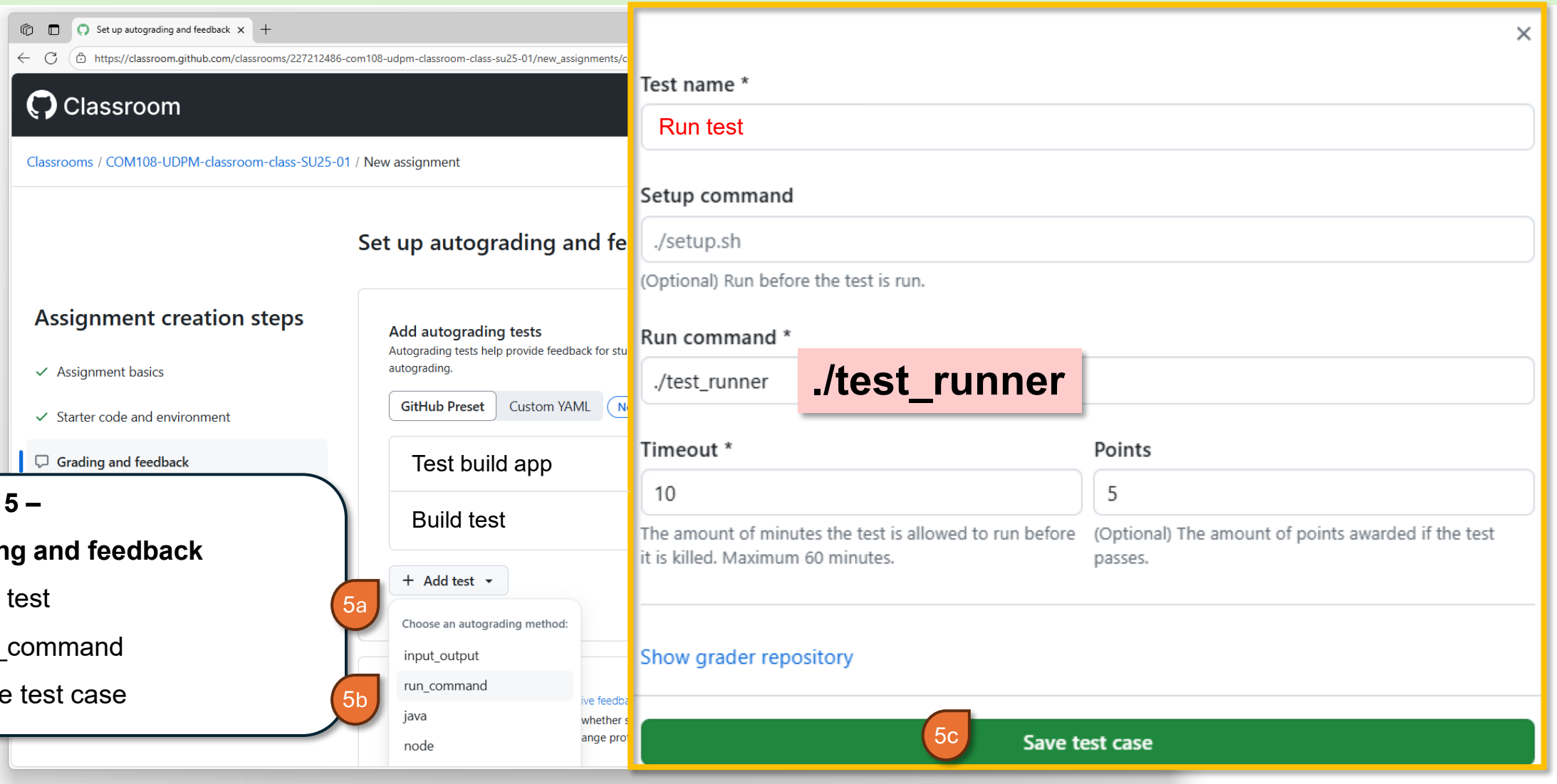
Tạo Assignment Auto-grading, Build test

Run_command có trật tự và nội dung như sau :

- "`gcc -c cunit-src/*.c -I include`" : lệnh biên dịch CUnit source files thành objects
- `ar rcs lib/libCUnit.a *.o` : dịch các objects và tạo thành thư viện tĩnh "`libCUnit.a`"
- `&&` : nối các thành phần lệnh và đảm bảo thực hiện thành công thành phần lệnh trước rồi đến thành phần lệnh sau của toàn bộ câu lệnh
- Thành phần lệnh còn lại tương tự như cách 1



