

After doing this, write a one page description of how to create and use namespaces and separate compilation in your environment

To create a custom made namespace, you need to create a number of different files. You will need to create a header file that just like in your standard program, gives the function declaration that is to be called by your namespace program. In here you will include your parameters, just as you would for any function that is used. Once you've completed this, you will create another file where you will need to make sure to `#include "nameoftheheaderfile.h"`. You will use this file to create the namespace, much as you would create a function called namespace, initialize it as a type namespace, give it a title like namespace yourNameSpace, and then include the namespace body, or definition. Finally, in our last file, where we will run our actual program, we must once again `#include "namespacename.h"` file, and we can then use the namespace within our program. In order for it's definitions to be applied, we can either declare it as just `#include "yourNameSpace"` or we can call it on a specific line with `yourNameSpace::function` to use it within our program as we would any other.

In order to compile these files, we can compile them all together, but listing each file used, other than the header files, as they are already included within our others. To compile them separately, we can use the `-c` flag to compile just the source code, creating object (.o) files such as `fgmain.o`, `f.o`, and `g.o` in our case. We will then compile these just as normal, with the new .o, in order to link together the object files. To simplify this process, we can create a makefile, which will search for any alterations within our program, or header files, and then consequently recompile those files for us, without us having to go through the process of compiling on our own. It essentially is able to keep track of the changes so that we don't have to.