

Exercise 3.1

- create a file labs/lab03/Makefile and add it to your repository.
- Copy the files hello.cpp main.cpp, factorial.cpp and functions.h from /home/csci553/lab03dat to your lab03 directory
- Add and check in these files to your repository

Exercise 3.2

Your job is to write the makefile for this “project”. Note that you could compile this project with one line:

```
$ g++ main.cpp hello.cpp factorial.cpp -o hello
```

However, you are to create a Makefile that will avoid updating and recompiling up to date files. Here are specifications for the Makefile you are to write:

- The makefile creates an executable called hello (default target)
- The makefile should have a rule, called run, this rule runs the hello program with default arguments foo and 5. However it must be possible to set the arguments for the program by setting a variable: ARGS when calling make
- The makefile must include rules for all and for clean
- You need to use patterns to first make object files (.o) from the .cpp files, then make the executable hello by linking the source files:
 - If you did this by hand, you would run the commands:

```
$ g++ -c main.cpp
$ g++ -c hello.cpp
$ g++ -c factorial.cpp
$ g++ -o hello main.o hello.o factorial.o
```
- Your makefile should only cause object files that are out of date to be recompiled
- Hint: start off simple (with the one line g++), then break it up to deal with dependencies, then introduce variables, and finally implicit rules.
- When you are done, make sure you check in your updated Makefile.