Tạo Assignment Auto-grading, Run test



Bước 5 –

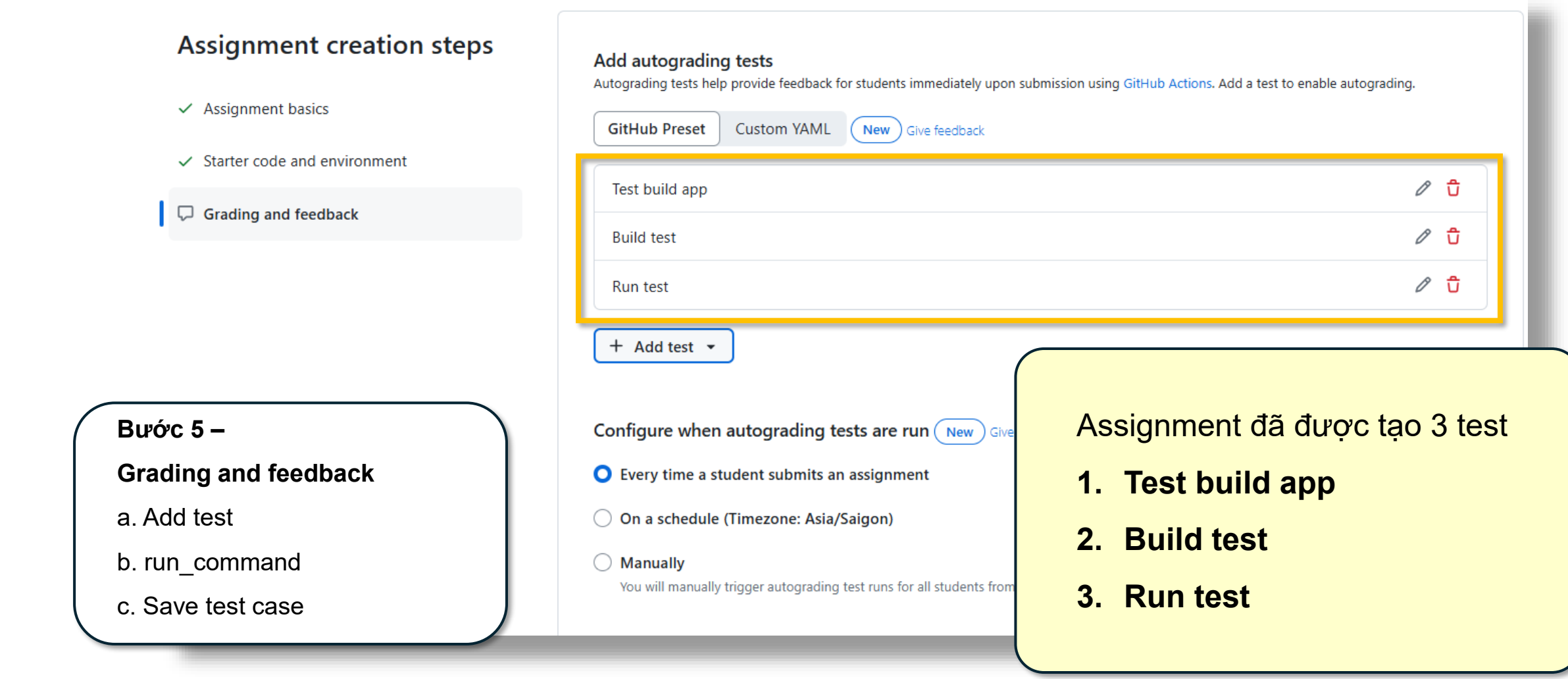
Grading and feedback

a. Add test

b. run_command

c. Save test case

Tạo Assignment Auto-grading, các tests



Bước 5 –

Grading and feedback

a. Add test

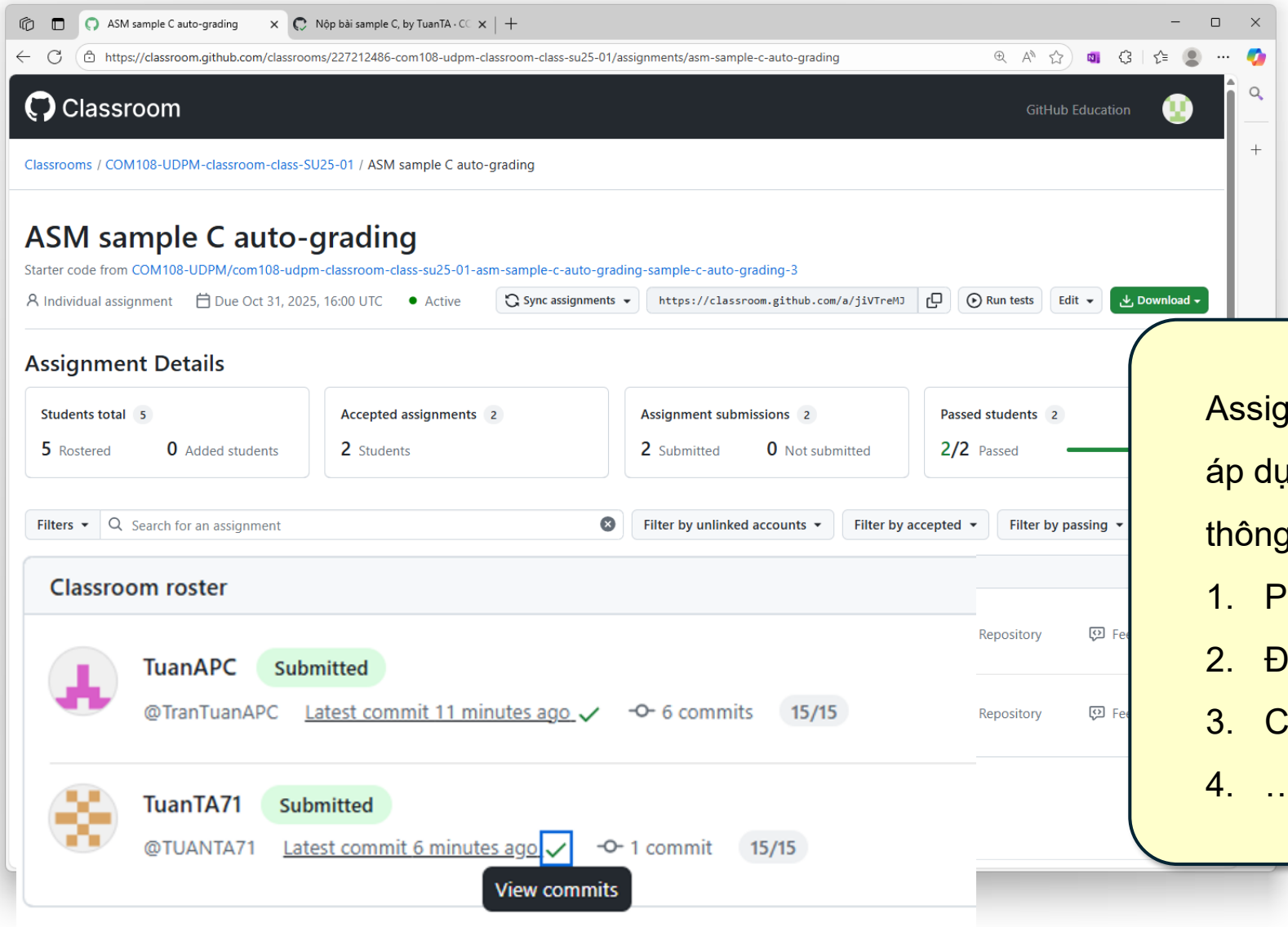
b. run_command

c. Save test case

Assignment đã được tạo 3 test

1. **Test build app**
2. **Build test**
3. **Run test**

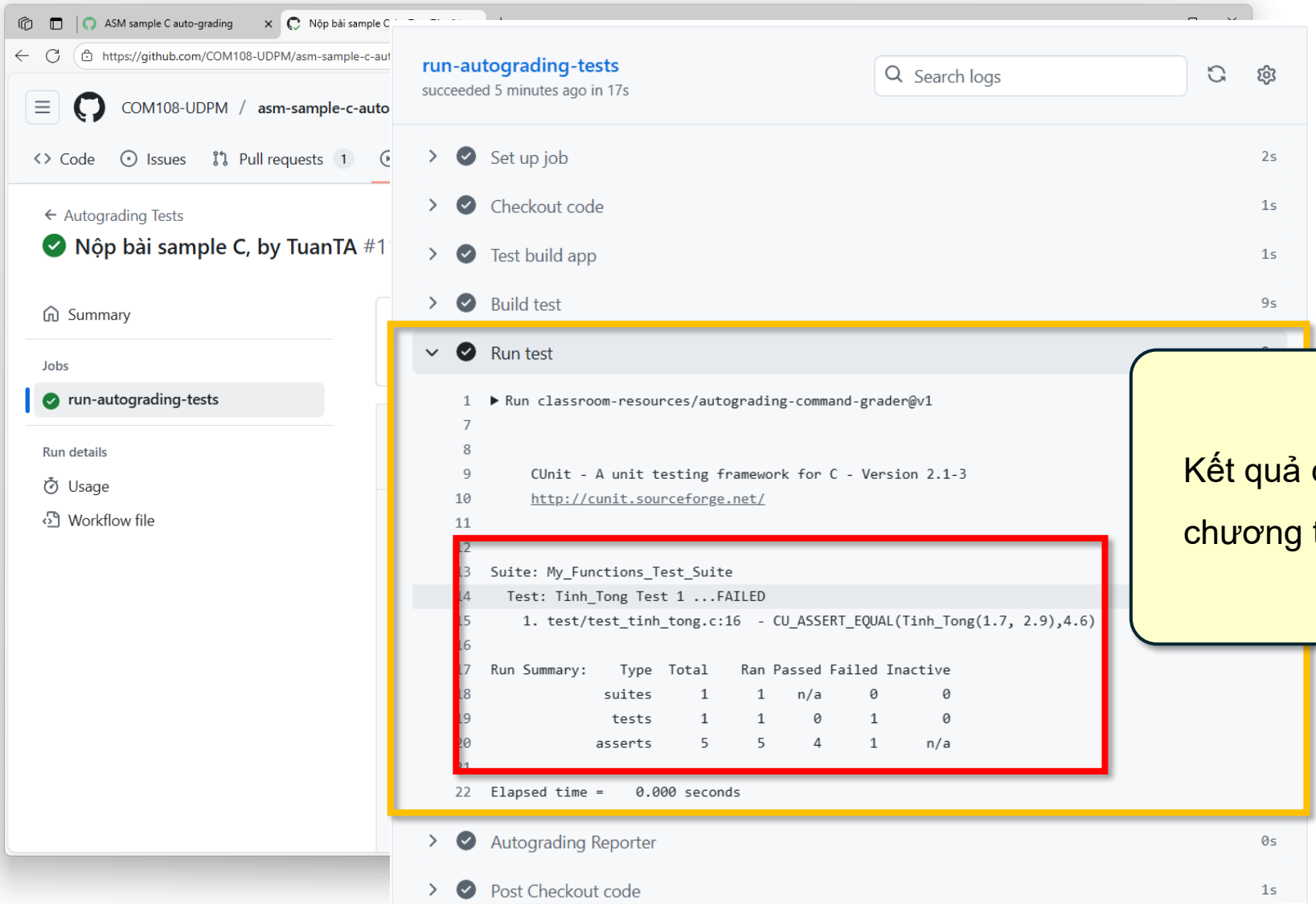
Assignment Auto-grading



