Name: ______________________________

# Midterm Exam
## CSC 271 (Software I)
Dr. S. Bloch
Oct. 18, 1994

You may use your own class notes and the Hahn textbook. You may *not* discuss the problems with other human beings (except me), nor use computers during the exam.

1. (24) This is an exercise in reading documentation. Following is part of the `man` page for an imaginary command; your job is to distinguish legal from illegal commands.

   NAME **dwim** — Read my mind and "do what I mean"

   SYNOPSIS **dwim** [*option*] ... [*file*]

   The **dwim** command reads your mind and does whatever you want to the specified file. It accepts several options, which may be combined:

   **-a** Specifies output in American Standard English. The default output format.

   **-k** Specifies output in Korean.

   **-o** [*filename*] Output to the specified file. If no filename is given, output to **stdout**, the default.

   **-q** Quiet mode. Do not produce warning or error messages.

   Assuming there were such a program, **which of the following commands would be legal?** For the legal ones, *briefly* describe what they do; for the illegal ones, *briefly* explain why they're illegal.

   (a) `dwim`

   (b) `dwim file1`

   (c) `dwim file1 -k`

   (d) `dwim -k file1 -o`

   (e) `dwim file1 file2`

   (f) `dwim -o file1 -k`

   (g) `dwim -o file1 file2`

   (h) `dwim -kq file1 <file2 >>file3`

2. (30) For each of the following tasks, **give a single command line** to accomplish it on our Unix system.

   (a) Copy all the files in the directory `/users/std/emily/class/hw3` to the `sw1/hw3` subdirectory of your own home directory.

   (b) As above, but some of the things in `/users/std/emily/class/hw3` may themselves be directories; they should be copied along with their contents. In addition, give everybody on the system read access, and yourself write access, to the new copies of the files.

   (c) Get a list of all the directories in `/users/std/emily/class/hw3`, with their permission bits. You do *not* want to see all the other files that aren't directories.

3. (36) **Write a shell script** with two parameters, the first a file name and the second a directory name (for extra credit, have the second parameter default to the current directory if not specified). Your script should execute an `ls` command on the specified directory, putting the output into the specified file name in the directory *above* the specified one. (This example is for real: I actually needed to do something like this last month.) Don't worry about what to do if the user gives the wrong number or kind of parameters.

   For example, if I named the shell script "`prob3`" and typed "`prob3 outfile ~/install`", it would execute the command "`ls ~/install`" and put the output into the file `~/outfile`.

   Extra credit: write the script so it'll run, successfully and without warning messages, regardless of whether the output file already exists.

4. (10) **Write a shell script** which outputs an exact copy of itself to **stdout**. You may *not* assume the script is named anything in particular, *i.e.*, it should still work if somebody renames it.