UNL-UAV C++ Assignement Spring 2020

Explaination

Before we get to our activity today we are go though unit testing.

Unit testing is a level of software testing where individual units/ components of a software are tested.

What we can do with unit testing we can make sure a features is working and outputting the correct information, instead of going into gdb or make printf statements everywhere.

How do we use Unit Testing

We are going to be using a libarary called catch2. Which provides an all in one header implmentation of unit testing.

Activity

Before we begin we need to set up our workspace structure. Make sure you have premake5 installed.

Find decent directory to start your workspace in. I recommend starting a folder in ~/git/ for all your git projects. Then when making a workspace just start a new folder in ~/git/. Make a new folder call cpp−tutorial. Then make your file structure like the following:

```
cpp-tutorial/
├── include/
├── src/
├── test/
└── premake5.lua
```

Execute the following command:

```
$ git submodule add https://github.com/catchorg/Catch2 vendor/catch2
```

What this do is pull in a clone of Catch2 as a dependancy. Think of this as a Git repo that links back to the orignal in our Git repo.

In the premake5.lua file copy and paste the code from the following website: https://git.root3287.site/snippets/2

We want to edit a few components.

1. Add the following under line 3

   ```
   includeDir["catch2"] = "vendor/catch2/single_include"
   ```

2. Copy Line 38-56 and paste it under line 56.
3. Change the string on the newly created project to "Test"
4. On the newly created project change the location varible to "workspace/test"
5. On the newly created project change all the src/ to test/

Save the file the execute the following command in the root of the workspace.

If you have Gmake installed:

```
$ premake5 gmake2
```

If you have visual studio installed:

```
$ premake5 vs2019
```

If you have xcode installed:

```
$ premake5 xcode4
```

Now you should see a workspace folder, here you is where you can compile your code using the following command. Your executable will be in bin/. Gmake:

```
$ make
```

Other: Open up the file with your file explorer.

Notice: For all objects make a .h file in the include directory and make a corresponding .cpp

1. Make UAV::Test::Printable with a pure virtual method called printable;
2. Make UAV::Test::Address that implments UAV::Test::Printable with standard postal fields with getter and setters.
3. Make a UAV::Test::Banking::Bank object that have the following fields and the propper getters.
   - std::string name;
   - Address address;
   - std::vector[SafetyDepositBox] boxes;
   - destructor
4. Make a UAV::Test::Banking::SafteyDepositBox object with the following methods and fields:
   - int boxNumber;
   - float money;
   - UAV::Test::Person person;
5. Make a UAV::Test::Person object that implements UAV::Test::Printable and have the following fields:
   - std::string name;
   - int age;
6. Make a test.cpp in the test/ folder. define CATCH CONFIG MAIN, and include catch2 main header. **Note**: You can just use standard assert
7. Create a test case for address using Catch2.
8. Create a test case for person using Catch2.
9. Create a test case for Safty deposit box using Catch2;