CIS-350
INFRASTRUCTURE TECHNOLOGIES

LAB # 2

Due date: **See Blackboard**

Objectives: Learn about **Windows Command Prompt (MS-DOS)**

This is not a group lab. You should do this lab individually.

You will be doing much more with **.bat** files than is listed in Lab 1 (chapter 7 in Davis & Rajkumar textbook) on MS-DOS. Follow this list of explanations and activities. Create the required batch files on your thumb drive or CD and submit it for evaluation. (Many commands below will refer to the E drive – thumb drive. If you use a CD, the drive name will change to D or Z depending on the system.)

If you choose to submit Lab1 and Lab2 on a single medium, first create two directories name Lab1 and Lab2, respectively. Then move all Lab1 related directories and files to the Lab1 directory, and move Lab2 files to the Lab2 directory. The root directory should only contain Lab1 and Lab2 directories.

1. PARAMETERS - Parameters may be used in any place in a batch file where a parameter would normally be used as a part of the MS-DOS command being run. Markers are used within the batch file to signify which parameter goes where. Markers are comprised of a percent sign (%) and a single digit between 0 and 9. Type in the following batch file named **REPEAT.BAT**. (If you use a thumb drive, change the default drive to E: so that all these activities are done on your disk in E: drive. If you use other medium, change the default drive accordingly.)

From the E (or another) prompt, type:

EDIT REPEAT.BAT <hit Enter>

The above *edit* command is available in Windows XP. If the *edit* command does not work in Windows Vista or Windows 7, **you may use Notepad to create** this and all other **batch files** required in this assignment.

The above command invokes the MS-DOS editor screen. Now, type:

ECHO %0 %1 %2 %3

ECHO can show messages on the screen. Open the File pull-down menu, save the file, and exit the editor.

Now at the E (or another) prompt, type: **REPEAT Red Blue Green**

Four parameters are passed to the .bat file and are placed into the ECHO command in order received. Note an important point -- marker zero is assigned the name of the .bat file in the form you typed it in. The parameters and markers were related as follows...

```
        REPEAT      Red   Blue  Green
ECHO      %0        %1    %2    %3
```

2. BATCH SUBCOMMANDS - In addition to the normal MS-DOS commands, batch files have their own subcommand structure.

Here are 3 simple Batch subcommands.

a. ECHO - This turns command display on/off or may display a message. You have already seen it send a message to the screen. It can help clear up the clutter on the screen when a batch file executes. Normally, all the batch file commands would show up on the screen as if you were typing them. This can get distracting. If you want to suppress the command echoing type **ECHO OFF** or to start echoing, type **ECHO ON**. Create the following short batch file and save it on the E thumb drive.

From the E prompt, type:

EDIT TRYECHO.BAT <hit Enter> or use Notepad

Put the following commands in the file:

```
ECHO OFF
DATE
ECHO THE ECHO IS OFF
TIME
ECHO ON
DATE
ECHO THE ECHO IS ON
TIME
```

Save the file and exit the editor.

When you run this, you should see how ECHO works.

b. REM - REMark can be used to send messages to the screen or simply to document some part of your batch file's operation.

c. PAUSE - This command is to stop the execution of the batch file until you press a key. This can allow you to perform a necessary task - like changing a disk or verifying errors - as the remaining parts of the batch files are executed. PAUSE will optionally display a message if you desire. Type and run this short batch file.

From the E (or another) prompt, type:

EDIT REMPAUSE.BAT <hit Enter> or use Notepad

```
REM THIS WILL SHOW HOW REM WORKS.
PAUSE PUT ANY MESSAGE HERE.
REM THIS WILL SHOW PAUSE ALSO.
PAUSE
```

REM YOU ARE NOW DONE.

Save the file and exit the editor.

d. You may interrupt any batch command by hitting the **Break** key, <**CTRL-C**> or <**CTRL-S**>. Try it. It will give you a chance to verify. If you say no, it will ignore the current command and execute the rest.

3. Here are two more complicated batch commands to try.

a. GOTO - The format for this command is: GOTO LABEL where LABEL is a line in your batch file that starts with a colon (:) followed by up to an eight character name (actually you can type any number of characters, but MS-DOS only recognizes the first eight).

Here is an endless loop program.

