

## Copilot

Certainly! A test fixture in Google Test allows you to reuse the same configuration of objects for multiple tests. Here's how you can set up a test fixture:

### 1. Derive a Class from `testing::Test`:

- Create a new C++ source file (e.g., `dummy_test.cpp`).
- Inside this file, derive a class from `testing::Test`:

```
#include <gtest/gtest.h>

class MyTestFixture : public ::testing::Test {
    // Declare any objects you plan to use
    // (e.g., your dummy function objects or other resources)
};
```

### 2. Prepare Objects for Each Test:

- If necessary, write a default constructor or a `SetUp()` function to prepare the objects for each test:

```
class MyTestFixture : public ::testing::Test {
protected:
    void SetUp() override {
        // Initialize your objects here
        // (e.g., create instances of your dummy function)
    }
};
```

### 3. Write Your Test Cases:

- Define your test cases inside the test fixture class:

```
TEST_F(MyTestFixture, MyFirstTest) {
    // Your test logic here
    // (use the objects you prepared in SetUp())
}

TEST_F(MyTestFixture, MySecondTest) {
    // Another test case
}
```

### 4. Compile and Run:

- Build your project, including both your main code and the test code.
- Run the compiled test executable (usually named something like `dummy_test`).

Remember to adjust filenames, function names, and other details according to your project. If you have any further questions, feel free to ask! 😊🚀

For more details, you can refer to the [GoogleTest Primer](#).