INFO 341 Assignment 1 - A Simple Console

Due: Friday October 5, 2012 by 10:00pm

This assignment is to build a small "console" application. A console is a text based interface that allows a user to command the computer to do something. Your console application will know just a few simple commands and will respond to those commands in a reasonable way. The assignment is comprised of several tasks. When you have completed all of the tasks you will have completed the assignment and can submit it for grading. The tasks are listed in a sequence that builds one upon the other. If you complete the optional task, you may be eligible for a small amount of extra credit.

Task 1 – Echo Text

Build a Java class that reads text entered at a command prompt and writes the exact same text back to the user. The easiest way to do this is to read input from `System.in` and write output to `System.out`. For example:

```
java Console
command> this is a command
this is a command
command> what is up, doc?
what is up, doc?
command>
```

Your Java console should specifically prompt the user to enter a command with the string "`command> `" (without the quotes). Note that there is a space at the end of the prompt string to prevent the user command from looking like an extension of the command prompt when they enter it.

Task 2 - Responding to Commands

Your console will need to take special action when it sees certain commands. For example, when your console sees the command "`halt`" it should stop the entire program. Modify the simple console in Task 1 to handle the following commands:

`halt` - Your console should stop running and exit cleanly with no errors.

`time` - Your console should print the current wall clock time in the format HH:MM:SS [AM/PM]. If you output time in a 24 hour clock format then you should not include an AM/PM indicator. If you use a 12 hour clock format then you must include the AM/PM indicator.

`date` - Your console should print the current date in the format MM/DD/YYYY, where MM is a two digit integer month, DD is a two digit day, and YYYY is the full four digit year. In cases where the month or day is not two digits, a zero "0" place holder should be used to make the month or day two digits.

`help` - Your console should print an explanation of the commands that it knows

If your console application receives a command that it does not recognize your console should specifically print "`Unrecognized command:`" (without the quotes) followed by the string of the command that it did not recognize.

Task 3 - (Optional) Setting a Command Prompt
A command will sometimes take command parameters. A simple example of this is to implement a new command that will allow the user to set the command prompt to any string that they want.

`setprompt <`*`a new prompt string`*`>` - Your console should take the string entered by the user after the "`setprompt`" command beginning with the first non-whitespace character up to a newline or carriage return character, as a new prompt. This prompt should replace the old prompt until the user issues the `setprompt` command with a new prompt.

If you optionally implement the setprompt command you should update your console `help` command (from Task 2) to indicate that you have that capability.

Assignment Turn In
Your Java code should compile and run correctly from the command line. If you use a development environment, you should make certain that your code will compile and run from the command line.

All of your Java classes should be in a `package` that is named by your lastname and the last three digits of your student ID. So if your last name was "McStudent" and the last three digits of your student ID were "820" your code would include a statement like:

```
package mcstudent820;
```

If you have your Java CLASSPATH variable set correctly you would be able to run your Console from the command line by typing:

```
java mcstudent820.Console
```

You will turn in your Java source files as one Zip file. Since packages are assumed to be in directories that are named the same as the package (e.g., in the example above, your code would be in a directory named "mcstudent820") you can just Zip that directory and submit the single file through the course Catalyst Dropbox. Absolutely no assignments will be accepted through email.