EDIT LOOP.BAT <hit Enter> or use Notepad

:START
REM ... This is being typed by an endless loop.
GOTO START

Save the file and exit the editor.

When executed, this file will continue to print the endless loop line and the GOTO command until you use <**CTRL-Break**>.

b. IF - This subcommand allows you to make decisions within your batch file. The format is:
If Condition Command where,

Command = Any legal DOS command or batch file subcommand
Condition = One of three tests that yield true or false:

1. The ERRORLEVEL of a program.
2. Two strings are equivalent.
3. A file exists in the current directory.

Unlike programming languages, which allow many logical tests in an IF statement, the batch IF statement is limited to only the three above.

Condition 1 : ERRORLEVEL is a number that indicates to DOS whether the last program run was successful. A zero (0) indicates a good run, anything above zero indicates and error condition. (Only DOS commands BACKUP and RESTORE have an exit code.) You probably won't use this one often.

Condition 2 : String comparison is indicated by double equal signs:
String1 == String2
compares the two strings. This comparison is often used with parameters and markers to check for a particular entry. For example,
IF %1 == 40 MODE C40
checks parameter one for 40 and, if so, changes the display to 40-columns wide.

Condition 3 : The logical test checking for a file has the format: EXIST E:Filename
You can use this test to check and see if a DOS disk is in the active drive. Another use might be to check for a particular file to see if the user has a valid disk in the drive. Pathnames are NOT allowed.

An example will follow.

c. One batch file may call another. This batch file can be used to establish a password for running a program. The batch file is named GO.BAT and calls the program named LOOP.BAT. (It could call any .COM file or application package.) This file attempts to demonstrate all the batch file commands you have learned.

From the E (or another) prompt, type

EDIT GO.BAT <hit Enter> or use Notepad

```
ECHO OFF
IF %1==XYZ GOTO GOOD
ECHO BAD PASSWORD... ENDING
GOTO END
:GOOD
ECHO YOU'RE OK...STARTING
IF EXIST E:LOOP.BAT ECHO YES THERE IS A LOOP
PAUSE
REPEAT RED BLUE GREEN
:END
```

Save the file and exit the editor.

Note: If you do not use the E thumb drive, change the drive name on the statement: IF EXIST E:LOOP.BAT …… accordingly.

Run this by typing **GO ABC** and also with **GO XYZ**.

4. Now try your hand at writing a simple batch file that will manipulate other files on your disk, or other batch files. Your batch file must use some parameters, ECHO, PAUSE, REM, GOTO, IF, and several DOS commands listed below.

DIR
DIR with * wild card
DIR with ? for one character

TREE (may not work on some installations)

VER to find the version of DOS in use

COPY

DATE to set the date as 04/03/10

TIME to set a different time than the actual time when you run this macro

MD to create at least two new subdirectories

CD to change between subdirectories

TYPE to display file content

RD to remove one of the directories you created

SORT by piping a file in and sending a file out. Display the file content before sorting and after sorting.

Your batch file (macro) should be executable by typing the file name followed by parameters, if any. As for file name use the first four letters of your last name (if your last name has less than four letters then use your first name to pad up to four characters). For example, W. H. Johnson will create the files as JOHN.BAT. To create a macro, you can use a Notepad or the mentioned MS-DOS editor called *edit*.

You may have to create several dummy directories and files to be able to check how the commands in your batch file work. You may use the directories and files you have created in lab #1.

Suitably title all your output at each stage. Use the *ECHO* command to achieve this.

The *ECHO* command turns command display on/off or may display a message. It can help clear up the clutter on the screen when a batch file executes. Normally, all the batch file commands would show up on the screen as if you were typing them. This can get distracting. If you want to suppress the command echoing, type *ECHO OFF* or to start echoing, type *ECHO ON*. The command *ECHO I love New York* will display the message *I love New York* on the screen. To suppress the word *ECHO* from displaying, type *@ECHO OFF*.

5. You should have the following batch files on your thumb drive: REPEAT, TRYECHO, REMPAUSE, LOOP, GO, and your own batch file. Submit your thumb drive or CD with these batch files and write your full name, section #, and lab # on it. Also, write and submit one- or two-paragraph evaluation of the lab. Describe what commands worked well and what did not.