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
└── premake5.lua
```

In the premake5.lua file copy and paste the following:

```
outputdir = "%{cfg.buildcfg}/%{cfg.system}-%{cfg.architecture}"
workspace "CPP-Tutorial"
        startproject "Source"
        location "workspace"
        architecture "x64"

        configurations{
                "Debug",
                "Release",
                "Dist"
        }

        filter "configurations:Dist"
                postbuildcommands{
                        "{COPY}␣bin/"..outputdir.."/*␣builds"
                }

        filter "system:windows"
                defines "_WINDOWS"

        filter "system:linux"
                defines "_LINUX"

        filter "system:macosx"
                defines "_OSX"

        filter "configurations:Debug"
                defines "_DEBUG"
                symbols "On"
        filter "configurations:Dist"
                defines "_DIST"
                optimize "On"
        filter "configurations:Release"
                defines "_RELEASE"
                optimize "On"
project "Source"
        cppdialect "C++17"
        location "workspace/source"
        kind "ConsoleApp"
        language "C++"

        targetdir ("bin/" .. outputdir)
        objdir ("bin-int/" .. outputdir)

        files{
                "src/**.h",
                "src/**.hpp",
                "src/**.c",
                "src/**.cpp"
        }
        includedirs{
                "src",
                "include"
        }
```

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

```
$ premake5 gmake2
```

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/.

```
$ make
```

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

1. Make UAV::Test::Address with standard postal fields with getter and setters.
2. 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;
3. Make a UAV::Test::Banking::SafteyDepositBox object with the following methods and fields:
   - int boxNumber;
   - UAV::Test::Person person;
4. Make a UAV::Test::Person object that have the following fields:
   - std::string name;
   - int age;