MakeFile Assignment

1/ A makefile gives directives to the utility called make and this utility will then be able to follow certain instruction and compile a set of source code.This is very useful for large projects, especially when you want to send the project to someone else.

2/Essentially a makefile contains a set of rules used to build an application. The first rule seen by make is used as the default rule. A rule consists of three parts: the target, its prerequisites, and the command(s) to perform:

```
target: prereq1 prereq2
        commands
```

The target is the file or thing that must be made. The prerequisites or dependents are those files that must exist before the target can be successfully created. And the commands are those shell commands that will create the target from the prerequisites.
Here is a rule for compiling a C file, foo.c, into an object file, foo.o:

```
foo.o: foo.c foo.h
        gcc -c foo.c
```

The target file foo.o appears before the colon. The prerequisites foo.c and foo.h appear after the colon. The command script usually appears on the following lines and is preceded by a tab character.
When make is asked to evaluate a rule, it begins by finding the files indicated by the prerequisites and target. If any of the prerequisites has an associated rule, make attempts to update those first. Next, the target file is considered. If any prerequisite is newer than the target, the target is remade by executing the commands. Each command line is passed to the shell and is executed in its own subshell. If any of the commands generates an error, the building of the target is terminated and make exits. One file is considered newer than another if it has been modified more recently.

4/If you run make clean(with space in between,it will run that target), it will remove the objects and the exe file.