

Copilot

Certainly! Let's create a Google Test (gtest) unit test for your newly added dummy function. I'll guide you through the process step by step:

1. Create a New Test File:

- First, create a new C++ source file (e.g., `dummy_test.cpp`) in your project directory.
- This file will contain your test cases for the dummy function.

2. Include Necessary Headers:

- In `dummy_test.cpp`, include the necessary headers:

```
#include <gtest/gtest.h>
#include "dummy.h" // Include the header file where your dummy function is declared
```

3. Write Test Cases:

- Define test cases using the `TEST` macro. For example:

```
TEST(DummyFunctionTest, ReturnsExpectedValue) {
    // Arrange (setup)
    // (You can set up any necessary data or context here)

    // Act (call your dummy function)
    int result = dummyFunction();

    // Assert (check the result)
    EXPECT_EQ(result, 42); // Replace with the expected value
}
```

4. Compile and Link:

- Make sure your project is set up to compile both your main code and the test code.
- Link against the Google Test library (usually `libgtest.a` or `gtest.lib`).

5. Run the Tests:

- Build your project.
- Run the compiled test executable (usually named something like `dummy_test`).

6. Observe the Results:

- If all tests pass, you'll see a success message.
- If any test fails, it will provide details about which test case failed and why.

Remember to adjust the filenames, function names, and any other details according to your project. Happy testing! If you need further assistance, feel free to ask. 😊🚀

For more examples and advanced features, check out the [GoogleTest Primer](#) and the [GoogleTest Samples](#).