Assignment auto-grading đã được áp dụng, với mỗi bài nộp. Các thông tin như

1. Passed hay Failed
2. Điểm số
3. Chi tiết repository đã nộp
4. …

Auto-grading reporter, Detail report



Kết quả các test case đã viết trong chương trình kiểm thử

Tóm tắt

GitHub Classroom

- Create classroom (Tạo lớp)
- ✓ Create assignment, starter code
- ✓ Add auto-grading

✓ ĐÃ THỰC HIỆN

Với trường hợp đơn giản, là

✓ 1. Chương trình đơn giản **một source file**

✓ 2. Lỗi đơn giản, lỗi biên dịch (**build error**)

C & CUnit project

BỔ SUNG

Với trường hợp phức tạp hơn, là

✓ 1. Chương trình **nhiều source files**

✓ 2. Tạo **Test case** để kiểm tra, sử dụng thư viện **CUnit**

Auto-grading

ĐÃ THỰC HIỆN

1. Add test, run command
2. Auto-grading reporter

VS Code – launch.json (debugger configuration)

Đang thực hiện …

HẾT PHẦN 2 – Nhóm 1

GitHub – Classroom
Auto-grading
In C, CUnit programming

TRÂN TRỌNG CẢM ƠN

09-2025

Nhóm thực hiện
Thân Hoàng Lộc
Trần Quang Bình
Trần Văn Huy
Trần Anh Tuấn

Tham khảo

1. https://cunit.sourceforge.net/
2. https://cmake.org/